UNIVERSITY OF THE PELOPONNESE
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

Master of Science in
Computer Science

Post-Quantum Cryptography in Modern Critical Environments

Postgraduate Thesis

Author:
Konstantinos Spalas

Supervisor:
Nicholas E. Kolokotronis
Professor

May 2026

Preface

The era we live in is shaping many changes and developments that will affect various modern environments. The development of quantum computers is a great technological achievement that will accelerate the solution of computational problems that were either considered unsolvable until now, or the available resources were not sufficient. Various cryptographic systems are based on such problems and as a result will be considered obsolete in the post-quantum era.

In this context, the National Institute of Standards and Technology (NIST) initiated in 2016 processes to search for cryptographic algorithms that would resist the attacks known so far based on these quantum algorithms. Therefore, NIST, after a thorough evaluation, standardized certain post-quantum algorithms that can be integrated by various communication systems. Specifically, algorithms based on the difficult problem of finding a "Shortest Path", on the mathematical structure of the lattices, prove to be very fast and with small key sizes. Nevertheless, NIST continues the process of standardizing new cryptographic systems, now giving weight to algorithms based on the difficult problem of codes.

Post-quantum cryptographic algorithms are resistant to quantum attacks, but may be vulnerable to side-channel attacks. These attacks are based on recording the electrical energy emissions of a system at the time it performs cryptographic operations. This thesis aims to study the effectiveness of these attacks using new technologies such as machine learning, targeting the key encapsulation mechanisms. The question that arises is whether these techniques are capable of predicting the secret structures of post-quantum cryptosystems, which, by extension, will lead to an effective attack against vital structures even without the use of quantum computers. The possible verification of this theory should lead to the improvement of the cryptosystems in question.

Critical environments such as blockchain, aviation, and smart cities require the integration of these post-quantum algorithms, which should be resistant to any kind of attack. This fact will enhance the security of the environments but raises the question of how much this integration will affect the environments' performance. So, after examining potential leakage during their cryptographic operations, this thesis studies the complexity that post-quantum cryptography will potentially add to critical environments. This time, the study focuses on digital signatures and hash functions, taking into account the size of their key elements, such as cryptographic keys, and their nominated performance, which will be normalized to indicative hardware performance.

In this context, this study is structured according to the following five chapters:

1. Introduction to the main idea.

2. This chapter attempts to analyze in brief the types of modern classical cryptography and their key representatives to help the reader to understand their mathematical fundamentals, key elements, and usage.

3. The third chapter analyzes the two major quantum algorithms that can jeopardize the security level of well established classical cryptographic algorithms, such as RSA, Elliptic Curves, and Hash-based cryptography. In addition, it introduces the five post-quantum cryptography families and the initiative of the National Institute of Standards and Technology to standardize the most effective of them for key encapsulation mechanisms and digital signatures.

4. The fourth chapter represents the vulnerabilities of the post-quantum cryptographic algorithms besides these seem to be unbreakable with respect to quantum attacks. While traditional attacks can be enhanced by quantum computational strength, secret primitives can be disclosed utilizing solely classical computers and special equipment that can collect electromagnetic emissions during their cryptographic operations.

5. The final chapter focuses on the effectiveness and performance of post-quantum cryptography in critical environments. In particular, the mega question that arises regarding their complexity and performance is whether or not post-quantum cryptography can be implemented and utilized during the pre-quantum era.

Finally, I express my sincere gratitude to my supervisor, Professor Nicolas Kolokotronis, for introducing me to the field of cryptology and subsequently to the emerging area of post-quantum cryptography. I am deeply grateful for his guidance and continuous support throughout my research.

Konstantinos Spalas
Tripolis
February 2026

Σύνοψη

Η εποχή που ζούμε εκκολάπτει πολλές αλλαγές και εξελίξεις οι οποίες θα επηρεάσουν διάφορα σύγχρονα περιβάλλοντα. Η ανάπτυξη των κβαντικών Η/Υ αποτελεί μεγάλο τεχνολογικό επίτευγμα που θα επιταχύνει την επίλυση προβλημάτων υπολογισιμότητας που είτε μέχρι σήμερα θεωρούνταν μη επιλύσιμα, είτε οι διαθέσιμοι πόροι δεν ήταν επαρκείς. Ο σύγχρονος κβαντικός αλγόριθμος του Lov Grover επισπεύδει την αναζήτηση στοιχείων σε μια μη ταξινομημένη λίστα, κάτι που μπορεί να βοηθήσει διάφορους τομείς, όπως η ιατρική, ταυτόχρονα όμως δύναται να χρησιμοποιηθεί κακόβουλα στην προσπάθεια αποκάλυψης κρυπτογραφικών κλειδιών.

Από την άλλη, ο κβαντικός αλγόριθμος του Peter Shor δύναται να παραγοντοποιήσει έναν μεγάλο αριθμό σε δυο πρώτους, ένα μαθηματικό πρόβλημα που μέχρι τώρα ήταν δύσκολο να επιλυθεί. Πάνω σε αυτό το πρόβλημα βασίζονται διάφορα κρυπτογραφικά συστήματα με αποτέλεσμα να θεωρούνται απαρχαιωμένα στην μετακβαντική εποχή.

Στο πλάισιο αυτό, το National Institute of Standards and Technology (NIST) ξεκίνησε το 2016 διαδικασίες αναζήτησης κρυπτογραφικών αλγορίθμων οι οποίοι θα αντιστέκονται στις μέχρι τώρα γνωστές επιθέσεις που βασίζονται στους ανωτέρω κβαντικούς αλγόριθμους. Ως εκ τούτου, ο NIST, κατόπιν διεξοδικής αξιολόγησης, τυποποίησε ορισμένους μετακβαντικούς αλγόριθμους οι οποίοι μπορούν να ενσωματωθούν από διάφορα επικοινωνιακά συστήματα. Συγκεκριμένα, αλγόριθμοι βασισμένοι στο δύσκολο πρόβλημα εύρεσης «Σύντομου Δρόμου», επί της μαθηματικής δομής των πλεγμάτων, αποδεικνύονται ταχύτατοι και με μικρό μέγεθος κλειδιών. Παρόλα αυτά, ο NIST συνεχίζει τη διαδικασία προτυποποίησης νέων κρυπτογραφικών συστημάτων δίνοντας πλέον βαρύτητα σε αλγόριθμους που βασίζονται στο δύσκολο πρόβλημα των κωδίκων.

Είναι γεγονός πως, η σωστή επιλογή μεγέθους κλειδιών σε αλγόριθμους βασισμένους σε κώδικες τους καθιστούν ανθεκτικούς απέναντι σε κβαντικές επιθέσεις. Εντούτοις, οι μετακβαντικοί αλγόριθμοι μπορεί να αποδειχθούν ευάλωτοι στις μέχρι τώρα γνωστές επιθέσεις πλευρικού καναλιού. Αυτές οι επιθέσεις βασίζονται στην καταγραφή εκπομπών ηλεκτρικής ενέργειας ενός συστήματος τη στιγμή που εκτελεί κρυπτογραφικές πράξεις. Η παρούσα διπλωματική εργασία έχει ως στόχο να μελετήσει την αποτελεσματικότητα των εν λόγω επιθέσεων κάνοντας χρήση νέων τεχνολογιών όπως η στατιστική και η μηχανική μάθηση. Το ερώτημα που γεννάτε είναι αν αυτές οι τεχνικές είναι ικανές να προβλέψουν τις μυστικές δομές των μετακβαντικών κρυπτοσυστημάτων, όπου, κατ' επέκταση, θα οδηγήσει σε αποτελεσματική επίθεση εναντίων ζωτικών δομών ακόμα και δίχως τη χρήση κβαντικών Η/Υ. Η πιθανή επαλήθευση αυτής της θεωρίας θα πρέπει να οδηγήσει στην βελτίωση των υπόψη κρυπτοσυστημάτων.

Σε κρίσιμα περιβάλλοντα όπως η Αεροπλοΐα, οι έξυπνες πόλεις, το blockchain κ.α, επιβάλλει η ενσωμάτωση των εν λόγω μετακβαντικών αλγορίθμων, οι οποίοι θα πρέπει να είναι ανθεκτικοί σε κάθε είδους επίθεση. Το γεγονός αυτό, αφενός θα ενισχύσει την

ασφάλειά αυτών των περιβαλλόντων αφετέρου, εγείρει το ερώτημα κατά πόσο αυτή η ενσωμάτωση θα επηρεάσει την απόδοσή τους. Η παρούσα μελετά την πολυπλοκότητα που δυνητικά θα προσθέσει η μετακβαντική κρυπτογραφία, στα κρίσιμα περιβάλλοντα, αναλύοντας τα δομικά τους στοιχεία και ταυτόχρονα, κατόπιν σύγκρισης, θα προτείνει πιθανούς συμβιβασμούς και βέλτιστους συνδυασμούς.

Τέλος, εκφράζω την ειλικρινή μου ευγνωμοσύνη στον επιβλέποντά μου, Καθηγητή Νικόλαο Κολοκοτρώνη, που με εισήγαγε στον τομέα της κρυπτολογίας και στη συνέχεια στον αναδυόμενο τομέα της μετακβαντικής κρυπτογραφίας. Είμαι βαθιά ευγνώμων για την καθοδήγηση και τη συνεχή υποστήριξή του καθ' όλη τη διάρκεια της έρευνάς μου.

Contents

List of Figures

List of Tables

List of Algorithms

CHAPTER 1

Introduction

In the late 1970s, coding theory started shaping public-key cryptography. Some other cryptosystems, like the Rivest-Shamir-Adelman (RSA), relies on difficult mathematical problems such as factoring large prime numbers, and are proven to have substantial cryptographic strength which can achieve high security standards. However, the advent of quantum computing poses a significant threat to these foundational assumptions. In particular, Peter Shor's quantum algorithm demonstrates that a sufficiently large quantum computer could efficiently solve the aforementioned problem of factoring large prime numbers, which several cryptographic systems rely on. As a consequence, much of the cryptographic infrastructure in use today is rendered insecure in the post-quantum era. This looming vulnerability has catalyzed global efforts to develop post-quantum cryptography (PQC), a new class of cryptographic algorithms believed to be secure against both classical and quantum adversaries. The urgency to migrate to quantum-resistant cryptographic primitives is not merely speculative. In addition, Lov Grover's modern quantum algorithm speeds up the search for elements in an unordered list, which can help various fields, such as medicine development, but at the same time it can be used maliciously in the attempt to reveal cryptographic keys.

To avoid such threats, transitioning to PQC is essential to ensure long-term data confidentiality and system integrity. PQC schemes are based on a range of mathematical foundations that are currently considered resistant to quantum attacks. These include lattice-based cryptography (e.g CRYSTALS-Kyber, CRYSTALS-Dilithium), code-based schemes (e.g Classic McEliece), multivariate quadratic equations, hash-based signatures, and isogeny-based cryptography. Each of these families presents unique advantages and limitations in terms of security guaranties, computational efficiency, key and ciphertext sizes, and implementation complexity.

Beyond theoretical security, practical implementation concerns must also be addressed. Cryptographic algorithms, especially those deployed in embedded and constrained environments, are susceptible to a variety of side-channel and fault attacks that exploit physical or timing characteristics rather than cryptanalytic weaknesses. Consequently, a comprehensive analysis of post-quantum cryptographic schemes must consider both algorithmic robustness and implementation-level resilience. This thesis aims to investigate the current landscape of post-quantum cryptography, with a par-

ticular focus on the security, efficiency, and practicality of selected algorithms. It further explores known attack vectors, both theoretical and physical, that challenge the robustness of these schemes, and discusses mitigation strategies necessary for their secure deployment in real-world systems.

More precisely, in Ch. 4 an ambitious experiment was executed that attempts to artificially implement a side-channel analysis. Instead of the expensive equipment needed to participate in such an attack, a low cost set up capable of simulating dedicated probes and digital oscilloscopes was used. The findings reveal that unmasked software implemented code-based post-quantum key encapsulation mechanisms can leak significant information. Assuming the null hypothesis statement that "Software post-quantum cryptography does not leak information", the computation of the correlation between the traces and the intended leakage is capable of rejecting this hypothesis, leading to the conclusions that there is significant leakage. The conclusions allow to move forward using machine learning models that managed to predict secret bits with scores above the probability of a raw guess. Under the same rationale and with the help of consulting the literature, post-quantum digital signature secret values can also potentially leak information. As a countermeasure, hardware implementations can prevent unintended emissions by injecting random noise, which can harden the malicious job of an adversary.

The following is Ch. 5 focuses mainly on post-quantum digital signatures with respect to their performance in vital environments. For example, the *Blockchain* is a constantly emerging structure that has been gaining participation in several infrastructures such as healthcare and transportation. Breaking down blockchains primitives, it is obvious that transaction verification and the Merkle-Trees utilize digital signatures and hash functions, and thus quantum computers could pose a threat, signifying that post-quantum cryptography should be included. In this context, this chapter come to the conclusions that there is no significant overhead if the blockchain inherits post-quantum algorithms substituting the quantum vulnerable classic cryptography.

Another critical environment configured to contain cryptography is *Aviation*. Until now, civil aviation communications are insecure channels that rely mostly on their effectivity on the human factor. Due to increasing number of flights, the tendency towards automation, and the lack of additional frequencies, future communications would substitute human factor with text based messages. Thereby, by default, such a communication demands the implementation of a secure public key infrastructure being able to serve aviation during the post-quantum era. In this chapter, an attempt was made to calculate the performance of future civil aviation communications, regarding the authentication procedures between two parties, the aircraft and the control tower. The results show that quantum-safe digital signatures perform as well as the classical ones. Therefore, once more, the question of utilizing a post-quantum algorithm from the very beginning needs a solid answer. The main reason for this question is to avoid transitioning from the pre to post-quantum era, which may lead to hardware incompatibilities, more maintenance and diversity.

Finally, *Smart Cities* is another emerging batch of technological entities that will convert our daily living into a modern and automated environment. The internet of things is the key particle of this attempt, and as a result should be shielded against any kind of attack. Therefore, this thesis studies smart transportation systems, focusing on their security concepts. In particular, it pinpoints several communication protocols between smart cars and sensors that must exchange information securely. Hence, smart means of transportation should authenticate themselves, leading to the inevitable adoption of post-quantum signature schemes.

CHAPTER 2

Introduction to Cryptography

When transmitting sensitive information over a public channel, there is the risk that an eavesdropper might intercept, steal, or alter it. Thankfully, modern communication is safeguarded by state-of-the-art cryptography, securing messages even if they are intercepted. The core concept of cryptography is that the sender chooses a message to send, encrypts it using a specific method, and then transmits the encrypted message. The recipient receives this encrypted message, known as ciphertext, and applies a decryption process to retrieve the original message. The key point is that even an adversary intercepts the message, he will not be able to understand it without knowing the decryption method. The encryption and decryption processes typically rely on a piece of confidential information, called a shared secret key, which is essential to both operations. Encryption and decryption are essentially reverse processes of each other.

2.1 Types of Cryptography

Cryptography refers to the science of securing communications, while a cryptosystem is a specific implementation of a cryptographic technique, also called a cipher or a cryptographic system. Cryptanalysis, on the other hand, involves identifying and exploiting weaknesses in a cryptosystem, often referred to as *attacks*. Cryptosystems are generally classified into two types: *private-key* (or symmetric) and *public-key* (or asymmetric), but we can also include *hash functions* as a type of cryptography.

2.1.1 Private (or Symmetric) Key Cryptography

Symmetric key cryptography refers to encryption methods in which both the sender and receiver share the same secret key. Until the introduction of public key cryptography, symmetric schemes were the only form of encryption known.

Symmetric ciphers are typically classified into two families: *block ciphers* and *stream ciphers*. Block ciphers encrypt fixed-size blocks of plaintext, i.e 128 bits, into ciphertext blocks of the same size, whereas stream ciphers operate on plaintext one symbol or bit at a time, producing a continuous stream of encrypted output.

The Data Encryption Standard (DES) and the Advanced Encryption Standard (AES) [1] are examples of block ciphers designated as US government standards. Although DES was withdrawn after AES, its stronger variant Triple-DES (3DES) became the official standard and had been widely deployed for decades in several domains. However, 3DES is also being phased out due to efficiency and security concerns. Many other block ciphers have been proposed, but only a few have withstand extensive cryptanalysis.

On the contrary, stream ciphers generate a key-stream from an internal state initialized with the secret key. This key-stream is then combined with plaintext (often via the XOR operation) to produce ciphertext, resembling the one-time pad. A widely used historical example is RC4 [2]. Additionally, block ciphers can emulate stream ciphers when used in certain modes of operation, i.e. counter mode, producing a key-stream from block output.

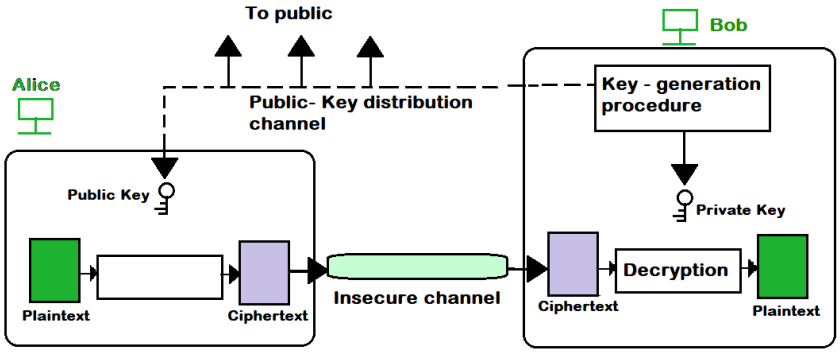

Figure 2.1: Symmetric key encryption.

In symmetric cryptography, both encryption and decryption are based on the same shared secret key as seen in Fig. 2.1. For an outside observer, the only available information is intercepted ciphertexts, and without knowledge of the private key, the adversary is assumed to have negligible advantage in recovering the plaintext. However, while the strength of the symmetric key cryptography is its key, simultaneously is its weakness because both parties should exchange the key through a secure channel. Furthermore, they must exchange several keys to be more secure for future communications, because switching keys maintains the message secrecy against known-plaintext attacks. In this context, the key management becomes harder, and additionally, if the parties want to communicate become more, then each pair of peers must have its dedicated secret key, which converts the system complicate. This difficulty of handling multiple keys overcomes public key cryptography.

2.1.2 Public (or Asymmetric) Key Cryptography

Public key cryptography, also known as asymmetric cryptography, provides an alternative approach to encryption and decryption compared to symmetric key methods,

trying to eliminate aforementioned problems of keys handling. The logic of symmetric encryption is shown in Fig. 2.2. In this paradigm, each participant, such as Alice and Bob, maintains a distinct private key and a corresponding public key. The public key is widely available, whereas the private key remains secret, stored in the participant computer. Typically, a public and a private key pair is mathematically related in a manner that when one is used for encryption, the other one is used for decryption.

Suppose that Alice wishes to send a message to Bob. She first obtains his public key and uses it to encrypt her message. The encrypted message is then transmitted to Bob, who decrypts it using his private key, recovering the original message. Unlike symmetric key cryptography, an attacker now has access not only to the ciphertext but also to Bob's public key. Therefore, the security of the system relies on the computational difficulty in deriving the private key from the corresponding public key. A well-known example of public key cryptography is the RSA algorithm, which underpins many secure communication systems.

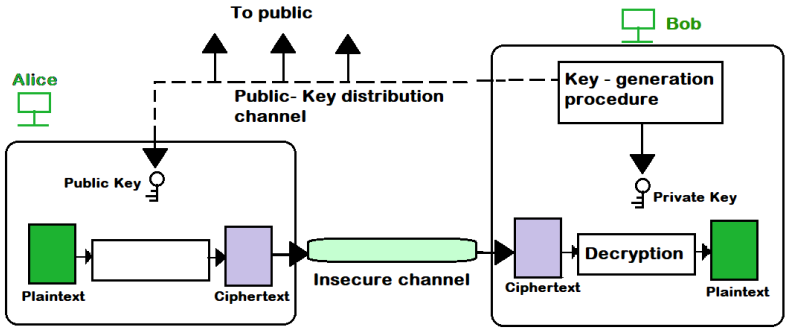

Figure 2.2: Asymmetric encryption.

2.1.2.1 The Rivest, Shamir, Adleman (RSA) Cryptosystem

RSA, named after its inventors Rivest, Shamir, and Adleman [3], is one of the most widely used public key cryptosystems. To set up an RSA system, Bob first selects two large prime numbers p and q, and computes their product

$$n = p \cdot q$$

which serves as the modulus for both the public and private keys. The length of n, expressed in bits, defines the *key length*. Bob then chooses an integer e such that $1 < e < \phi(n)$, where $\phi(n) = \text{lcm}(p-1, q-1)$, and $\gcd(e, \phi(n)) = 1$. The pair (n, e) constitutes Bob's *public key*, which he publishes, while (p, q) is kept secret as part of his *private key*. Finally, the value d, used for decryption, is computed as $d \equiv e^{-1} \bmod \phi(n)$.

When Alice wishes to send Bob a secure message m, she retrieves his public key (n, e) and computes the ciphertext c as:

$$c = m^e \bmod n$$

Afterwards, Bob reveals the original message by executing:

$$m = c^d \bmod n$$

The security of RSA is fundamentally based on the hardness of factoring n to obtain p and q and consequently compute the value $\phi(n)$. Its advantages include simplified key management because there is no need for pre-sharing private keys, and the system scales well, as new users can be added without impacting existing participants. In addition, RSA, and generally public-key cryptographic schemes, can be used as digital signatures. In such a case, a message can be encrypted with the private key, which stands for the action of sign. Consequently, the receiver must decrypt the message with the sender's public key to verify his identity.

Asymmetric key cryptosystems are designed to allow the same algorithm to support both encryption and sign, simplifying key management, and broaden their use across communication protocols. RSA keys can be distributed openly, enabling secure communication without the need to share secret keys beforehand, a major benefit for large-scale and decentralized systems. Simplified key management means that there is no need for pre-sharing private keys, and the system scales well as new users can be added without impacting existing participants. Because RSA has been widely adopted for decades, it is supported by a mature ecosystem of hardware, software, and standardized implementations, ensuring interoperability and ease of deployment. Moreover, with sufficiently long key sizes, RSA remains highly resistant to attacks using classical computing methods, contributing to its continued reliability in many security applications. On the other hand, a notable disadvantage, in contrast to the symmetric systems, is that an attacker has more information, the public key, available for cryptanalysis compared to symmetric cryptosystems.

2.1.2.2 Elliptic Curve Digital Signature Algorithm (ECDSA)

The Elliptic Curve Digital Signature Algorithm (ECDSA) [4], [5] is a widely used public key cryptosystem that leverages the mathematics of *elliptic curves* over finite fields to provide digital signatures. ECDSA is commonly used in blockchain systems, including Bitcoin and Ethereum, to sign transactions and verify the authenticity of messages.

In ECDSA, each participant, such as Alice or Bob, generates a private key d, randomly selected from the interval $[1, n-1]$, where n is the order of a predefined elliptic curve point G, also called the *generator point*. The corresponding public key Q is computed as follows:

$$Q = d \cdot G$$

where the operation $\cdot$ denotes scalar multiplication on the elliptic curve. The private key d is kept secret, while the public key Q is publicly available. According to Fig. 2.3, a potential public key, produced from the generator point, could be $Q = 4 \cdot G$, where the private key d is equal to 4.

To sign a message m, the sender performs the following steps:

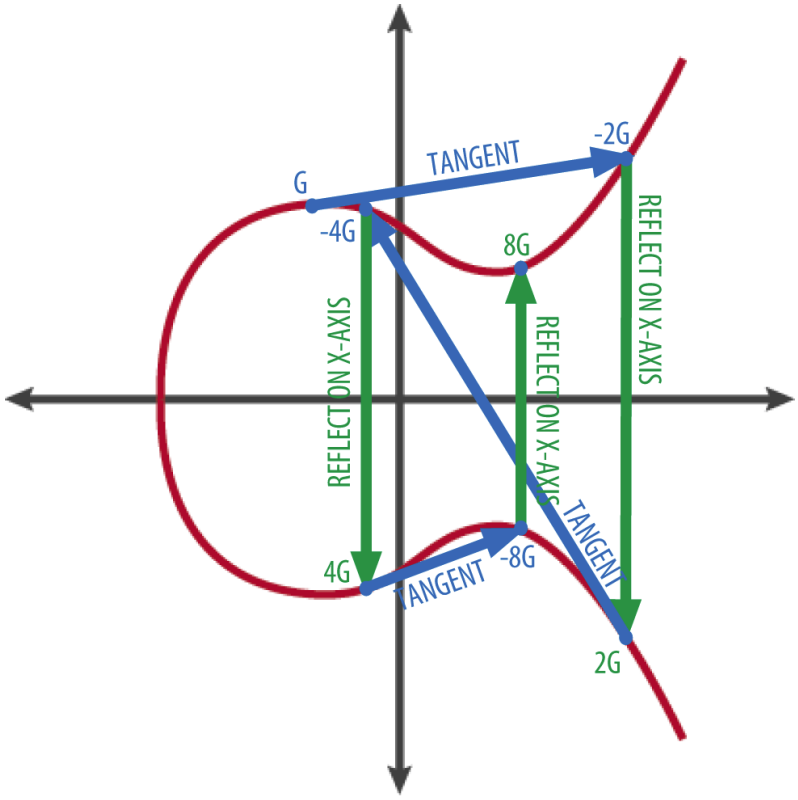

Figure 2.3: Elliptic curve.

1. Compute the hash $e = H(m)$ of the message using a cryptographic hash function (e.g. SHA-256).
2. Choose a random ephemeral key $k \in [1, n-1]$ and compute the point $R = k \cdot G$. Let $r = x_R \bmod n$, where x_R is the x-coordinate of R.
3. Compute $s = k^{-1}(e + d \cdot r) \bmod n$.
4. The signature is the pair (r, s).

To verify a signature (r, s), the recipient performs the following:

1. Check that r and s are integers in $[1, n-1]$.
2. Compute $w = s^{-1} \bmod n$, $u_1 = e \cdot w \bmod n$, and $u_2 = r \cdot w \bmod n$.
3. Compute the point $X = u_1 \cdot G + u_2 \cdot Q$. Let $v = x_X \bmod n$.
4. Accept the signature if and only if $v = r$.

The security of ECDSA relies on the computational hardness of the *Elliptic Curve Discrete Logarithm Problem (ECDLP)*, which states that given $Q = d \cdot G$, it is computationally infeasible to determine d. Moreover, ECDSA offers several advantages, such as

smaller key sizes compared to RSA for equivalent security levels, leading to faster computation and reduced storage, and strong security guaranties. In addition, several optimizations and alternative algorithms are often proposed to improve efficiency while maintaining security.

2.1.3 Hash-Based Cryptography

A *cryptographic hash function* is a mathematical algorithm that maps data of arbitrary size (commonly referred to as the "message") to a fixed-size bit string, known as the *hash value*, *hash*, or *digest*. A hash function h ideally works one-to-one, where each input x maps an output $y = h(x)$, fast. In addition, the inverse procedure should be, by default, computationally infeasible, making hash functions act as non-invertible. Furthermore, if the input x is altered even by little, then the output y will be completely different. In this case, if a sender sends a message m together with its hash value $h(m)$, the receiver, who receives both, can easily compute this value $h(m)$ and, if it is equal to the one received, is confident that he has an unchanged message.

Fig. 2.4 represents in a simple manner how almost identical inputs generate totally different digests. The only practical method to find an input that produces a specific hash is through brute-force search or the use of precomputed tables, such as rainbow tables. Due to these properties, hash functions are fundamental tools in modern cryptography.

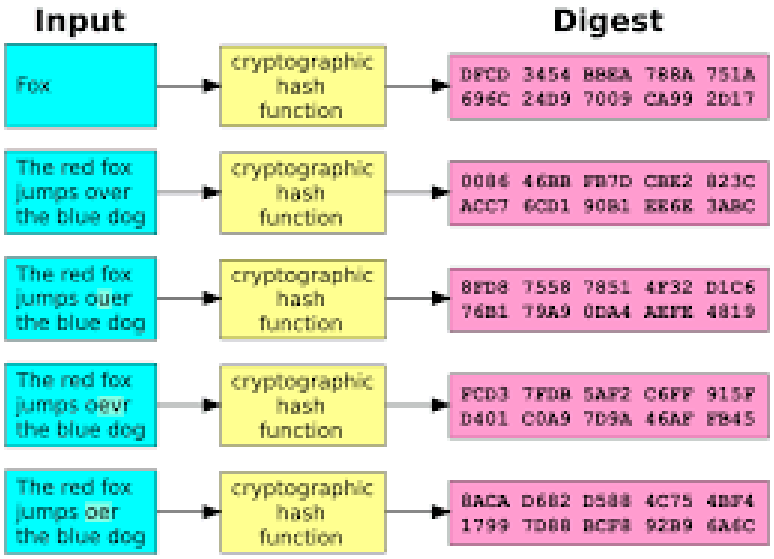

Figure 2.4: Hash-based cryptography.

Due to the property that any digest is unpredictable, hash functions are also widely used in *digital signature schemes*. Digital signatures serve a role similar to ordinary handwritten signatures: they are easy for the signer to produce, but difficult for others to forge. Furthermore, digital signatures are cryptographically bound to the content of

the message, ensuring that they cannot be transferred from one document to another without detection. Any modification of the message will result in a mismatch during verification, preserving the integrity and authenticity of the signed content.

CHAPTER 3

Introduction to Post-Quantum Cryptography (PQC)

Quantum computers [6] are a new paradigm of computation that uses principles of quantum mechanics, such as superposition and entanglement, to perform calculations fundamentally different from classical computers. Unlike classical bits, which represent either 0 or 1, quantum bits (*qubits*) can exist in superpositions of states, enabling a quantum computer to process a large number of possibilities, simultaneously. This property allows quantum computers to potentially solve certain computational problems much faster than classical machines.

Figure 3.1: The Quantum Computers expected capabilities until the year 2033.

The development of quantum computing has evolved rapidly over the past two decades. Early experiments focused on small-scale quantum systems with a few qubits. Recent advances in that field have led to medium-scale quantum processors with dozens to hundreds of qubits. Furthermore, ongoing research aims at building *fault-tolerant quantum computers* capable of executing large-scale algorithms reliably. Companies and research institutions around the world are actively pursuing scalable quantum ar-

chitectures, including superconducting qubits, trapped ions, and photonic qubits, each offering different trade-offs in terms of coherence times, error rates, and connectivity.

Although quantum computing promises breakthroughs in fields such as optimization, material science, and artificial intelligence, it also presents significant risks to classical cryptography. Many widely used cryptographic schemes, including RSA, ECC (Elliptic Curve Cryptography), and traditional discrete logarithm-based protocols, rely on the computational difficulty of problems such as integer factorization and discrete logarithms. Quantum algorithms, most notably Shor's algorithm [7], can solve these problems efficiently, rendering classical public-key systems insecure once sufficiently powerful quantum computers become available.

Symmetric key algorithms, such as AES, are also affected by Grover's algorithm [8], which can reduce their effective security by roughly half. In particular, this quantum algorithm can speed up the search for a key value in an unsorted list of size N, in $O(\sqrt{N})$ steps. Recall also that classical computers require $O(N)$ steps to obtain the key value. That is, the quantum algorithm speeds up by a quadratic rate, which is rather significant. For example, if someone searches for an item in a list of one billion items, then he would need to execute $O(10^9)$ steps in the worst case. On the other hand, using the quantum algorithm, someone would need, in the worst case $10^{4.5}$ steps to find the element in the list.

3.1 The Quantum Vulnerabilities of Classical Cryptographic Algorithms

The aforementioned potential vulnerabilities have driven the field of PQC [9], which seeks to develop cryptographic primitives resistant to quantum attacks. PQC includes lattice-based, hash-based, code-based, and multivariate polynomial schemes that are believed to remain secure against both classical and quantum adversaries. The adoption of PQC will be critical to the security of sensitive data and communications in a future where quantum computing capabilities will efficiently compromise current cryptographic standards.

In summary, quantum computers represent both an unprecedented computational opportunity and a major challenge for information security. Understanding the roadmap of quantum computing and its implications for cryptography, it is essential to develop robust and future-proof security mechanisms.

3.1.1 The RSA Cryptosystem

In the widely used RSA cryptosystem, the public key is defined as the product $n = p \cdot q$ of two secret prime numbers p and q. The security of this scheme depends critically on the computational difficulty of factoring in a large integer n into its prime numbers. However, in 1994, Shor introduced a quantum algorithm capable of efficiently factoring large integers, thereby threatening the security of RSA.

In particular, consider an integer $x < n$ that is coprime to n, and let r denote the order of x modulo n, i.e., the smallest positive integer such that $x^r \equiv 1 \pmod{n}$. Shor's algorithm relies on finding this order r. If r is even and $x^{r/2} \not\equiv -1 \pmod{n}$, then computing the greatest common divisor (gcd) of $x^{r/2} - 1$ and n, as well as the gcd of $x^{r/2} + 1$ and n, yields nontrivial factors of n. Formally,

$$\gcd(x^{r/2} - 1, n) \quad \text{and} \quad \gcd(x^{r/2} + 1, n)$$

produce two factors of n. The Euclidean algorithm allows the computation of gcd values in polynomial time, so the primary challenge is to determine r. Classical computers require exponential time to find r, as this involves solving the discrete logarithmic problem [10].

The probability that a randomly selected $x < n$ and a coprime to n will have an even order r that satisfies the above conditions is at least $1 - \frac{1}{2^{k-1}}$, where k is the number of distinct odd prime factors of n, and at most $\frac{1}{2}$. Therefore, the practical feasibility of factoring n using Shor's algorithm depends primarily on the quantum computer's ability to efficiently determine the order r. On classical computers, this task is infeasible for large n, making RSA secure in the absence of quantum computing resources.

3.1.2 The Elliptic Curve Cryptography

Elliptic Curve Cryptography (ECC), including widely used schemes such as ECDSA, is based on the hardness of the *Elliptic Curve Discrete Logarithm Problem (ECDLP)* [11]. Recall the previous chapter, which defines the fundamentals of elliptic curves. So, given a point G and a public key $Q = d \cdot G$, where d is the private key and $\cdot$ denotes scalar multiplication, the ECDLP asks for the determination of d given Q and G. Classical computers cannot efficiently solve this problem for sufficiently large private key sizes, which ensures the security of ECC in traditional cryptography. However, Shor's quantum algorithm can solve the discrete logarithm problem in polynomial time. Specifically, a quantum computer can compute the private key d from the public key Q by exploiting quantum parallelism and interference, making the security of ECC vulnerable when large-scale quantum computers become available.

Formally, if $Q = d \cdot G$, a quantum computer can find d such that:

$$d = \text{ECDLP}^{-1}(Q, G)$$

in polynomial time. This capability implies that signatures generated using ECDSA can be forged and encrypted data relying on elliptic curve keys can be compromised. The practical implication is that any system that depends on ECC for confidentiality, authentication, or digital signatures could be broken by a sufficiently powerful quantum adversary. This realization has motivated the development of post-quantum cryptographic schemes that aim to provide equivalent security against both classical and quantum attacks.

3.1.3 The Symmetric and Hash-based Cryptography

Recall that Lov Grover's quantum algorithm is able to speed up the search of an unsorted list, which could potentially attack symmetric cryptosystems. Such symmetric keys do not have any pattern regarding their structure and thus it is totally upon brute force attack to reveal such keys. As a result, to maintain equivalent protection against quantum-enabled adversaries, symmetric keys must be substantially increased in length, and cryptographic protocols may require reconfiguration to accommodate stronger parameters. Although symmetric encryption remains more resilient than traditional public-key systems, the quantum threat still dictates proactive adjustments to safeguard long-term confidentiality. For example, a symmetric key with 128 bit security, in the pre-quantum era, offers $\sqrt{2^{128}} = 2^{64}$ which is 64 bit security, in the post-quantum era.

In conjunction with Grover's algorithm and as per hash functions, there are several algorithms that are able to speed up more the search and are effectively affect the hash functions. According to the definition of hash functions, choosing a suitable size for the fixed length hash output y, the *digest*, it should be computationally impossible to calculate $x = h^{-1}(y)$, given y. Nevertheless, the hash functions face several vulnerabilities and scientists constantly work on mitigating them. Some of these vulnerabilities lead to attacks [12] that reveal the input x. In particular, *collision attack* refers to the process of identifying two distinct input strings, x_i and x_j, that produce the same hash digest y. Since hash functions are designed to accept inputs of arbitrary length but generate outputs of fixed length, it is mathematically inevitable that multiple inputs will map to the same output value. This type of occurrence, known as a collision, arises when different inputs yield identical hash results. This property can be exploited in applications where hash values are used for verification, such as password authentication systems, digital signatures, and file integrity checks.

In addition, research in [13] has shown that when an additional structure is present in the hash function or the input space, quantum preimage attacks may reduce the effective complexity to the cubic-root of the original search space. This means that instead of requiring 2^n operations to find a preimage, a quantum attacker could achieve success in approximately $2^{n/3}$ operations, resulting in a far more significant reduction in security. Such capabilities have serious implications for cryptographic hash functions used in authentication and integrity verification, as security assumptions based on classical brute-force complexity no longer hold. To remain resilient in the post-quantum era, systems relying on hash-based security must choose larger output sizes or adopt new designs that avoid structural weaknesses that quantum algorithms can exploit.

3.2 NIST's PQC Standardization

Considering upcoming quantum computers and their potentials to jeopardize current state-of-the-art cryptographic schemes, the National Institute of Standards and

Technology (NIST) has initiated a program to develop and standardize public-key cryptographic algorithms resistant to quantum attacks [14]. As part of the process, NIST determined the requirements, submission criteria, and evaluation methods for candidate algorithms. All comments and summaries of revisions based on community feedback are available at NIST's website[1]. The call for submissions aimed to identify algorithms suitable for both key establishment and digital signature applications, with a final submission deadline.

During the third round in 2020, NIST announced seven finalist algorithms and eight alternative candidates. Finalists continued to be considered for standardization, while alternate candidates were evaluated for their security confidence, performance, potential improvements, and suitability for specific use cases. Some alternative candidates, despite the lower performance, were retained because of their high confidence in security, whereas others required additional analysis to validate their robustness.

After the end of the third round, in 2022, NIST selected four quantum-safe algorithms for standardization. In particular, the selected are CRYSTALS-Kyber for the key encapsulation mechanism (KEM) and CRYSTALS-Dilithium, FALCON, and SPHINCS+ for digital signatures. Additionally, in 2025, the hamming quasi-cyclic (HQC) scheme was also standardized as quantum-resistant KEM. These are currently the selected tools to secure communications in the era of quantum computing but simultaneously await further evaluation upon the fourth-round submitted algorithms in order to enrich the quantum-resistant pool.

3.3 Categories of Post-Quantum Cryptographic Algorithms

Besides that most of the standards are based on lattices, the candidates of NIST's contest for standardizing quantum-safe cryptographic schemes belong to several families based on the mathematical problems where their strength is relied. In this context, several families incorporate PQC and their corresponding standard schemes are summarized as follows:

1. **Lattices:** Algorithms belonging in this class are based on the presumed difficulty of lattice problems [15], such as the shortest vector problem (SVP). It is highly difficult to calculate the vector B_x, depicted in Fig. 3.2, as a linear combination of two vectors b_1 and b_2 that are denoted as the basis. They are among the most studied PQC approaches, yet standardized, and are known for offering efficient implementations and strong security reductions. Examples include the aforementioned CRYSTALS-Kyber [16] KEM and the signature schemes CRYSTALS-Dilithium [17] and Falcon [18].

[1] https://www.nist.gov/pqcrypto

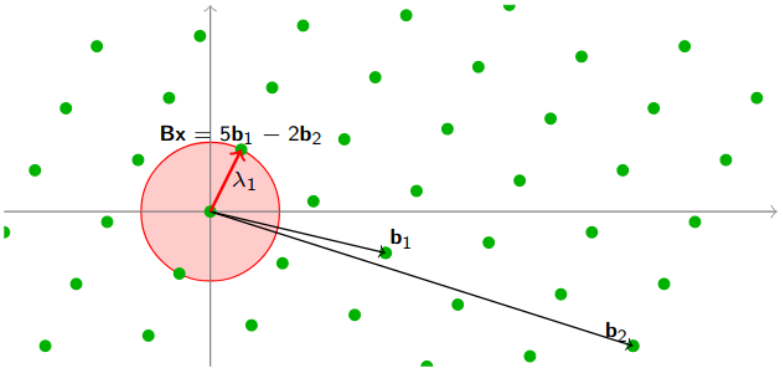

Figure 3.2: The difficulty to express a vector as a linear combination of two, too close base vectors.

2. **Code-Based:** These algorithms are built on the difficulty of decoding a general linear error-correcting code [19]. The first notable candidate that has been selected as standard is bit flip key encapsulation (BIKE) [21] KEM, using quasi-cyclic codes. Later, the prominent ode-based algorithm named HQC [22] KEM is the most recent entry into the pool of quantum-resistant standards. Finally, the McEliece cryptosystem [20], proposed in 1978, remains one of the most prominent examples due to its long-standing resistance to classical and quantum attacks. One of its drawbacks remains that it requires large public keys. The classical-McEliece edition incorporates the strength of binary Goppa codes and is currently being evaluated to be the next code-based standard.

3. **Hash-Based:** These signature schemes derive their security directly from the properties of cryptographic hash functions. They offer strong security assurances under minimal assumptions. Classic examples include the Merkle signature scheme, as well as more advanced constructions such as SPHINCS+ [23] , which provide stateless and scalable designs. In the next chapters, there will be a thorough analysis of this kind of algorithms, such as SHAKE256.

4. **Multivariate Quadratic:** This family [24] exploits the hardness of solving systems of multivariate quadratic equations over finite fields, a problem known to be NP-hard. Multivariate schemes are often efficient for signing, but can face challenges in terms of key size or verification performance. The Rainbow digital signature was a prominent candidate for the NIST's contest but was broken in 2022, leading to its withdrawal.

5. **Isogeny-Based:** It is another cryptographic scheme based on the family of elliptic curves. However, in this case, the rationale is based on the presumed difficulty of finding isogenies ϕ between elliptic curves [25], illustrated in Fig. 3.3. These cryptographic schemes provide relatively small key sizes compared to other PQC approaches. However, they remain computationally slower and less

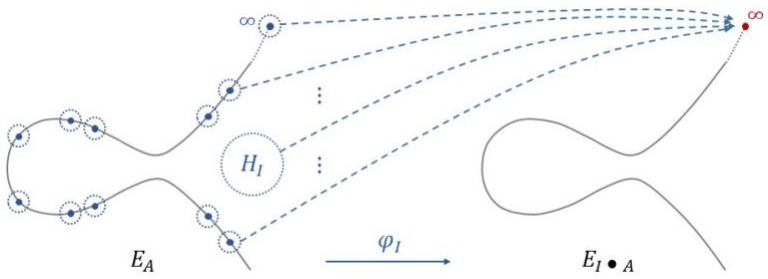

Figure 3.3: Isogenies between two elliptic curves.

mature. A well-known example is the supersingular isogeny key encapsulation *SIKE* protocol, though it has since been broken by classical attacks.

Each family represents a distinct set of assumptions and trade-offs in terms of key size, computational cost, and security confidence. Together, they provide a diverse foundation for developing standards that can withstand the future threat posed by quantum computers.

CHAPTER 4

Attacks Against PQC Schemes

Each of the aforementioned post-quantum cryptography types is based on different mathematical problems. Hence, strategies to perform cryptanalysis against these systems require one to act individually. In particular, different strategies might be required even between algorithms of the same family. So, while PQC schemes are designed to withstand attacks from quantum computers, they are not immune to exploitation. Instead of directly breaking the underlying hard mathematical problems, attackers can target the implementation layer, aiming at side-channel leakage, fault injection, and protocol misuse. Many lattice-based schemes, for example, reveal subtle information through power consumption, electromagnetic emissions, or timing variations during key operations. Code-based and hash-based constructions can also be vulnerable if randomness is poorly generated or if decapsulation errors disclose partial secret key data. Additionally, hybrid deployments, where classical and quantum-safe algorithms coexist, can introduce weaknesses in key management and downgrade paths. These challenges highlight that the security of PQC depends not only on strong mathematical foundations but also on careful engineering, hardened implementations, and continuous evaluation against emerging attack techniques.

4.1 General Attacks

4.1.1 Lattice-Based Cryptosystems

In the aforementioned NIST's contest for PQC standardization, lattice-based cryptography gained a large area, and consequently, its security strength has been under long-term research assessment to eliminate any potential vulnerabilities. As a result, significant progress has been made in improving the asymptotical and practical efficiency of SVP and lattice reduction algorithms. Based on SVP and lattice reduction models, some generic cryptanalysis methodologies with extensive experimental verifications were presented. In particular, solving the *learning with errors* (LWE) problem [26], is on average at least as difficult as solving SVP, in the worst case scenario. In this context, the SVP leaves inside the LWE. In other words, matrix A is the lattice, generated by the basis vector such as the aforementioned b_1 and b_2, where $s \equiv B_x$, a linear

combination of b_1, b_2. More precisely, an attacker is given the ciphertext of a sample c, of the form:

$$c = As + e \mod q$$

where $A \in \mathbb{Z}_q^{m \times n}$ is a public matrix, $s \in \mathbb{Z}_{\|}{}^n$ is a secret vector, and e is the error vector that is small in size and is generated from a discrete Gaussian distribution and is also secret. Cryptanalytic attacks typically attempt to recover s by either reducing the problem to a bounded distance decoding instance or by transforming the LWE samples into a lattice basis and applying lattice reduction algorithms.

Another prominent attack strategy is the primal lattice attack, where the adversary constructs a lattice whose basis embeds the public matrix A and the modulus q, and then searches for a short vector corresponding to the error vector e. The success of this approach relies on achieving a sufficiently small approximation factor in the SVP.

4.1.2 Code-Based Cryptosystems

Code-based cryptographic schemes derive their security from the hardness of the *syndrome decoding problem*, which can be stated as follows: given a parity-check matrix $H \in \mathbb{F}_2^{(n-k)\times n}$, a syndrome $s = He^\top$, and a weight bound w, the goal is to recover an error vector $e \in \mathbb{F}_2^n$ such that $weight(e) = e$. This problem is NP-hard in general and serves as the foundation for the security of schemes such as McEliece and BIKE. The most effective known attacks are variants of Information Set Decoding (ISD) [27], which attempt to guess a subset of positions assumed to be error-free, thereby reducing the decoding problem to a smaller, more tractable instance. The success probability of ISD depends on the binomial distribution of errors, and its expected complexity is typically expressed as

$$T_{ISD} \approx \frac{\binom{n}{w}}{\binom{k}{w}}, \tag{4.1}$$

up to polynomial factors, highlighting the exponential dependence on both the code length and the error weight.

Modern ISD algorithms refine this basic idea through meet-in-the-middle strategies, partial Gaussian elimination, and representation techniques that trade memory for time. Algorithms such as Stern-type and BJMM-style attacks decompose the error vector into structured components and exploit collisions in intermediate sums to accelerate decoding. For structured code-based schemes, additional algebraic properties, such as quasi-cyclic or quasi-dyadic constructions, may further reduce attack complexity by introducing symmetries that effectively lower the search space. Consequently, the concrete security of code-based systems is highly sensitive to parameter choices, particularly the ratio w/n and the dimension n. While quantum speedups for ISD remain limited to generic amplitude amplification, careful parameterization is required to ensure that the exponential gap between the best known decoding attacks and feasible computation remains sufficiently large.

From a complexity-theoretic perspective, the impact of quantum computation on code-based cryptography is significantly more limited than in other post-quantum families. The best-known quantum speedups for ISD [28] rely on augmentation, which provides at most a quadratic improvement over classical exhaustive search. Thus, Eq. 4.1 becomes

$$T_{Quantum-ISD} \approx \sqrt{\frac{\binom{n}{w}}{\binom{k}{w}}}$$

Unlike lattice attacks, no quantum algorithms are known that fundamentally alter the exponential structure of syndrome decoding. As a result, code-based schemes achieve quantum resistance primarily through conservative parameter choices that ensure even the square-root reduction remains computationally infeasible.

4.1.3 Multivariate Cryptosystems

The ultimate objective of cryptanalysis is to retrieve the original message from its encrypted form, which in multivariate systems translates to solving a multivariate quadratic equation set over a finite field. Although this problem has proven to be NP-hard, meaning that it is computationally intractable in the general case, several practical strategies have emerged that can exploit structural or implementation weaknesses. Thus, there are several approaches targeting multivariate public-key cryptography [29]. The ultimate purpose is to reduce the complexity of recovering the hidden private key or plaintext. Among all, the most significant are:

Structure attacks exploit algebraic properties such as hidden structure in the public key, leading to the recovery of the private key more efficiently than brute force. One of the most prominent structural attacks mentioned is the MinRank attack. Through MinRank or similar algebraic techniques, attackers can often find a lower-rank representation that reveals secret relationships between polynomials, allowing full key recovery.

Differential attacks leverage the response of the cryptographic scheme to carefully chosen inputs to extract information about the secret key or break the one-wayness of the system. These are based on analyzing how small changes in input affect the output to gradually recover secret parameters.

Implementation and parameter is the careless deployment of a multivariate scheme and can make it vulnerable. Many of the attacks on this kind cryptographic systems are not augmented by quantum techniques but on classical algebraic or heuristic techniques.

4.1.4 Isogeny-Based Cryptosystems

Recall Ch. 3, related to cryptographic schemes based on isogeny. Theses rely on the presumed hardness of computing isogenies between supersingular elliptic curves,

a problem believed to resist both classical and quantum subexponential attacks. Given two supersingular elliptic curves E_1 and E_2 defined over $\mathbb{F}_{p^2}$, the underlying hard problem is to find an isogeny $\phi : E_1 \to E_2$ of prescribed degree. Mathematical attacks typically model this task as a path-finding problem in the supersingular isogeny graph, where vertices correspond to isomorphism classes of curves and edges represent low-degree isogenies. Classical attacks based on random walks or meet-in-the-middle techniques have complexity on the order of $O(p^{1/2})$, while quantum algorithms such as claw-finding reduce this to $O(p^{1/3})$, still exponential in the bit-length of p. Consequently,isogeny-based constructions must select large primes to maintain adequate security margins, particularly in the presence of quantum adversaries [30].

4.2 Attacks Based On Side Channel Analysis (SCA)

Side-channel analysis (SCA) [31] refers to a class of attacks that exploit unintended information leakage produced during the execution of cryptographic algorithms. Rather than targeting solely on mathematical weaknesses in the algorithm itself, as explained in the previous chapter, SCA exploits observable characteristics such as power consumption, electromagnetic emissions, or execution timing to infer sensitive internal states. These leakages arise from the physical implementation of a system and are often influenced by hardware architecture, signal integrity, and environmental noise. As a result, implementations that are theoretically secure can still be vulnerable when deployed on real devices. Evaluating and mitigating side-channel leakage is therefore essential, particularly for embedded and resource-constrained platforms, where limited countermeasures and noisy measurement conditions can significantly affect the security of cryptographic operations.

For example, *timing side-channel* attacks [33] exploit variations in execution time that depend on secret data or control flow. Even minor differences caused by conditional branches, memory accesses, or early termination conditions can leak information about internal states. In cryptographic implementations, non-constant-time operations may therefore enable an adversary to infer secret-dependent behavior through repeated measurements. On the other hand, *simple power analysis* (SPA) [34] relies on direct visual or statistical inspection of power consumption traces to identify patterns associated with cryptographic operations. Distinct instruction sequences or data-dependent operations may produce recognizable power signatures, allowing attackers to recover partial or complete secret information without complex post-processing.

Information Set Decoding augmented by Side-Channel Analysis (ISD+SCA) [32] combines algorithmic decoding techniques with physical leakage exploitation to significantly reduce the practical complexity of code-based cryptanalysis. Although classical ISD treats all candidate information sets as equally likely, side-channel observations such as power consumption, timing variations, or electromagnetic emissions can introduce measurable biases that guide the selection or pruning of candidate subsets. By correlating leakage patterns with intermediate decoding operations, an at-

tacker can prioritize high-probability information sets, and thereby shrink the effective search space without modifying the underlying code structure. This hybrid attack model highlights that implementation-level weaknesses can undermine parameter choices that are otherwise secure in the black-box setting, emphasizing the need of side-channel–aware security evaluations for code-based cryptographic implementations. For example, pure ISD against Classic-McEliece-460896, which is NIST's security level-3, corresponds to the 2^{192} security level, a number that until now is not feasible to be processed even with quantum computers. In that case, future cryptanalysis may alter the attack route, utilize side channel traces, and reveal secret values, even without the augmentation of quantum computation. Recall Ch. 4.1.2 where mentioned that quantum computers may enhance the ISD algorithms providing them perspective to be more efficient attacking code based systems. Then, there is the availability of the above three distinct attacks to be able to combine their strength with already pure mathematics, resulting efficient attacks against code based algorithms, such as McEleice.

4.2.1 Experimental SCA Against KEMs

Hardware Setup

In the context of SPA, this thesis attempts to correlate leakages with recovered power traces while specific hardware executes cryptographic operations. In general, SPA requires relatively modest experimental equipment compared to more advanced side-channel techniques, as it relies on direct observation of power consumption patterns rather than extensive statistical processing. Essential tools include a target device that performs the cryptographic operation, a stable power supply, and a means of measuring instantaneous power consumption, such as a shunt resistor or current probe coupled with a high-bandwidth digital oscilloscope. A trigger mechanism is often used to align measurements with specific execution phases, while basic data acquisition software enables visualization and manual inspection of traces. Unlike differential techniques, SPA generally does not require large trace sets or complex analysis frameworks, making it accessible yet effective against implementations with clearly distinguishable power signatures.

In line with this rationale, the equipment that was recruited to participate in such an experiment, acting analogously to a real SPA equipment, was the following:

1. A raspberry pi 3 that acts as a device under test (DUT), also mentioned as the target device. This device is the hardware platform on which the target cryptographic algorithm is implemented and evaluated. DUT executes the cryptographic operations while exposing physical characteristics, such as power consumption or electromagnetic emissions, that may unintentionally leak information about internal computations. In side-channel experiments, the DUT is typically instrumented to allow controlled execution, repeatable measurements, and precise triggering without altering its functional behavior.

2. A shunt resistor, which is a low-resistance component inserted in series with the power supply of the DUT to enable indirect measurement of its instantaneous current consumption. By observing the voltage drop across the shunt during cryptographic execution, fine-grained power traces can be captured that reflect data-dependent switching activity within the device. Proper selection of the shunt value balances measurement sensitivity and minimal impact on normal operation, making it a fundamental and non-invasive element in practical power-based side-channel analysis setups.

3. An operational amplifier (op-amp) is used to amplify the small voltage drop across the shunt resistor to a level suitable for accurate acquisition by measurement instruments. Configured with appropriate gain and bandwidth, the op-amp enhances signal visibility while preserving the temporal characteristics of power consumption variations. Careful design of the amplification stage is essential to minimize noise, avoid signal distortion, and ensure that the amplification process does not introduce measurement artifacts.

4. A Raspberry Pi Pico, which is a low-cost microcontroller platform used to translate op-amp voltage variations by shunt, monitoring the cryptographic operations, in the logic of a related application[1], acting like the probes that covert electromagnetic fields, generated by hardware operations, to power spikes, analogously to bitwise operations.

[1]https://docs.edgeimpulse.com/projects/expert-network/collect-data-keyword-spotting-raspberry-pi-pico

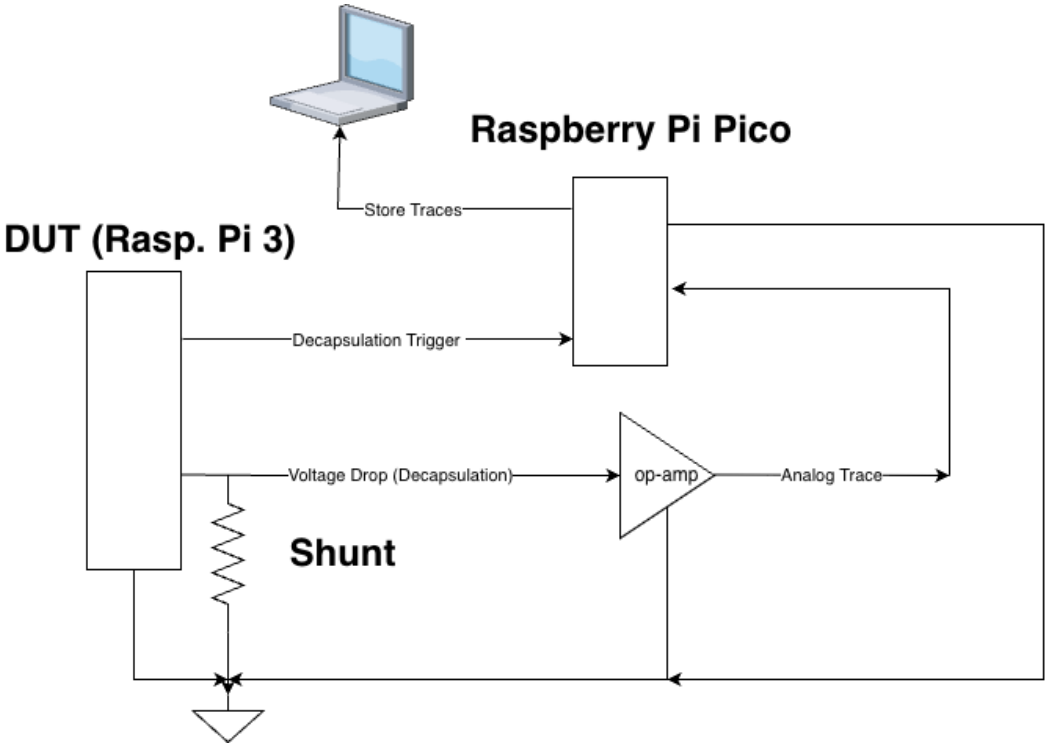

Figure 4.1: Pi Pico collects amplified power traces when DUT executes PQC operations, populatig machine learning models.

Having this set up, it is possible to create an artificial, yet realistic, SCA experiment. The setup is schematically drawn as per Fig. 4.1 and is represented in Fig. 4.2. The ultimate problem for this kind of cryptanalysis is the special hardware that is either expensive or difficult to purchase. Such an equipment is a digital oscilloscope or an appropriate probe that can translate electromagnetic radiation, generated by DUT's processor during operations, to voltage spikes that oscilloscope can save. Instead of these, a shunt, an op-amp, and a micro-controller are combined to substitute the oscilloscope and the probe. This act has a significant trade-off such as low voltage sampling resolution and a noisy environment. This drawback narrowed the number of PQC algorithms that this set up was able to be studied. The lattice-based systems, by default, perform better and faster than the code-based systems. Thus, this equipment combination, with respect to the lattices, either cannot distinguish the noise from the cryptographic operations or cannot stop sampling when normally these algorithms terminate in a few milliseconds. However, the power traces collected from the code-based systems, BIKE and McEliece, are able to reveal patterns, helping ML models predict secret bits.

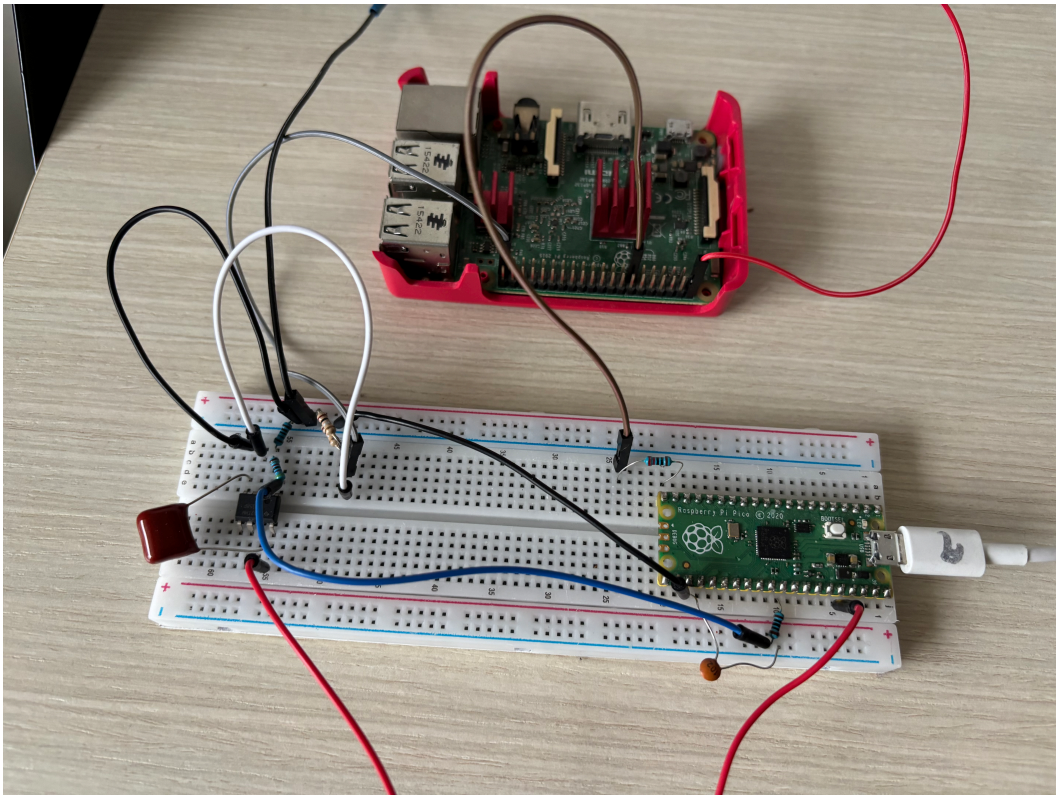

Figure 4.2: Side channel analysis setup. The raspberry Pi and Pico, and a breadboard containing the shunt and the op-amp.

4.2.2 Evaluating Results

The primary objective of power analysis is to evaluate whether a cryptographic implementation unintentionally leaks sensitive information through its power consumption behavior during execution. By observing and analyzing power traces, an attacker or evaluator aims to identify correlations between measured signals and internal operations, intermediate values, or secret parameters. This experiment used the Python wrapper of the open quantum safe (OQS) open source library where the code based KEMs, McEliece and BIKE, both for the NIST's security level-3. During several decapsulation operations, Python code A′.1.1 saved a csv file that contains: a byte of the shared secret key, its hamming weight, and also the execution time for each decapsulation. The secret byte would be used as ML labels, in order to train models, where the role of the features played the power traces captured by the Pico micro-controller, with the help of the script in Appx. A′.1.4. For both algorithms, the number of captured traces is 250, all of them using the same public-private key pair.

McEliece

The normal distribution of Fig. 4.3 and the waveforms of the power traces of Fig. 4.4 reveal that all the observed operations run without having significant variations, with respect to the execution time. In other words, the decapsulation waveforms and

execution time are well aligned, allowing the ML models to separate high from low processing workload. These physical variations manifest themselves as distinct features in the recorded waveforms, allowing an observer to associate specific signal patterns with underlying secret-driven computations. In McEliece decapsulation, the sequence of operations and the overall computational workload are well defined, particularly toward the final stages, where error correction and key derivation are completed.

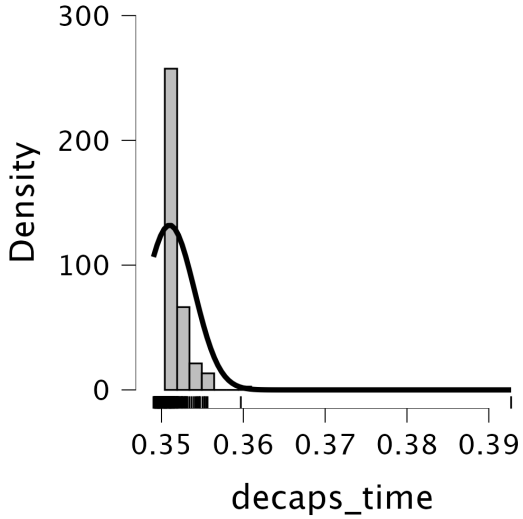

Figure 4.3: Normal distribution of the McEliece decapsulation time, during more than 250 repetitions.

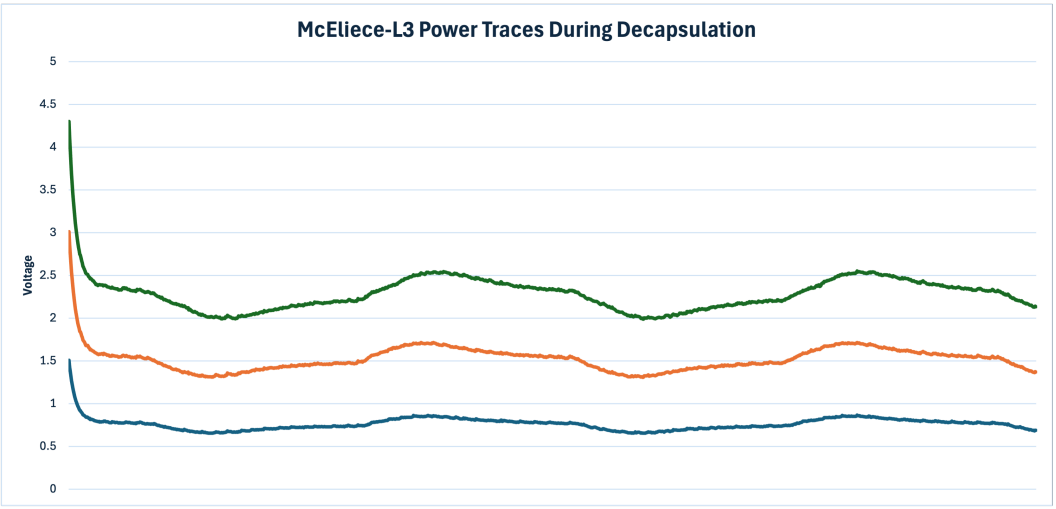

Figure 4.4: Three random McEliece power traces waveforms, during decapsulation.

The correlation coefficient plays a critical role in power analysis by quantifying the statistical relationship between measured power traces and hypothesized internal

values or operations. A high correlation indicates that variations in the observed waveform are consistently aligned with the predicted leakage model, suggesting the presence of secret-dependent information. In contrast, low correlation values imply weak or absent leakage, making the correlation coefficient a key metric for evaluating both attack success and the effectiveness of side-channel countermeasures. Fig. 4.2.2, exported by the python script as per Appx. A′.1.2, illustrate the correlation coefficients between the measured power traces and different bit-width hypotheses. This bit sequence belongs to the first byte of the McEliece shared secret produced during the final stage of decapsulation. Note that, according to [35], the correlation values around 0.1 reveal significant leakage. This strategy helps the ML models to become more efficient by attacking the windows which have increased correlation. In this experiment, the number of samples is not too large, allowing the models to process the total number of samples per trace.

Byte0 denotes the first byte of the shared secret in memory order. However, in McEliece decapsulation, the shared secret is typically generated only after the decoding phase completes, which means that byte0 is computed and processed during the final stage of execution. Although it is the first byte of the secret, its leakage appears near the end of the decapsulation timeline, which explains its consistent visibility in late-stage side-channel measurements.

So, distinct correlation peaks appear at consistent sample positions across the plots, indicating time intervals where secret-dependent operations occur. As the number of jointly modeled bits increases, the correlation structure becomes more pronounced but also more distributed, reflecting the aggregation of leakage from multiple dependent operations. These results demonstrate that the end phase of McEliece decapsulation exhibits measurable and temporally aligned power leakage that can be exploited through correlation-based side-channel analysis.

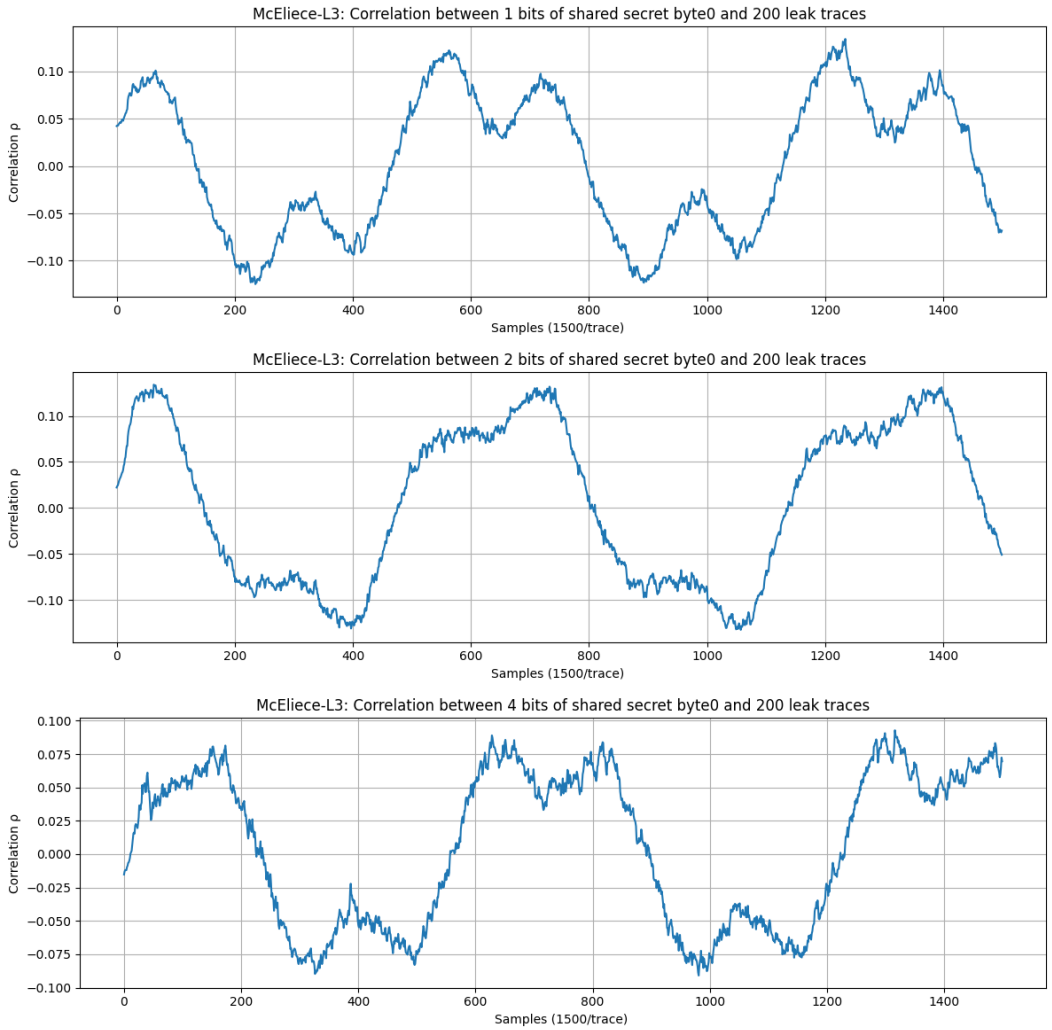

Figure 4.5: Correlation between voltage traces and secret key leakage during McEliece-L3 decapsulations.

Classifier	Class bit 0 F1	Class bit 1 F1	Accuracy
SVM-RBF	0.59	0.65	0.63
kNN	0.57	0.67	0.63
Linear SVM	0.47	0.68	0.60

Table 4.1: A single bit prediction using only 200 McEliece-L3 decapsulation traces.

Tab. 4.2.2 hosts the performance of different classifiers to predict a single bit of the McEliece-L3 shared secret using only 200 power traces. All models achieve accuracies above random guessing, indicating the presence of exploitable leakage even with a limited number of measurements. The support vector machine (SVM) with an radial basis function (RBF) kernel and k-neural networks (kNN) show comparable overall accuracy, while the linear SVM exhibits more imbalanced class-wise performance, suggest-

ing that non-linear decision boundaries are better suited to capture the structure of the side-channel leakage in this setting. The F1-score provides a balanced measure of precision and recall, making it particularly suitable for evaluating bit-level predictions in side-channel analysis. The consistently higher F1-scores observed for the bit 1 class , compared to the bit 0 class, indicates that the leakage associated with bit 1 is more distinctive and easier for the classifiers to identify. This asymmetry suggests that secret-dependent operations produce stronger or more consistent power signatures when processing a logical '1', revealing an inherent imbalance in the physical leakage characteristics of the implementation.

BIKE

Working in the same manner for BIKE-L3, Fig. 4.6 indicates that power traces can also reveal notable leakage, which may be processed by models in order to predict secret values. In this case, as shown in Fig. 4.7, the traces are perfectly aligned, leading to the hypothesis that an attacker can easily extract the significant bits that match them to cryptographic operations during decapsulation.

In particular, using the same script used for the calculation of the McEliece correlation, the BIKE's correlation between voltage traces and the leakage bits, the results overcome the threshold of the value 0.15, where in accordance with the [35] indicates significant leakage (byte0). Consequently, running ML models to calculate the accuracy of predicting bits of byte0, the results in Tab. 4.2.2 again dictate the prediction above from just raw guessing, as well.

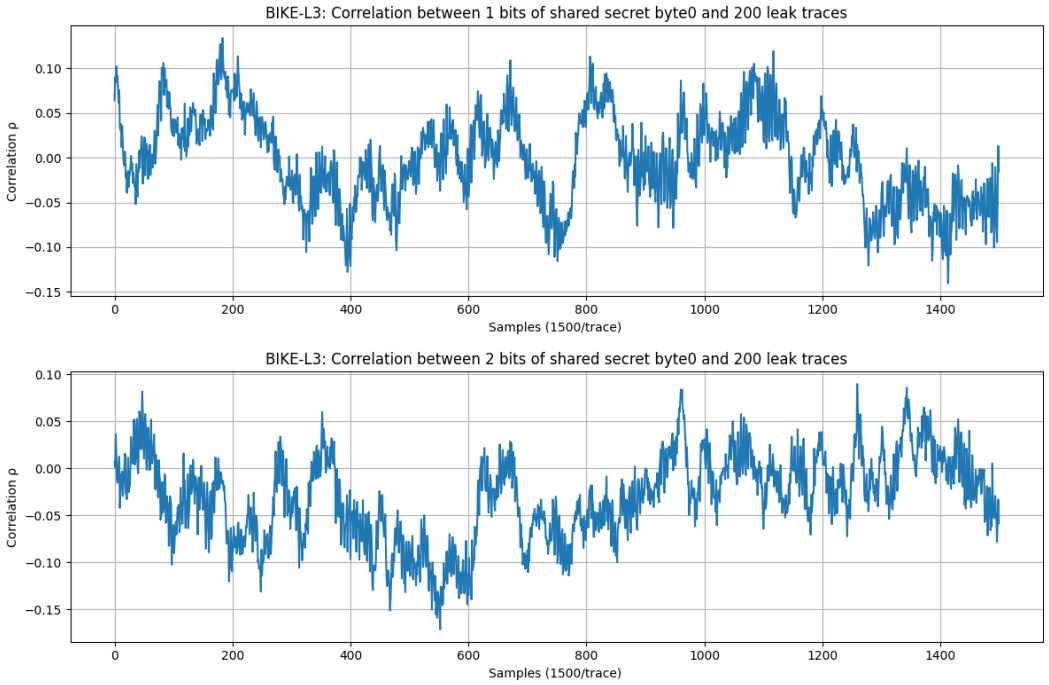

Figure 4.6: Correlation between voltage traces and secret key leakage during BIKE-L3 decapsulations.

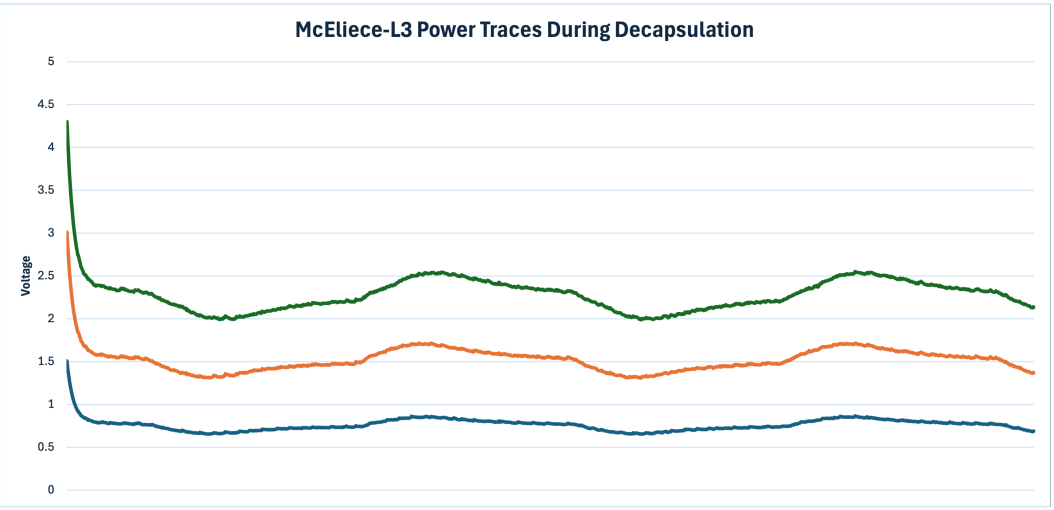

Figure 4.7: Correlation between voltage traces and secret key leakage during BIKE-L3 decapsulations.

Classifier	Class bit 0 F1	Class bit 1 F1	Accuracy
Logistic Regression	0.58	0.62	0.60
Random Forest	0.53	0.65	0.60
Linear SVM	0.60	0.60	0.60

Table 4.2: A single bit prediction using only 200 BIKE-L3 decapsulation traces.

Observing the classifiers' results (accuracy & F1), for both of the PQC KEMs, we are able to deduct that the McEliece cryptosystem appears to be more predictable. Indeed, comparing the power waveforms, we notice that McEliece acts in a more nervous manner, and this behavior may allow the models to extract better knowledge about hidden values.

4.2.3 SCA Against Signatures (Falcon)

Falcon is a lattice-based digital signature scheme built on the NTRU (another lattice-based public-key cryptographic scheme) framework and relies on fast Fourier sampling to generate short signatures. Its security is grounded in the hardness of lattice problems, while its design prioritizes compact signature sizes and high performance. However, the use of Gaussian sampling and floating-point or fixed-point arithmetic introduces implementation complexity, which makes Falcon particularly sensitive to side-channel and implementation-level leakage. As a result, practical evaluations of Falcon often focus not only on its cryptographic strength, but also on the leakage behavior of its sampling and arithmetic components under real hardware constraints.

In [36], the correlation results indicate that successful side-channel exploitation of Falcon is possible even in the presence of very weak linear leakage. Fig. 4.8 shows

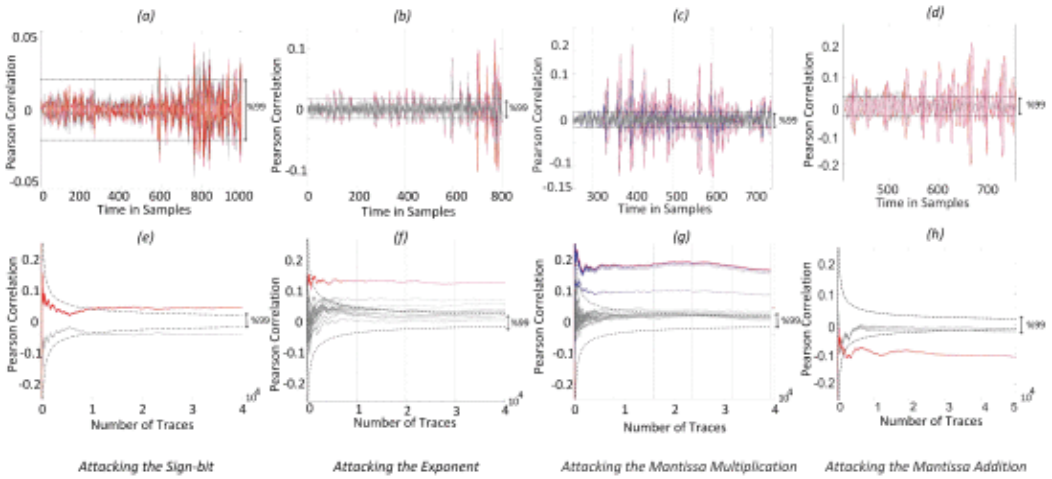

Figure 4.8: Power traces and leakage correlation for the PQC Falcon signature.

that targeting the sign bit of floating-point coefficients, the correct hypothesis exhibits correlation values on the order of 0.04, requiring several thousand traces to reach statistical significance. In contrast, attacks on the exponent achieve higher correlation peaks of approximately 0.10, enabling reliable discrimination with roughly a thousand measurements. The mantissa operations demonstrate the strongest leakage, with correlation values reaching up to 0.20. Although mantissa multiplications initially produce multiple false positives due to similar leakage behavior, subsequent attacks focusing on mantissa additions effectively eliminate these ambiguities and isolate the correct hypothesis. In general, the results confirm that Falcon implementations can leak exploitable information at correlation levels well below 0.1, provided that sufficient measurements are available and appropriate statistical tools are applied.

4.2.4 Countermeasures Against SCA

Algorithmic countermeasures aim to reduce side channel leakage by eliminating secret-dependent control flow and data access patterns within cryptographic implementations. Techniques such as *constant-time* execution [37], avoidance of conditional branches based on secret values, and uniform memory access help ensure that operations exhibit consistent physical behavior. These measures are particularly important in decoding-based schemes, where early exits or variable iteration counts can otherwise introduce strong leakage signatures.

Masking countermeasures [38], [39] introduce randomness into intermediate values by combining secrets with random masks, thus decorrelating the processed data from the true secret. When properly implemented, masking reduces the effectiveness of statistical attacks, such as correlation power analysis, by requiring attackers to combine multiple leakage sources. Complementary randomization techniques, including execution order shuffling and noise injection, further obscure deterministic leakage

patterns without altering cryptographic functionality.

Hardware-level countermeasures address leakage in the physical layer by modifying the electrical behavior of the device. Examples include balanced logic styles, power-line filtering, voltage regulation, and dedicated noise sources that reduce signal-to-noise ratios in measured traces. However, such techniques can significantly increase the cost and complexity of side-channel attacks by being most effective when combined with algorithmic and masking approaches to provide defense in depth.

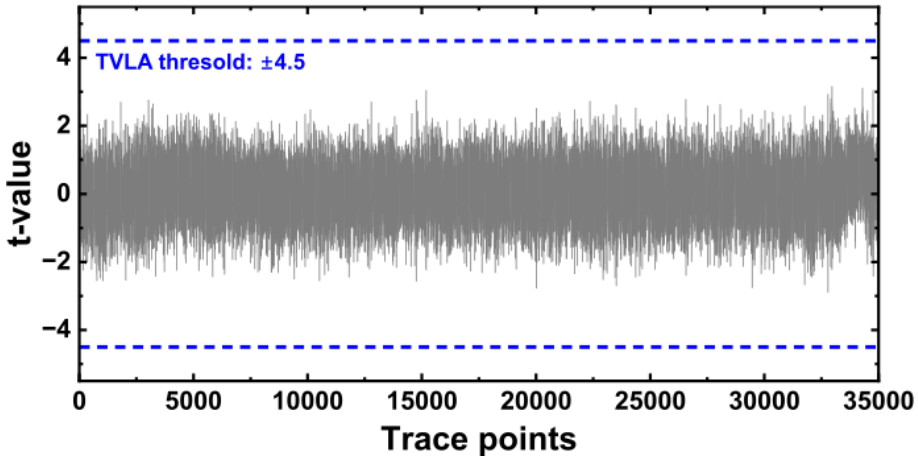

Figure 4.9: Hardware McEliece Implementation Leakage Analysis

In relation, the authors in [40] utilize the test vector leakage analysis (TVLA) to evaluate potential leakage. The t-values computed, at all trace points, result in leak detection to overcome the threshold of ±4.5 in Fig. 4.9 . Since the observed t-values remain well within these bounds throughout the measurement window, no statistically significant first-order leakage is detected. This result suggests that the applied countermeasures effectively reduce exploitable power leakage under the tested conditions, providing evidence of resistance against basic statistical side-channel analysis. Note that both the correlation coefficient r used in this thesis and the t-values t of the aforementioned paper are related to the following formula:

$$t = r\sqrt{\frac{n-2}{1-r^2}}$$

where n is the number of samples.

CHAPTER 5

Applications Adopting PQC

In our modern life software and hardware application play a key role in every day life and by default leveraging its quality. In this context, the digitization of methods and tools, we already have been using an ”analog” manner, demand stability, availability, confidentiality, and integrity. In the analog era, human is the key factor for authenticity, whereas in digital life this role has been taken over by the machines. In the following chapters, we can go through several aspects of the modern world where cryptography is the major ingredient that ensures their proper functionality.

5.1 Blockchain Databases

5.1.1 Blockchain Background

Blockchain technology [41] became popular and known after it was utilized in cryptocurrencies, such as Bitcoin, introduced by Satoshi Nakamoto in 2008. Bitcoin was the first electronic payment system without third party intervention using decentralized and distributed peer-to-peer (P2P) networks. The terms ”Block” and ”Chain” were initially used separately by the founder Nakamoto.

Although digital wallets are not part of the foundational design of blockchains, they represent an indispensable tool for ensuring the practical usability of cryptocurrencies. A wallet enables users to generate and manage public–private key pairs, maintain corresponding addresses, record transaction histories, and initiate new transactions. In general, wallets are categorized into two types: hot storage and cold storage. Hot wallets, which remain continuously connected to the Internet, store frequently used keys and thereby facilitate rapid and convenient transactions. However, this persistent connectivity increases susceptibility to security breaches, such as key theft. In contrast, cold wallets store keys offline, which substantially reduces the risk of unauthorized access, but also limits convenience due to their inaccessibility in real time.

The operational model of a cryptocurrency, such as Bitcoin, follows a systematic process. To perform transactions, an individual must first install wallet software on a computing device, such as a personal computer or smartphone. This software incorporates not only wallet functionalities but also essential blockchain components, typ-

ically retaining only block headers. Subsequently, the user acquires cryptocurrency, either through an exchange platform or through P2P transfer from an existing holder. The acquired currency is then deposited into the digital wallet. For smaller transaction amounts, the currency becomes spendable once the deposit is verified. In contrast, larger amounts generally require confirmation through their inclusion in a newly validated block before they can be securely transacted.

In blockchain, each block contains a set of transactions that are digitally signed by its "verifier" and stored on the distributed network so that all legitimate stakeholders can access and verify them. Attributes of a blockchain, such as decentralization, immutability, audit, transparency, and security, offer various benefits to different domains such as cryptocurrency, financial sectors, private and public segments, insurance, healthcare, supply chain management, the Internet of Things (IoT), etc. Blockchains are typically managed by a peer-to-peer (P2P) computer network for use as a public distributed ledger, where nodes collectively adhere to a consensus algorithm protocol to add and validate new transaction blocks. In addition, such records are considered secure by design with high fault tolerance.

This technology provides various benefits, which a significant one is **transparency**. Transactions stored on the blockchain are transparent to all participants. Blockchain uses the distributed ledger (a shared copy of a document) kept by individual parties and can only be updated by the consensus mechanism, which means that the file can only be updated if all legitimate parties agree to do so.

There are many ways in which the blockchain is more **secure** than the other record management systems. Transactions are added after consensus by all permitted parties. Once everyone agrees on the transaction, it is encrypted and securely linked to the previous block. Secured hashing mechanisms attached to each block are used to secure the blocks that hold the number of transactions. Hence, it is practically infeasible to alter a block because it requires modifications to other blocks in the chain, also.

Data tracking is easy within a blockchain. Transactions are visible to all parties, leading to traceability for any operation, which permits **traceability**. Furthermore, in traditional systems, the paperwork is time-consuming, tedious, and prone to human errors. So, by automating it with blockchain, the process becomes **faster and more efficient** and operates without third-party intervention. Finally, regarding the benefits of blockchain, for any business, profit versus cost-effectiveness is important. Using this technology, there is no further requirement for any intermediary or third party entity. Hence, it becomes **cost-effective**.

Blockchain, in a simple word, is a technology that provides access to any verified data control over the decentralized environment to every participant node in a fast and convenient way. There is no single or centralized authority to validate or verify the nodes. In order for a new node to participate in a network, it has to validate itself by solving a mathematical puzzle called a *proof of work* [42]. The node that succeeds in a proof-of-work can introduce a block. The action in which new data must be validated and become content of a blockchain is initiated by a node and called *transaction*.

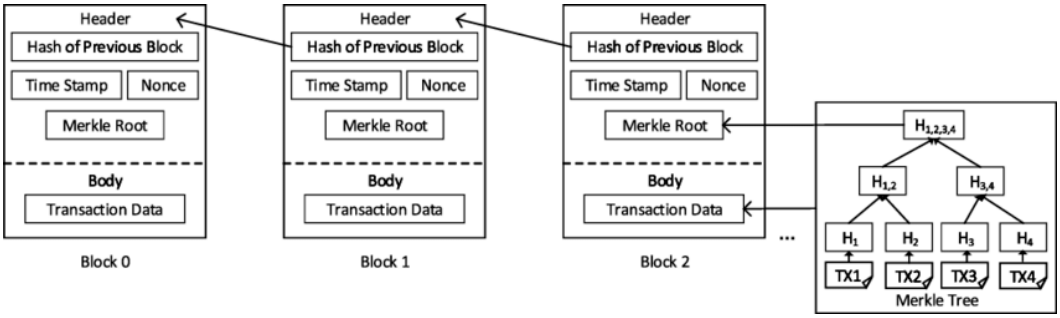

Figure 5.1: The structure of a blockchain.

Blockchains are separated into two types. The *public*, which are less safe due to the lack of restrictions implemented and simultaneously slow because they are open to anyone who wants to become a node. On the other hand, there are *private* blockchains that are fast and supposed to be safe. Safety though is a matter of assumptions we make because whenever a provider is private, there are few ways to investigate its mechanisms.

5.1.2 Blockchain Building Blocks

The blockchain represents a transaction ledger structured as a linked list of blocks, beginning with the genesis block, ensuring that all transactions are strictly ordered and unambiguous. In an open P2P network, a robust consensus mechanism is essential to guaranty that all nodes maintain identical copies of the blockchain, and thereby preserving the integrity and usefulness of the ledger. In this context, in order to establish a symmetric shape along the ledger, there are key parts and functionalities required to do so, depicted in Fig.5.1. In particular, few transactions are grouped to form a **block**, and the action where a chunk of data that will be stored on the blockchain is called **transaction**. Finally, process where these transactions are verified is called **transaction verification process**.

The transaction verification sequence involves several key steps: initiation, broadcast, validation, and confirmation. First, a user generates a transaction, signs it with his private key, and sends it to all nodes on the network. Then, nodes on the network decipher with user's public key and validate the transaction, checking for things like sufficient funds and proper formatting. Finally, the validated transaction is included in a new block, which is added to the blockchain, making the transaction permanent. A more detailed analysis of the transaction procedure is referred to [43] and is listed below, representing the operations required to accomplish it. These functions incorporate several cryptographic primitives, which will be the subject of further analysis in the following paragraphs. These primitives comprise *hash functions* and *public key cryptography*, which were analyzed in previous chapters, but also *Merkle trees*, which will be analyzed in the following lines.

The transaction procedure serves a pivotal function within the consensus mech-

anism by arranging and assembling them into blocks prior to distribution to peers for validation. To guaranty both the *integrity* and the *authenticity* of these blocks, the nodes employ cryptographic signatures as part of the block creation process. In detail, this procedure contains the following steps:

Transaction batching. Ordering nodes collect transaction requests from clients and group them into blocks. The grouping process follows predefined parameters, such as the maximum block size or specified time intervals.

Block formation. Each block is structured into three parts, the header, data, and metadata. The header contains essential identifiers, including the block number, the hash of the previous block, and the hash of the current data. The data section records the transactions, while the metadata carries supplementary details, such as digital signatures and subsequent validation information.

Hash computation. Before the block is signed, the ordering node calculates a cryptographic hash on the data and metadata of the block. This hash, embedded in the header, uniquely represents the block's content.

Digital signing. The ordering node signs the block header with its private key. The signed elements include the block number, the hash of the previous block, and the computed data hash. This operation guaranties the authenticity and integrity of the block, confirming that it originates from a legitimate ordering authority.

Signature integration. The generated signature is stored within the block metadata. This addition allows peers to later verify the origin and validity of the block.

Block distribution. Once signed, the block is transmitted to all peers in the network. Each peer can verify the ordering signature of the node using its corresponding public key, thereby ensuring that the block is genuine and tamper-free.

In particular, after the signed block distribution, the peers perform a two-step verification procedure. First, they validate the signature using the ordering node public key and confirm that the header values and the data hash are consistent with the actual contents of the block. Second, the transactions within the block are checked against the endorsement policies and evaluated for conflicts using a method that provides concurrent access, called multi-version concurrency control. Only transactions that meet these conditions are committed to the ledger and are reflected in the world state. The validation results are then recorded in the block metadata, which specifies the status of each transaction. This mechanism guaranties that all peers share a synchronized and verifiable view of transaction validity across the network.

5.1.3 Traditional SQL vs NoSQL Databases

The concept of a relational database was defined by E. F. Codd at IBM in 1970 [45]. A database system typically comprises a data store that includes both external storage

and main memory, a user interface that enables interactions such as submitting queries and receiving responses, and a query processor. The query processor is responsible for interpreting and executing user queries before returning the results. In practice, query processors are highly sophisticated software components that play a fundamental role in the overall functioning of database systems. In particular, the structure query language (SQL) offers a variety of functions that give the user the ability to extract a lot of data stored in linked tables. This relation prevents identical data from being repeated in each new tuple. Instead, repetitive data are stored once in separate linked tables and recalled each time a tuple must be generated. In this case, relational databases are more flexible and have a better potential to combine data implementing the fundamental properties of the relational database theory. In summary, SQL databases have been developed to manage structured information and provide support for scaling across multiple machines.

On the other hand, non-relational (NoSQL) databases store data using non-table formats, offering flexible, horizontally scalable, and high-performance solutions for handling large volumes of rapidly changing data, unstructured data, and big data applications. Unlike traditional SQL databases with fixed rows and columns, NoSQL databases use structures such as documents, key-value pairs, or graphs, making them ideal for modern applications in the cloud, mobile, social media, and big data.

The decision to adopt SQL or NoSQL solutions is heavily dependent on organizational requirements, but identifying the most suitable option can often be complex. The key distinction lies in how each paradigm structures and manages data. Many NoSQL systems also adopt hybrid data models, which complicates the migration between cloud service providers [46].

Several studies often focus on comparisons between traditional relational database management systems (RDBMS), such as Oracle, and document-oriented NoSQL solutions like MongoDB [47]. These comparisons typically evaluate scalability, performance, consistency, availability, and sharding strategies. The literature suggests that while SQL databases remain more suitable for online transaction processing, NoSQL technologies, with their flexible and domain-specific designs, are often more effective for big data analytics.

The distinction between SQL and NoSQL databases extends beyond the simple categorization of relational versus non-relational systems. A major point of comparison lies in how each manages transactions. In general, a database transaction represents a sequence of operations executed as a single unit of work. SQL databases traditionally adhere to the *atomicity*, *consistency*, *isolation*, and *durability*, namely the (ACID) model. These principles ensure reliable handling of transactional data. In particular, **atomicity** ensures that each transaction is treated as an indivisible unit: it either completes entirely or has no effect at all. This eliminates the risk of partial updates that could leave data in an inconsistent state. Consistency enforces that all database rules and integrity constraints remain valid throughout a transaction. When a transaction finishes, the system must transition from one valid state to another, preventing illegal or

contradictory data. **Isolation** dictates that concurrent transactions operate as if they were executed sequentially. Temporary states produced during a transaction cannot be observed by others, minimizing interference and race conditions. Finally, **durability** guaranties that committed data persist even in the presence of failures such as power loss or system crashes. Together, these properties form a robust framework for systems that require strict correctness and reliability, particularly in financial and critical information environments.

Many NoSQL developers considered the strict enforcement of ACID principles to be a limiting factor when dealing with massive, distributed datasets. A well-known result in distributed computing, Brewer's *consistency*, *availability*, and *partition-tolerance*, consisting the (CAP) theorem [48]. This architecture formalizes the inherent trade-offs in system design. The theorem states that a distributed system cannot simultaneously guaranty consistency, availability, and partition tolerance. At most, two of these properties can be achieved at the same time. Partition tolerance refers to the ability of the system to continue functioning correctly in the presence of partial network failures, where some nodes may become unreachable but the system as a whole remains operational.

5.1.4 Modern Blockchain Database Models

Blockchain databases operate on an append-only log of transactions, typically grouped into blocks that are *cryptographically* linked. The state of the system is derived by sequential execution of these transactions rather than through table-based relations. Most blockchain platforms employ key–value storage structures rather than relational schemes [49]. As a result, they do not inherently support relational operations such as multi-table joins or referential integrity constraints.

Moreover, while relational databases are based on the ACID model, blockchain databases generally adopt weaker guaranties, closer to eventual consistency due to distributed consensus protocols [50]. Data integrity in blockchains is enforced by cryptography and consensus, rather than by schema rules or foreign-key dependencies. Consequently, blockchain databases are optimized for immutability, transparency, and distributed trust, but not for the expressive relational querying offered by traditional RDBMS and SQL. This distinction explains why relational databases are widely used for complex information systems requiring rich queries, while blockchain databases are more suitable for environments where audit, tamper-resistance, and decentralized control are primary concerns [51]. However, there are several reasons generated by the evolving technology that eventually enforce the mitigation of using traditional databases compared to modern distributed ones.

More particularly, recently studied *big data* introduces significant challenges in terms of storage, analysis, and high speed data delivery, as these tasks require substantial computational and storage resources. Efficient processing and analysis of large-scale datasets require extensive infrastructure, which can be difficult to provision and maintain. Cloud computing, through its shared resource model, offers a practical solu-

tion to these challenges by providing on-demand access to computing power, storage, and networking capabilities. This integration enables big data systems not only to store, process, and distribute information efficiently, but also to facilitate seamless access across different geographical regions. Furthermore, the growth of social networking platforms that use cloud infrastructures has become a major contributor to the generation of big data [52].

Consequently, with regard to big data, the primary objective is to facilitate efficient utilization of distributed data blocks to support various data processing, allowing large-scale analysis, and eventually producing more big data. As a result, there is a recent model, called BASE [53], that relies on its foundations in the following principles: 1) **basically available**, where according to the CAP theorem, this property ensures that the system always provides some form of response to a request. However, the response may indicate a failure to retrieve the desired data, or the returned data may be incomplete or in a transitional state, 2) **soft state** system, which is not a fixed one and may evolve over time, even in the absence of new inputs, 3) **eventual consistency** which guaranties that once input activity ceases, the system will ultimately converge to a consistent state. Updates will propagate throughout the system, ensuring eventual synchronization across all nodes. In practice, the system often continues to process new input without enforcing strict consistency checks at each step.

5.1.5 Centralized and Decentralized Blockchain Databases

Centralized database architectures play an important role in private blockchain systems, where data management is governed by a single authority or a consortium of trusted entities. These systems follow a hierarchical structure in which a central controller oversees the updates, access, and maintenance of the data. This arrangement offers advantages in terms of efficiency, as decisions can be made rapidly without the need for agreement across a large number of distributed nodes. As a result, centralized solutions are often associated with faster transaction processing and improved scalability, making them well suited for applications that require high performance in terms of latency.

Enterprise blockchains are common examples of centralized blockchains, where access is restricted to authorized participants who are allowed to validate and update the ledger. The controlled nature of these environments generally facilitates better performance compared to public blockchains, where every node must participate in consensus. Centralized oversight also simplifies administrative tasks, enabling easier deployment of updates and system changes. This makes centralized databases particularly attractive in domains that value efficiency and rapid decision-making.

However, centralized database solutions also introduce notable challenges. A key concern is the security risk due to the single control point. This fact creates potentials of single point of failure. An adversary who successfully compromises this authority could jeopardize the entire system. Moreover, the absence of full transparency in centralized architectures may be a key factor towards implementing a blockchain.

Sai et al. [54] map the various layers where centralization can emerge within supposedly decentralized public blockchains. It categorizes centralization risks in areas such as consensus control, governance, mining pools, and network topology. The authors argue that decentralization should be quantitatively evaluated across these dimensions instead of assumed from design intent alone.

In addition, Homoliak and Szalachowski [55] introduce *aquareum*, a ledger system controlled by a central authority, but reinforced with blockchain principles and trusted computing hardware. Their model preserves auditability and immutability without requiring full decentralization, making it appropriate for regulated environments that must maintain operational oversight.

On the other hand, *decentralized* database architectures [56] form the backbone of many blockchain systems. In more particular, public blockchains such as Bitcoin and Ethereum are a representative paradigm of such an architecture. Unlike centralized models, the data in these systems are replicated across multiple nodes, with each node maintaining a complete copy of the ledger. Additionally, the absence of a central authority enhances both security and resilience because any modification to the ledger must be verified and approved by the majority of network participants. This makes unauthorized alterations extremely difficult. In Tab. 5.1, there is an effort in order to quote the most significant properties regarding the comparison between the centralized and decentralized architectures and how these affect several aspects of a ledger.

Atzori [57] analyzes how blockchain systems aim to replace conventional trust institutions with a distributed mechanism of governance. The study differentiates between decentralization as a technical property (through distributed consensus) and as a social-political aspiration (autonomous governance without intermediaries). The paper highlights both benefits and governance challenges that arise when authority is spread over a peer-to-peer network.

It is obvious that with respect to overall performance the centralized database systems act more effectively but may face security and transparency issues. On the contrary, decentralization incorporates, on one hand, cryptographic primitives, on the other hand, implements more democratic principles eliminating the single point failure and setting it more robust and trustful. In this context, scientists managed to combine architecture leveraging the advantages of each and, hence, developing the called *hybrid* models.

5.1.6 Hybrid Models

Different business environments often present distinct requirements, leading to the development of hybrid blockchain database systems [58], customized for specific application scenarios. This chapter examines existing hybrid blockchain database models and provides a brief overview of related systems that can also be categorized as ledger databases. In summary, hybrid blockchain databases can be characterized as decentralized architectures composed of three fundamental components: a *shared data-*

Aspect	*Centralized Database*	*Decentralized Database*
Control	Single authority	All nodes share control
Performance	Fast, efficient(consensus)	Slower(no consensus)
Scalability	Easier to scale	Easy to add nodes
Transparency	Limited	High transparency
Security Risks	Single point of failure	Compromising multiple nodes
Use Cases	Enterprise environments	Distributed applications

Table 5.1: A comparison between centralized and decentralized database systems.

base, implemented with storage engines such as MySQL, MongoDB, etc. A *shared ledger*, maintained and replicated through consensus mechanisms such as Kafka [59], [60], and a *user interface*, which typically offers basic operations.

Integrating blockchain technology with traditional databases has led to the development of hybrid systems that combine the strengths of both approaches. These systems aim to provide the security, transparency, and immutability of blockchains alongside the efficiency, scalability, and complex querying capabilities of conventional databases. Several real-world applications utilize this hybrid model. For example, the mBridge[1] is a collaborative project involving central banks from Hong Kong, Thailand, the United Arab Emirates, China, and the Bank for International Settlements. This platform utilizes a blockchain-based ledger to facilitate real-time peer-to-peer cross-border payments and foreign exchange transactions using CBDCs. The system ensures compliance with specific regulations while leveraging blockchain's transparency and security features.

The DataStax[2] is a company specializing in database management, which launched Astra Block in February 2023. This cloud-based service integrates the Ethereum blockchain with DataStax's database platform, and Astra DB, to support the development of Web3 applications. By streaming data from the Ethereum blockchain into a scalable database environment, Astra Block uses a hybrid system that *merges blockchain data with traditional database capabilities*.

Xu et al. [61] designed a hybrid blockchain framework to protect communication and data exchange in multi-domain avionics systems. The proposed fabric combines blockchain control with decentralized trust mechanisms to achieve data integrity and provenance between the aircraft subsystems that collaborate. It demonstrates the applicability of blockchain technology to aerospace contexts where full decentralization is impractical.

In [62] a thorough analysis of the reviewed studies reveals several key motivations behind the integration of blockchain technology with IoT systems. A significant proportion of the work emphasizes the need to improve overall system security. Many approaches focus on maintaining the integrity and confidentiality of data collected by

[1]https://en.wikipedia.org/wiki/MBridge
[2]https://en.wikipedia.org/wiki/DataStax

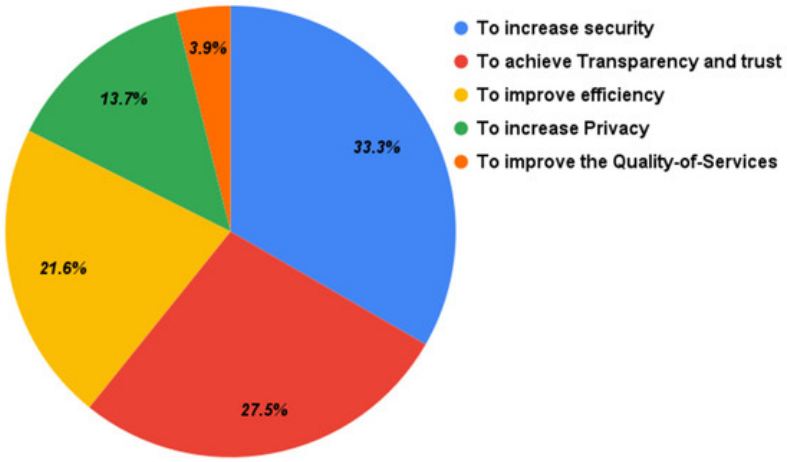

Figure 5.2: Reasons that companies and organizations use hybrid models.

IoT devices, as well as ensuring the availability of systems that operate without centralized control points vulnerable to disruption. In certain cases, blockchain has also been applied to safeguard communication channels, such as protecting data transmitted by unmanned aerial vehicles against plaintext or ciphertext attacks. Since data stored on the blockchain cannot be altered or deleted, this immutability ensures a reliable record of transactions and enhances traceability across applications such as supply chain management and food safety monitoring. Furthermore, the decentralized structure of blockchain enables open access to stored information, promoting accountability among participants. Efficiency also plays an important role, with blockchain-based smart contracts reducing communication delays between devices, lowering operational costs, optimizing energy consumption, minimizing latency, and improving throughput. Privacy considerations form another major motivation, as blockchain allows users to interact through public keys without revealing personal identities, thereby protecting sensitive information.

These examples, along with the motivations to adopt hybrid models, highlight the practical applications of hybrid blockchain-database. Fig. 5.2 shows how much various industries use the combined technologies to address specific business needs by leveraging the unique advantages of each.

Taking into consideration the aforementioned data, it is obvious that the blockchain cannot substitute traditional databases. Thus, to overcome the limitations of blockchain while leveraging its benefits, many organizations adopt a hybrid approach. In such systems, blockchain is used to record critical and immutable data (e.g., financial transactions, digital identities, and supply chain records), while traditional databases manage high-speed transactions and allow flexible data querying. This combination provides a balance between security, efficiency, and scalability, making it a practical solution for enterprise applications.

5.1.7 Cryptography in Blockchain

Related to the building blocks of the blockchain architecture, the hash functions and public-key cryptography play a fundamental role of a blockchain's operations. For example, in the block formation, each block is structured into three parts, the header, data, and metadata. The header contains essential identifiers, including the block number,the **hash of the previous block**, and the **hash of the current data**.

5.1.7.1 Merkle Tree and Hash Functions

The Merkle trees [63],[64], [65] introduced by Ralph Merkle represent a cornerstone in both computer science and cryptography, with significant importance in blockchain systems. These hierarchical tree structures provide an efficient mechanism for summarizing and validating large collections of data, such as transaction records in blockchains.

In a binary Merkle tree, each leaf node stores the blockchain's transaction data, while every internal non-leaf node stores the cryptographic hash of the concatenation of its two children, as depicted in Fig. 5.3. Given transaction data T_i, T_j and a hash function h, their hashed values are $H_i = h(T_i)$ and $H_j = h(T_j)$, respectively, and their parent hash is $H_{ij} = h(H_i \parallel H_j)$. This recursive structure ensures fast and secure verification of large data collections.

The Merkle path is defined as the ordered sequence of hashes $H = \{H_1, H_2, \ldots, H_k\}$ that links a leaf node T_i to the root r. This path is used to confirm that a specific data block is included in the tree. A *Merkle Proof* consists of three components: the data block txn_i, its corresponding Merkle path $H = \{H_1, H_2, \ldots, H_k\}$, and the Merkle root r. To validate the proof, one begins by hashing T_i, then iteratively concatenates and hashes it with each element of the Merkle path, finally checking whether the resulting hash equals the root r. In Fig. 5.4 is represented such a procedure. There, in order to prove that the orange transaction $T3$ has taken place, the hash functions $h4$, $h7$ and $h12$ must be computed to finally reach the root.

The key ingredients of the Merkle tree can be summarized as follows:

Efficient Verification. With only the root hash r, the presence of a specific block b_i can be verified without traversing the entire dataset.

Tamper Resistance. Any modification in a block b_i changes its leaf hash, which propagates upward and alters the root r, thus revealing data tampering.

Compactness. Instead of storing all blocks $b_1, b_2, \ldots, b_n$, the dataset can be summarized into a single cryptographic value r, significantly reducing storage requirements and allowing efficient proof transmission.

In conclusion, Merkle trees serve as an essential building block in blockchain architectures, ensuring secure and efficient verification of data integrity. In other words, these represent in a compact way all transactions within a block. Each transaction acts

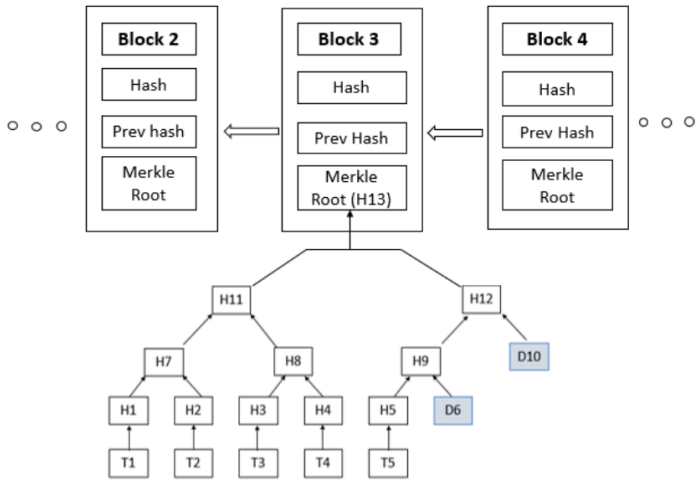

Figure 5.3: The Merkle tree.

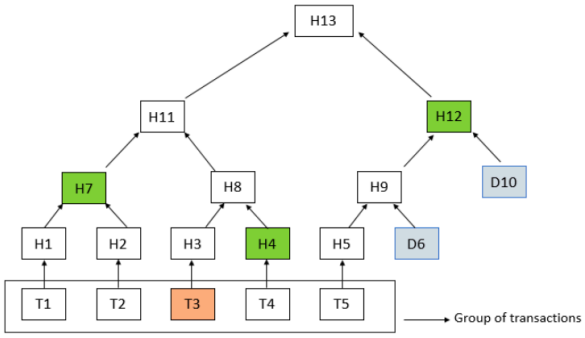

Figure 5.4: The Merkle tree proof path.

as a leaf node, allowing users to *confirm* the inclusion of a transaction without downloading the entire blockchain.

The Secure Hash Algorithm (SHA) family represents one of the most widely adopted primitives in modern cryptography, particularly for ensuring digital signature authentication. The first version of SHA was introduced by the National Institute of Standards and Technology (NIST) in 1991, and subsequent versions were formally standardized to establish a consistent framework for hash functions [67]. In the next chapters, a comprehensive presentation of the hash functions, along with a sufficient evaluation with respect to security and performance, will take place.

5.1.7.2 Digital Signatures

In the previous chapters, there was a brief reference on digital signatures and the way they are used in digital communications, but in this section there is a more thor-

ough analysis of the procedures using such a technology. Recall then that a digital signature is a cryptographic scheme that is used to verify the authenticity and integrity of digital messages or documents. This technology relies on public-key cryptography acting as a digital equivalent of a handwritten signature, ensuring that the message or document originated from a specific sender and has not been altered since it was signed. In order for a sender to prove that he ends a particular message m, the following sequence of actions must be executed:

1. The sender generates a pair of cryptographic keys: a private key $PrKey$ used for signing and a public key $PubKey$, used for verification.
2. He uses a hash function h to create a digest $d = h(m)$, and the private key to encrypt d creating a unique digital signature $\sigma = Enc(PrKey, d)$.
3. Sends m and σ to the recipient.
4. The receiver: computes $d' = h(m)$, decrypts with $PubKey$ the σ to get d, thus $d = Decr(PubKey, s)$ and if $d' = d$, then verifies that the message is authentic (owns the sender).

The Digital Signature Algorithm (DSA) employs asymmetric encryption (public key) to achieve identity authentication, data integrity verification, and tamper resistance. In blockchain systems, every transaction must be validated with a digital signature before being added to the block. However, blockchain faces increasing challenges in handling large-scale transaction volumes, particularly in electronic cash applications. With the rapid expansion of the digital cash market, the number and scale of transaction data have increased. To ensure real-time processing, it is necessary to verify multiple signatures simultaneously. However, this large-scale signature verification process introduces significant computational overhead for blockchain nodes [68].

5.1.8 PQC in Blockchain

Although blockchain technology is known for its decentralized, transparent, and immutable characteristics, it is not totally tamper-proof to systematic attacks, introducing several vulnerabilities. Recognizing this kind of issue is crucial for the development and deployment of secure blockchain systems. In other words, blockchain networks can face challenges related to data availability, where users may experience abnormal or incorrect access to blockchain data. Such issues can arise from network congestion, denial-of-service attacks, or inadequate network resources, affecting the overall reliability of the blockchain system.

Despite the pseudonymous nature of blockchain transactions [66], sophisticated analysis techniques can potentially reveal user names, leading to privacy breaches. Attackers may employ methods such as transaction links and disclose user identities, compromising confidentiality. Consequently, there is a lot of effort to improve the security of blockchain technology. In [69], the authors propose a novel approach for an

intrusion detection system based on collaboration to mitigate attacks. In this context, the paper introduces a trust-based blockchain framework for collaborative intrusion detection networks (CIDNs), known as *trust-chain*. This system aims to secure the integrity of the information exchanged between CIDN peers, strengthen accountability, and protect cooperative operations by mitigating insider threats.

Furthermore, in [70] the authors outline several strategies to strengthen the security of blockchain systems. These strategies involve the application of cryptographic primitives to ensure confidentiality and authentication, the adoption of resilient consensus protocols to strengthen resistance against tampering, and the integration of network-level protections to reduce exposure to external threats.

Then, following the potential vulnerabilities of the blockchain, the next paragraphs aim to highlight trends with respect to the security primitives of the blockchain. In addition to the classical cryptography, the computational strength of the upcoming quantum computers is also taken into account, where traditional and well established cryptographic algorithms will become obsolete, as also analyzed in Ch. 3, and consequently, the cryptographic foundations of the blockchain architecture will eventually be affected.

5.1.8.1 Quantum-Safe Digital Signatures

Currently, ECDSA remains one of the most widely used signature algorithms in the blockchain [75] because it has some advantages, compared to the RSA cryptographic system, namely shorter key sizes. Nevertheless, its implementation poses two major mathematical challenges. First, the modular inversion operation is inherently slow, taking nearly ten times longer than multiplication. Second, elliptic curve scalar multiplication, which is required in both the signature generation and verification phases, dominates computational cost and directly impacts the overall efficiency of ECDSA. To address these bottlenecks, several optimization schemes have been proposed. A fast variant ECDSA scheme relies only on scalar multiplication and modular multiplication, enhancing computational speed. However, this approach trades-off security, as reducing the number of inverse operations may expose the scheme to forgery risks and weaken forward security. To overcome such vulnerabilities, there are other ECDSA variants without modular inversion, which incorporate another technology to improve efficiency while maintaining robust security.

On the other hand, the analyzed in Ch. 3 Peter Shor's quantum algorithm can effectively execute integer factorization and discrete logarithms, and consequently can attack the RSA and ECDSA cryptographic systems. To overcome this obstacle, blockchain systems can integrate PQC signatures to ensure long-term security against adversaries equipped with quantum computers. By replacing these schemes with quantum-resistant signature algorithms, such as lattice-based, hash-based, or multivariate signatures, blockchains can preserve the authenticity and non-repudiation of transactions in a post-quantum era. PQC signatures enable participants to verify transaction ownership and block validity without relying on mathematically fragile assumptions, thereby

extending the security lifespan of decentralized ledgers. Although PQC signatures often introduce larger key sizes and higher computational overhead, their adoption can be managed through protocol-level optimizations, hybrid signature approaches, or selective deployment in critical consensus and governance mechanisms.

5.1.8.2 Quantum-Safe Hash Functions

The early generations of hash standards were based on Merkle-Damgard construction, in which an input message is partitioned into blocks, each processed by a compression function to produce a fixed-length output [71]. Hash functions such as MD4, MD5, SHA-1, and SHA-2 follow this design principle, although each employs different compression mechanisms and internal parameters.

In the literature, there are many studies proposing several hash-based cryptographic schemes, each of them targeting from a certain point of view. In [67] an improved variant of SHA-1 and SHA-2 is proposed that resists collision and length extension attacks. Their contribution strengthens the robustness of these hash functions against these particular cryptographic threats. However, their study does not explicitly examine how such vulnerabilities might impact the operation of Merkle trees in blockchain-based systems, which this study deals with.

With the increasing feasibility of collision attacks, NIST launched a public competition to identify a next-generation hash standard. After three rounds of evaluation, *Keccak* [72] emerged as the winner and was officially adopted as the SHA-3 [73] standard. Unlike the Merkle-Damgard family, Keccak uses *sponge construction* that operates in two distinct phases: *absorb* and *squeeze*. In addition to producing fixed-length digests (224, 256, 384, and 512 bits) similar to SHA-2, Keccak introduced extendable output functions, namely SHAKE-128 and SHAKE-256, providing greater flexibility for variable-length hash output. SHA-3 was selected and standardized by NIST through an open cryptographic competition and formally specified in *Federal Information Processing Standard Publication 202* (FIPS 202) as an alternative to the SHA-2 family. By now and with respect to PQC, this family is recognized for its pivotal role, receiving further support. For example, [74] implements the hardware design of the complete FIPS 202 specification to ensure compatibility with a broad range of cryptographic schemes that depend on SHA-3–based hash functions. This approach allows the accelerator to be reused across multiple algorithmic contexts while maintaining practical assumptions regarding its integration within a System-on-Chip (SoC) environment. The evaluation includes comprehensive performance metrics, reporting silicon area, operating frequency, and execution latency in clock cycles for several implementations.

This evolution of hash cryptographic primitives provides significant improvements, allowing them to be effectively implemented in modern applications such as the blockchain. This rationale could be further supported while modern hash cryptography can be assumed as quantum-safe. In this context, practical test-beds of well-known blockchains such as bitcoin should evaluate their performance in order to assess the future availability, allowing the continuation of their service in the post-quantum era.

5.1.8.3 PQC Blockchain Performance

When a state-of-the-art technological system is being built, the concept of shielding it from cyber-attacks is essential. Although cryptography is fundamental in blockchain, there are several provisions that must be taken into account in order blockchain retain a robust and secure setup, as analyzed in Ch. 3. In brief, these provisions are related to the enormous computational strength of *quantum computers*. Although there are not many sophisticated quantum algorithms that can efficiently compromise classic cryptographic schemes that are based on hard mathematical problems rather than Shor's and Grover's, in the near future current cryptographic systems will be considered obsolete affecting directly blockchain technology. In this case, there should be a transition period towards replacing the current cryptographic algorithms with those that are standardized as quantum safe.

To do so, several studies have been taken place in order to evaluate blockchain with respect to the state-of-the-art quantum resistant cryptographic algorithms. Brotsis et al. [77] delved deeply into such an analysis regarding the performance of the blockchain. Performance evaluations in this area have been conducted across diverse hardware platforms and networking protocols, often under varying assumptions regarding the characteristics of the communication channel.

Furthermore, in order to quantify and measure the overhead that may possibly add post-quantum cryptography at the blockchain, first we should isolate primitives that are correlated with cryptographic operations. For example, operations such as signing and verifying take place and are enclosed in a significant ingredient of the blockchain, the transactions. In addition, another cryptographic operation that may add overhead is blockchain, is hashing, which takes place into the Merkle tree. In this context, the following benchmark aims to assess these operations with respect to the domains of time and space, and breaking down the signature schemes at their sub-parts.

Signatures

To study the overhead mentioned , the forked tool[3] was recruited to benchmark PQC algorithms against classical algorithms, comparing key sizes, signature sizes, and performance metrics. The key features of this tool allow for a comparison between PQC and classical digital signature algorithms with respect to key generation, signing, and verification timing. In addition, it offers multiple input sizes that allow for the study of the overhead as a function of the input data size. The results produced are depicted in Fig. 5.5 where the performance of the algorithms is plotted on logarithmic scale. Thus, in terms of the time needed by an algorithm to sign the transaction data, we can clearly see that the hash-based scheme *XMSS* has the worst performance, in all types of data size. Furthermore, *CRYSTALS-Dilithium* and *Falcon* remark notable performance compared to current cryptographic schemes such as RSA and ECDSA. In addition, CRYSTALS-Dilithium thrives compared also to its post-quantum equivalents, Falcon-512 and SPHINCS+-128f/s. In this case, and after several years of studies trying to address CRYSTALS-Dilithium's vulnerabilities, it could be appropri-

[3] https://github.com/konnnGit/pqc-benchmark-f

ate to implement this post-quantum signature scheme into current environments like blockchain. Note also that a Bitcoin transaction's size varies, but a single-input, single-output transaction is typically around 191-192 bytes. As per level 3 security, we are able to deduct that RSA-4096 and SPHINCS+-192f/s perform in a less satisfactory time whereas CRYSTALS-Dilithium and Falcon-1024 act in a similar time considering one more that these lattice-base cryptographic schemes may serve critical infrastructures, setting simultaneously the transition to the post-quantum era smoothly.

Regarding the time it takes for a transaction to be verified, Fig. 5.5 shows that CRYSTALS-Dilithium variants perform faster compared to both levels of security. Moreover, another notable metric is the performance of pre-quantum schemes and post-quantum XMSS. In this case, the latter spent more time to verify a signed transaction of a blockchain peer, and hence it could not be a suitable candidate to shield blockchain technology against quantum attacks.

In the same rationale and with more realistic manner, the Python web3 library was used to simulate the cryptographic operations of Ethereum. The primary goal of another forked tool[4] is to benchmark and compare traditional cryptographic algorithms with post-quantum alternatives in the context of local blockchain operations. Specifically, the project evaluates the performance of classical algorithms, like those mentioned in the previous paragraphs, against the lattice-based digital signature algorithm CRYSTALS-Dilithium. The methodology of the project comprises benchmarking the raw cryptographic operations with respect to key generation, sign, and verify for the domain of time and size.

```
--- Ethereum Standard Simulation (ECDSA-secp256k1 via web3) ---

[Ethereum Address Generation (ECDSA-secp256k1)]
Public Address: 0x351F7385Aa54F725ABcD70E3E6e0f806D71A8481
Private Key: af70e3f3877c81902cc79ed84e539cbccfe8f7a61c6395de6e8fff5571a5a2c3

[Ethereum Transaction Signing (ECDSA-secp256k1)]
Transaction Data: 'transfer(10000000000000000000, 0x8461278E68e6200034babEFc1f17787fDeeB5198)'
Transaction Hash (Keccak-256): 0x2203ac467986916505c82fa77eefbc8ebf49e766d8f9ee3d35e3145df9204331
Signature size: 65 bytes
Signing Time: 3.9273 ms

[Ethereum Transaction Verification (ECDSA-secp256k1)]
Recovered Address: 0x351F7385Aa54F725ABcD70E3E6e0f806D71A8481
Expected Address: 0x351F7385Aa54F725ABcD70E3E6e0f806D71A8481
Verification Time: 10.5003 ms
Signature Valid: True
```

Figure 5.6: Ethereum operations insights using ECDSA as signature algorithm.

Hash functions

In relation to the hashes used in a Merkle tree, there are some trivial benchmarks to quantify the overhead of each algorithm. Hence, the **build time** evaluates the impact of

[4]https://github.com/konnnGit/PQC-Blockchain-Benchmarking-and-Testing-Fork/tree/main

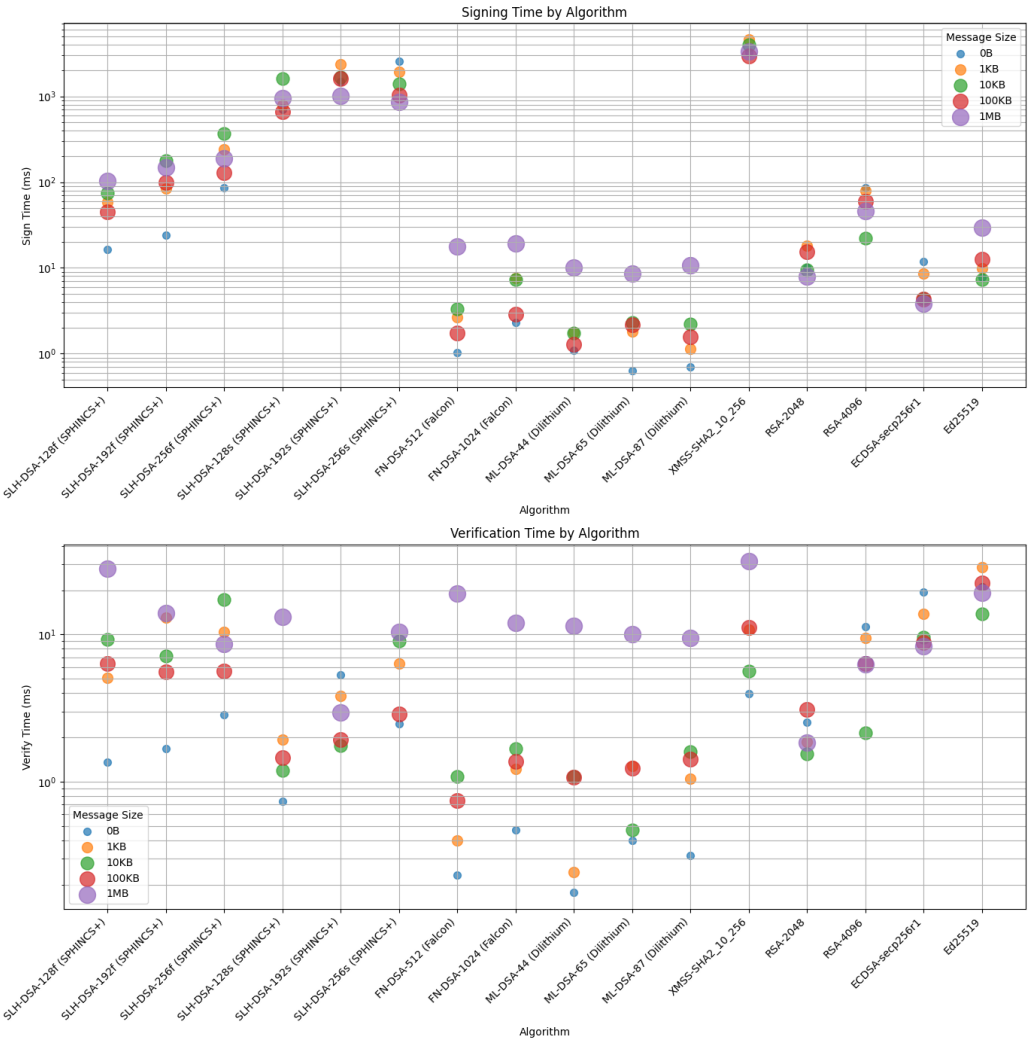

Figure 5.5: Signature and verification time benchmark of pre and post-quantum signatures.

different instantiations of the hash functions on the total Merkle tree construction time as the number of leaves increases. The **build time per leaf** normalizes Merkle tree construction cost by the number of leaves to compare the per-element hashing overhead across equivalent pre- and post-quantum hash functions. The **generation time** measures the cost of generating Merkle authentication paths, highlighting the overhead of proof extraction from a precomputed tree under different hash configurations. Finally, **verification time** of the proof compares the computational cost of verifying Merkle proofs, isolating the impact of hash function choice on membership verification performance. Together, these metrics characterize the construction and verification costs of Merkle trees under equivalent classical and post-quantum hash instantiations. So, hash functions are compared using equivalent output lengths to ensure a fair evaluation of classical and post-quantum security levels, and the results are represented

with the corresponding plot.

The python tool[5] was developed to assess the binary Merkle tree and simultaneously generate the corresponding plots, as per the aforementioned metrics. Furthermore, this python script allows the user to implement several combinations of his own logic by passing the corresponding parameters. Such parameters can be the size of the tree, the number of repetitions, the names of the output files, etc. For this example, the script was run under the following parameters:

```
1     --max-power 18 --iters 20 --bench-proof --csv merkle-
         bench.csv --save-prefix result
```

Thus, the generated graphs concern a Merkle tree with $depth = 18$, which means 2^{18} number of leafs, that is, the blockchain transactions. The algorithms that were under test are the SHA-256/512 and SHAKE256 representing, respectively, the pre and post-quantum era. In this context, the results quantify the overhead introduced by increasing hash output sizes to maintain post-quantum security while preserving the same Merkle tree structure.

As a result, Fig 5.7 shows that transitioning to post-quantum secure hash parameters does not alter the asymptotic construction cost of Merkle trees, supporting their continued use in post-quantum systems. By focusing on per-leaf cost, the experiment highlights the direct performance impact of larger hash outputs required to counter Grover-style quantum attacks. The results in Fig. 5.8 indicate that post-quantum security for hash-based constructions can be achieved with a predictable and linear increase in per-leaf computation, rather than a structural redesign.

[5]https://github.com/konnnGit/merkle-tree-PQC-benchmark/blob/main/mekle-bench-equiv-plot.py

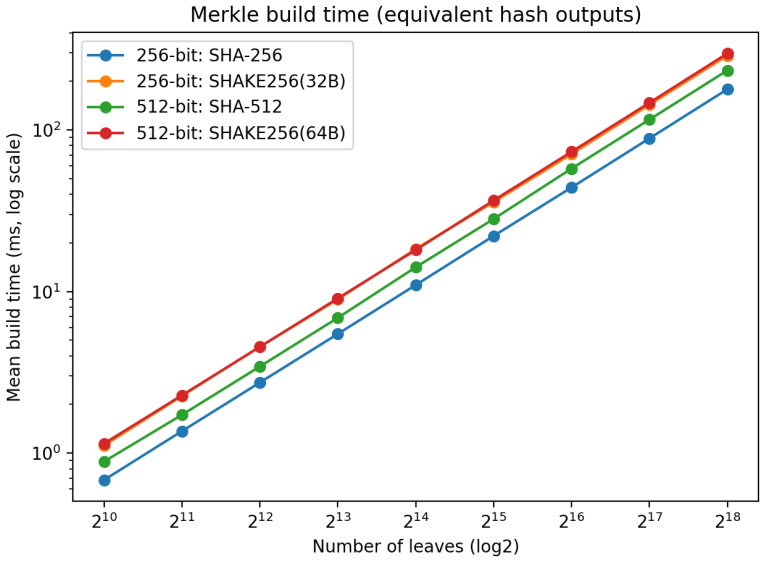

Figure 5.7: The impact of different instantiations of the hash functions on the total Merkle tree construction time.

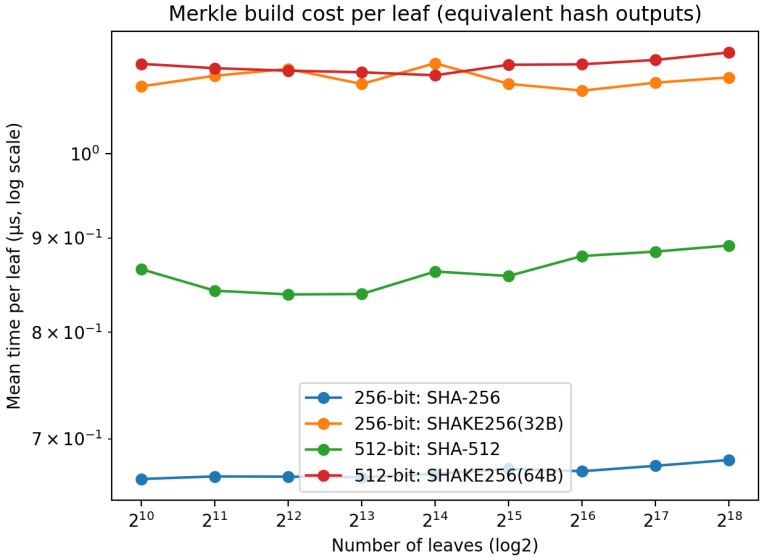

Figure 5.8: Post-quantum security appears with predicatively, small and steady overhead in per-leaf proof.

The following metrics concern the proof process, which is a fundamental function of the Merkle tree. Proof generation mainly involves memory access rather than cryptographic computation, and the results in Fig. 5.9 show minimal sensitivity to hash function choice. This confirms that post-quantum hash parameterization does not sig-

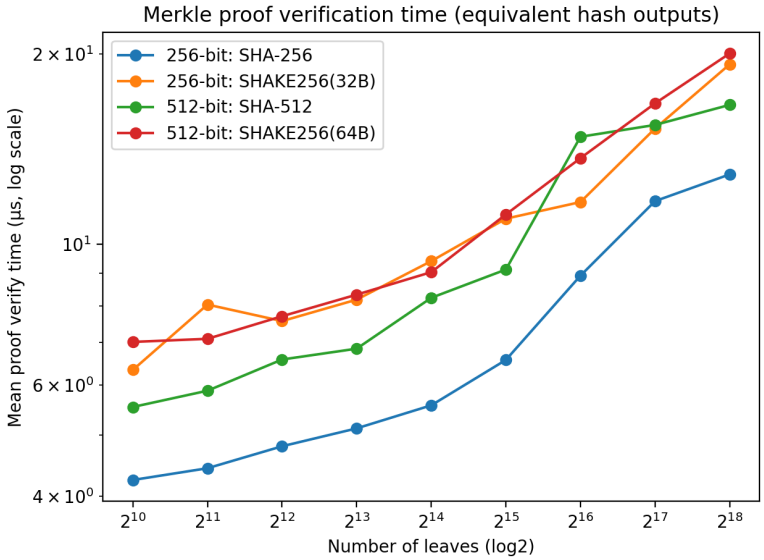

Figure 5.10: The impact of the hash function during of the verification procedure.

nificantly affect proof generation, which is critical for scalability in Merkle-based post-quantum protocols.

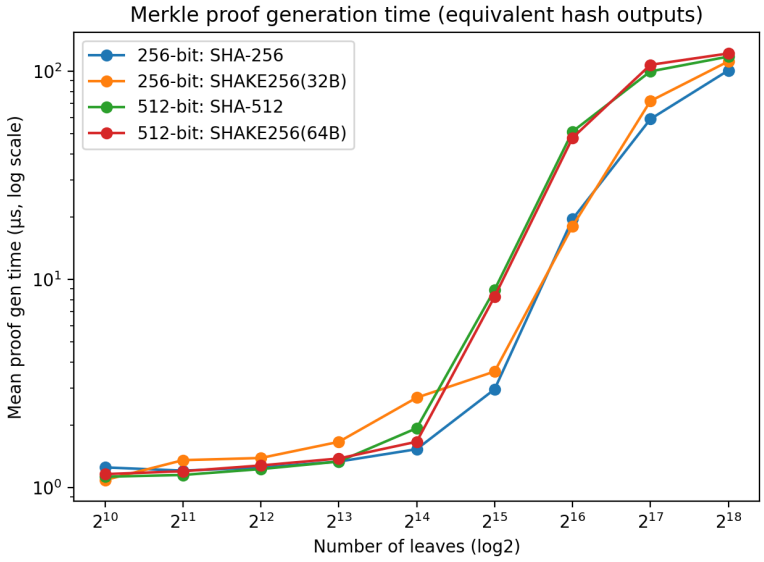

Figure 5.9: Low sensitivity of the proof generation per hash function.

Finally, the following results of Fig. 5.10 demonstrate that Merkle proof verification remains efficient under post-quantum hash settings, supporting the feasibility of Merkle-based authentication in quantum-resistant systems.

Overall, these baseline measurements show that Merkle tree operations retain their

efficiency and scalability under equivalent post-quantum hash instantiations, making them well-suited for integration into post-quantum cryptographic protocols.

5.2 Aviation

Aviation communications have evolved from simple analog voice transmissions to complex digital data links that ensure safety, efficiency, and security in modern air traffic management. Traditional systems such as the Aircraft Communications Addressing and Reporting System (ACARS) [78] provide global text-based messaging, while emerging solutions focus on higher bandwidth, stronger authentication, and resistance to cyber threats. Among these, the Aeronautical Mobile Airport Communications System (AeroMACS) [79] plays a crucial role on the airport surface by using the 5 GHz band allowing secure, high-speed, and locally confined communication between aircraft, ground vehicles, and control towers. At the same time, the L-band Digital Aeronautical Communications System (LDACS) [80] is being developed to extend secure broadband connectivity into en-route and terminal operations. Together, AeroMACS and LDACS illustrate the shift toward integrating secure aviation communication frameworks that support both local airport needs and long-range air traffic control.

In recent years, the aviation industry has witnessed a growing number of attacks targeting airport communication systems, ranging from signal jamming and spoofing of navigation aids to the interception of unprotected data links, such as legacy ACARS transmissions or communication frequencies. Researchers and security agencies have demonstrated that adversaries with relatively modest equipment can disrupt or manipulate communication channels, posing risks to operational safety and efficiency. These incidents highlight the urgency of strengthening authentication, encryption, and integrity mechanisms in both ground-based and air-to-ground networks. As airports and airlines continue to digitize their operations, the threat landscape expands, making cybersecurity a fundamental component of modern aviation communications. Effective classical cryptographic methods are expected to be vulnerable to quantum computing attacks that could compromise authentication, confidentiality, and integrity mechanisms. To secure future flight operations, air-to-ground data exchange, and airport surface communications, PQC integration is necessary. Quantum-resistant algorithms would ensure that sensitive information, such as flight plans, navigation updates, and control instructions remain secure even with the advent of large-scale quantum computers.

5.2.1 The L-Band Digital Aeronautical Communication System (LDACS)

The Aeronautical Telecommunication Network (ATN) [81] is a global digital communication infrastructure designed to interconnect aircraft with ground-based systems such as Air Traffic Control (ATC) and airline operations centers. Established under the guidance of the International Civil Aviation Organization (ICAO), the ATN integrates heterogeneous communication domains converting to a unified framework by

enabling Aviation Electronics (Avionics). Employing standardized protocols, primarily aligned with the Open Standard Intercommunication (OSI) reference model, the ATN ensures interoperability across different regions and service providers, creating a consistent worldwide network for aviation data exchange.

The Single European Sky Air Traffic Management Research (SESAR) [82] is actively developing next-generation aeronautical communications. One of its central objectives is the development of the LDACS as a modern communication standard to replace aging and inefficient aeronautical communication systems, such as VHF Data Link (VDL). Operating as a cellular-style digital communication system designed for Air-to-Air (A/A) and Air-to-Ground (A/G) exchanges, as well as flight guidance. It functions in the L-band frequency range between 960 and 1164 MHz, a spectrum specifically allocated for aeronautical services. It relies on Orthogonal Frequency Division Multiplexing (OFDM) to achieve efficient spectrum usage and resilience against interference, while also offering flexible channel bandwidths to adapt to different operational scenarios. LDACS aims to serve as a key enabler for future Air Traffic Management (ATM) modernization by using higher data rates, improved coverage, and enhanced reliability, compared to legacy systems.

The new standards of the ATN switch it from an insecure OSI to a secure TCP/IP-like communication architecture. Functions such as handshakes, mutual authentication, and symmetric key establishment will be implemented in this modern architecture. The ATN will include the Internet Protocol Suite (IPS) and thus the ATN/IPS would adopt IPv6, TCP/UDP, and other standards to align aviation with the rest of the internet world. Hence, moving aviation communications to IPS would inherit both the strengths and weaknesses of the internet world.

The Controller-Pilot Data Link Communications (CPDL), which is a communication system that allows pilots and ATC to exchange text-based messages instead of relying solely on voice radio, is a major candidate to employ LDACS and a key point reason where cryptography must secure the information in contrast to the traditional voice information exchanged. As LDACS moves toward standardization and deployment, its security requirements are becoming increasingly important, particularly in light of emerging quantum computing threats. Traditional cryptographic algorithms such as RSA and ECC, which have been proposed to secure aeronautical communications during the pre-quantum era, will be vulnerable to quantum attacks. This creates a pressing need to integrate PQC into the LDACS framework. Due to its flexible architecture and reliance on digital communication protocols, LDACS provides a suitable platform for experimenting with and embedding quantum-resistant algorithms such as lattice-based key encapsulation mechanisms or code-based cryptography. Incorporating PQC into LDACS can future-proof its authentication, confidentiality, and integrity services, ensuring that the next generation of aeronautical communications remains secure in both classical and post-quantum threat environments.

The specifications sheet of LDACS provision a TCP/IP-like architecture, then the traditional and well known architecture named Public Key Infrastructure (PKI) would

be the authentication and shared secret key exchanged mechanism. Besides that, some others introduce a more cutting-edge PKI but potentially more unstable, such as a Physical Unclonable Functions (PUF) based one. With respect to the mutual authentication point of view, the two options mentioned are briefly analyzed below.

5.2.2 PQC Mutual Authentication for LDACS

The **PUF-based** mutual authentication proposal [83] aims to be a lightweight set up system that would occupy a few bits in order for two parties to establish a session key. In particular, this promising approach aims to enhance the security of the LDACS integrating hardware characteristics. PUFs exploit inherent manufacturing variations in hardware to generate unique unpredictable responses to input challenges, making them highly resistant to cloning and manipulation. Within LDACS, a PUF-based authentication scheme can ensure that only legitimate aircraft and ground stations participate in communication, thereby reducing the risk of spoofing or unauthorized access. Unlike traditional key storage methods, which are vulnerable to extraction or compromise, PUFs derive cryptographic material directly from the physical properties of the hardware, eliminating the need to store long-term secrets. This theoretical lightweight and hardware-rooted security system could be well-suited for LDACS, where strong authentication is critical to safeguarding air-to-ground exchanges in both classical and post-quantum threat environments.

On the other hand, **PQC PKI-based** mutual authentication [84] focuses on mitigating vulnerabilities in contrast to a PUF-based, and can significantly strengthen its authentication and integrity services, particularly in anticipation of quantum-capable adversaries. In this case, as in the blockchain, a framework with post-quantum signature schemes such as CRYSTALS-Dilithium and Falcon can replace classical algorithms to ensure long-term cryptographic resilience. CRYSTALS-Dilithium provides a balance of strong security and efficient implementation, especially in a constraint environment. However, Falcon offers compact signatures suitable for bandwidth-constrained aeronautical channels. By embedding these schemes into a PKI for LDACS, aircraft and ground stations can authenticate digital certificates and verify message integrity without relying on vulnerable pre-quantum primitives.

Latency is the ultimate enemy in environments such as aviation, and concerning this, there are studies that propose other techniques, such as pre-handover [85], an excellent logic to allow the aircraft to be pre-authenticated with the next ground station before it starts communicating with it. To do so, this architecture uses the flight path, already registered during mission planning, and thus the way points are pre-selected and known. This kind of authentication procedure certainly reduces latency but may additionally reduce the flexibility to alter the way points, namely flight plan. Performing flight path deviation due to emergencies is a common tactic, and it is not crystal clear how the aforementioned pre-authentication architecture would handle such a case.

As a result and while the authentication procedure will highly resemble the well-established PKI mindset, this thesis aims to perform a thorough analysis of the authentication mechanism with respect to a post-quantum LDACS where aircraft and ground stations need to authenticate each other to establish a secure communication channel. In order to be more comprehensive, the authentication system described in the following lines maintains a more abstract architecture which focuses on studying the potential overhead that PQC signatures would add to the system. That is firmly acceptable, because the data that are transmitted are grouped to frames. So, the rationale of the thesis zooms in on a particular ingredient of the authentication mechanism, keeping the data exchanged in a minimal format, isolating them from other data described in the data sheet. Therefore, the architecture to be studied is the one depicted in Fig. 5.11 and proposed in [84]. As an alternative to the initial specs, which provision only NIST's Falcon, the following proposal would study NIST's standard CRYSTALS-Dilithium as the digital signature scheme between the aircraft and the certification authority (CA), while the Falcon standard would be used between the CA and the ground station. Each choice has its own pros and cons with respect to the size of the keys, or the computational cost for signing or verifying, and consequently, adding space or latency restrictions to the system.

So, based on the logic of the PQC PKI authentication procedures in Fig. 5.11, the messages that should be exchanged between the Aircraft Station (AS) and the Ground Station (GS), until they verify themselves, can be condensed in the following Alg. 5.2.2. The prerequisites to achieve mutual authentication are that both parties have stored their own valid certifications and have produced their equivalent key pair, as required according to the LDACS specification [86]. In addition, in the specifications datasheet, finalized in year 2023, provisions were made that peers would utilize the X.509 v3 certificate. So, as this environment operates in a restricted fashion, these peers will communicate in an intranet-like framework, which implies that during scheduled maintenance their certificates should be renewed as a standard practice. Any other illegitimate individual that attempts to exchange data would be rejected due to an invalid certificate that is not included in the peers databases.

In such a way, let us denote $Cert_{AS}$ and $Cert_{GS}$ the AS and GS X.509 certificates, respectively. Similarly, it is denoted (pk_{AS}, sk_{AS}) the public and secret key generated from the CRYSTALS-Dilitium algorithm, utilized by AS. Consequently, it is denoted (pk_{GS}, sk_{GS}) of the Falcon algorithm, utilized by the GS.

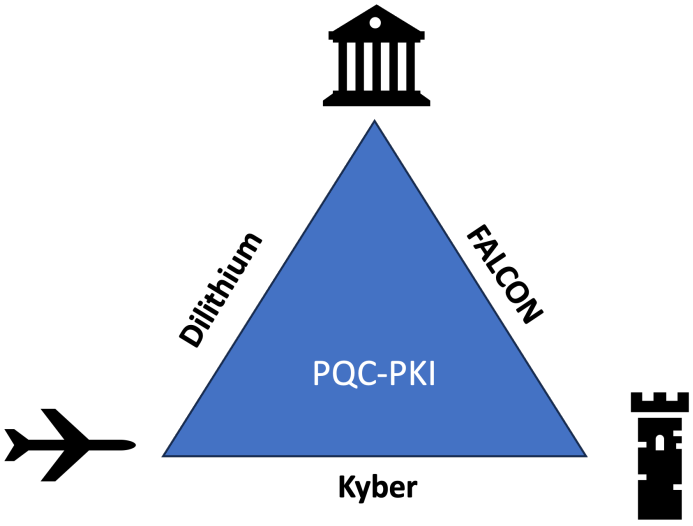

Figure 5.11: LDACS PQC-based PKI.

Algorithm 5.1 LDACS PQC PKI Mutual Authentication

Preconditions:
GS holds a valid X.509 $Cert_{AS}$
GS holds key pair $(sigpk_{GS}, sigsk_{GS})$ of the FALCON signature algorithm
AS holds a valid X.509 $Cert_{GS}$
AS holds key pair $(sigpk_{AS}, sigsk_{AS})$ of the CRYSTALS-Dilitium signature algorithm
$context \leftarrow$ "LDACS-Handshake"
Outcome:
If all certificate validations and signature checks pass, then mutual authentication holds.

(GS → AS): Hello
GS samples N_{GS}
$Data_{GS} = context||$"M1"$||ID_{GS}||ID_{AS}||N_{GS}$
$\sigma_{GS} \leftarrow Sign_{GS}(sigsk_{GS}, Data_{GS})$
Send $\boxed{M1 = Data_{GS}||\sigma_{GS}}$

AS processing of M1
Locate $Cert_{GS}$ by ID_{GS}
if $Verify_{GS}(sigpk_{GS}, Data_{GS}, \sigma_{GS})$ == false **then**
    abort
else
    (AS → GS): GS Authenticated
    AS samples N_{AS}
    $Data_{AS} = context||$"M2"$||ID_{AS}||ID_{GS}||N_{AS}||N_{GS}$
    $\sigma_{AS} \leftarrow Sign_{AS}(sigsk_{GS}, Data_{AS})$
    Send $\boxed{M2 = Data_{AS}||\sigma_{AS}}$
end

GS processing of M2
Locate $Cert_{AS}$ by ID_{AS}
if $Verify_{AS}(sigpk_{GS}, Data_{AS}, \sigma_{AS})$ == false **then**
    abort
else
    (GS → AS): Finish
    $h_{GS} = hash(M1||M2)$
    $\sigma_{GS} \leftarrow Sign_{AS}(sigsk_{AS}, h_{GS})$
    Send $\boxed{M3 = h_{GS}||\sigma_{GS}}$
end

5.2.3 LDACS Data Link (DL)

Before calculating the performance of the LDACS, is it essential to distinguish the internal building blocks related to the protocols and data frames. In a more abstract manner, within LDACS, the Service Network Protocol (SNP) acts as the link between the AS and the GS Data Link Layer (DLL), providing end-to-end user-plane communic-

ation between the two. The security architecture of LDACS is distributed across different layers, which is represented in Fig. 5.12, and includes the Logical Management Entity (LME) which handles *authentication* and key establishment, as it is the first logical component to initiate communication between the AS and GS. Security functions related to user data, such as encryption, integrity verification, and authenticity protection, are implemented within the SNP. In contrast, control data security mechanisms are implemented at the LME and Medium Access Control (MAC) layers.

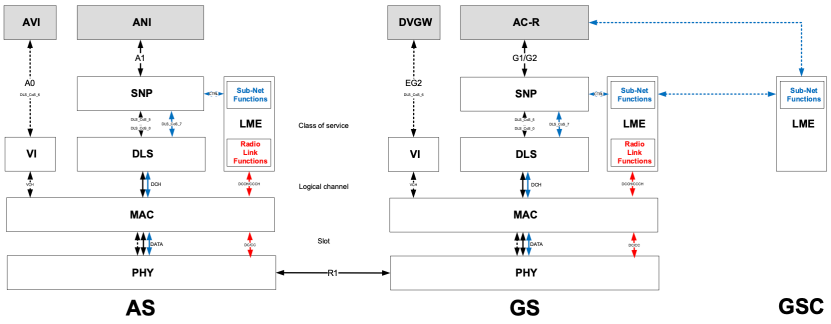

Figure 5.12: LDACS Sub-network protocol architecture.

In the LDACS specifications datasheet, the communication is organized into a Forward Link (FL) and a Reverse Link (RL). The FL refers to transmissions from the GS to the AS, typically used for air traffic control messages, broadcast information, and synchronization data. The RL is the channel from the aircraft back to the ground station, carrying pilot responses, aircraft status reports, requests, etc. Together, these links establish a bidirectional A/G communication path, with the FL often optimized for broad coverage and control, while the RL focuses on efficient reporting from multiple aircraft within the same cell. Although there is no other protocol above the LME, we can equal the output of the LME to the payload, during the Mutual Authentication Key Establishment (MAKE) protocol. In this case, and concerning the aforementioned mutual authentication procedures during the messages exchanged , as shown in Alg. 5.2.2, it is clear that the messages M1 and M3 are transmitted through the FL, whereas M2 is transmitted through the RL. Hence, the authentication payload can be expressed with the following equation:

$$Payload(Auth)_{bits} = Payload(M1) + Payload(M2) + Payload(M3) \tag{5.1}$$

$$= (|\sigma_{GS}| + |Data_{GS}|)_{FL} + (|\sigma_{AS}| + |Data_{AS}|)_{RL} + (|\sigma_{GS}| + |h_{GS}|)_{FL} \tag{5.2}$$

which is also equal to summing up the payload of the FL and RL that demand by the peers to establish the authentication between them.

Each of these links comprises several modules/blocks, most of them utilized to encode the initial message and randomize the codeword sent. This function is crucial in communications targeting to mitigate the probability of an information bit flipped due

to channel noise. The most important blocks according to [86] , [87] are represented in Fig. 5.13. Among all, those that alter the initial size of the information transmitted, and thus participate at the increase of the computational effort which is translated to bandwidth capacity, are the following:

Reed-Solomon (RS) codes are a class of error-correcting codes that operate over finite fields and are widely used in digital communication and data storage systems. Introduced by Irving Reed and Gustave Solomon in 1960, these codes can detect and correct multiple symbol errors within blocks of data. An (n_{RS}, k_{RS}) RS code encodes the message of m bits in blocks of a k_{RS} bytes each, also named symbols. So, for each block of the produced codeword, it contains n_{RS} byte symbols, incorporating $n_{RS} - k_{RS}$ redundant symbols that act as parity symbols. The main advantage of RS codes lies in their strong error-correction capability and their effectiveness in handling burst errors. LDACS can use several systematic RS codes, over a Galois Field (GF), where its general parameters can be denoted as $RS(n_{RS}, k_{RS})$.

Let us denote m, the message bits that must pass through RS. This binary information is converted by the RS and then the number of bits that exit the RS block, needed to protect the information from errors, is given by the following equation:

$$RS_{bits} = \frac{m \cdot (n_{RS} - k_{RS})}{k_{RS}} + m = m \cdot \frac{n_{RS}}{k_{RS}} \tag{5.3}$$

where $\frac{m \cdot (n_{RS} - k_{RS})}{k_{RS}}$ are the parity bits.

The **convolutional encoder** is a type of error-correcting code used in digital communication systems to improve the reliability of data transmission over noisy channels. Unlike block codes, which encode data in fixed-size blocks, convolutional encoders process input bits in a continuous stream, generating output bits that depend on both the current input and a number of previous inputs stored in the encoder's memory. The encoding process is typically represented using shift registers and modulo-2 adders, where the output bits are obtained by convolving the input sequence with a set of generator polynomials. The performance of the encoder is characterized by its rate R_c (the ratio of input bits to output bits) and the constraint length (the number of memory elements influencing each output). Convolutional codes are often decoded using algorithms such as the Viterbi or BCJR algorithms, which perform maximum likelihood decoding to correct errors introduced during transmission. Due to their robustness and relatively simple hardware implementation, convolutional encoders are widely used in applications such as satellite communications, mobile networks, and deep-space telemetry. According to Fig. 5.1, the input to the convolutional encoder is the output of the Reed-Solomon encoded word. Only these two blocks modify the size of the initial message in bits. The rest of the interleaving blocks just reshape the codeword. Usually, for every input bit this encoder doubles, or triples, the output putting additional bits. This can be mathematically expressed as $R_c = 1/2$ or $1/3$. In such a case, the total bit length of the codeword that outputs the convolution encoder is calculated by the following equation:

$$Convolution_{bits} = \frac{RS_{bits}}{R_c} \tag{5.4}$$

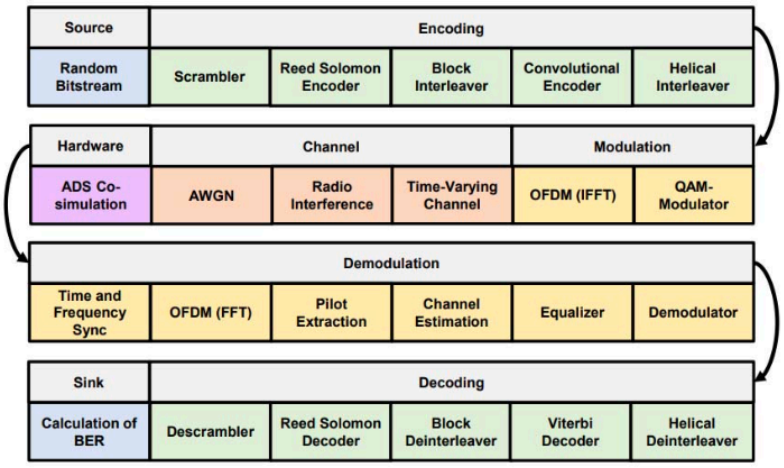

Figure 5.13: LDACS Transmission channel blocks, Tx & RX.

Digital modulators like Quadrature Phase Shift Keying (QPSK) or Quadrature Amplitude Modulation (QAM) are widely used in modern communication systems for efficient data transmission over bandwidth-limited channels. QPSK is a phase modulation scheme, where each symbol represents two bits of information by shifting the carrier signal among four distinct phases, typically separated by 90 degrees. This approach doubles the data rate compared to Binary Phase Shift Keying (BPSK) without increasing the bandwidth. QAM, on the other hand, combines both amplitude and phase modulation to represent multiple bits per symbol. For example, in 16QAM (or $2^4$4QAM), the carrier assumes 16 distinct states, each corresponding to a unique combination of amplitude and phase, thus encoding 4 bits per symbol. Although QAM offers higher spectral efficiency than QPSK, it is more susceptible to noise and nonlinear distortion due to its reliance on precise amplitude levels. Both modulation schemes strike a balance between data rate, power efficiency, and error performance and are extensively applied in systems such as Wi-Fi, LTE, and satellite communications. Apart from the QPSK that maps the input bits to two **symbols**, namely $QPSK_{symbols} = 2$, the other $XQAM$ modulator output is formulated as:

$$XQAM_{symbols} = \log_2(X) \tag{5.5}$$

For example, the 64QAM maps all the possible 5-bit length combinations to complex numbers as symbols on a cartesian plane represented in Fig. 5.14 . Then, the output of the modulator block is the payload that will later be fragmented into OFDM symbols (again complex numbers) and transmitted through the channel. The modulator output can be formulated as follows:

$$Modulator(XQAM)_{symbols} = \frac{Convolution_{bits}}{XQAM_{symbols}} \tag{5.6}$$

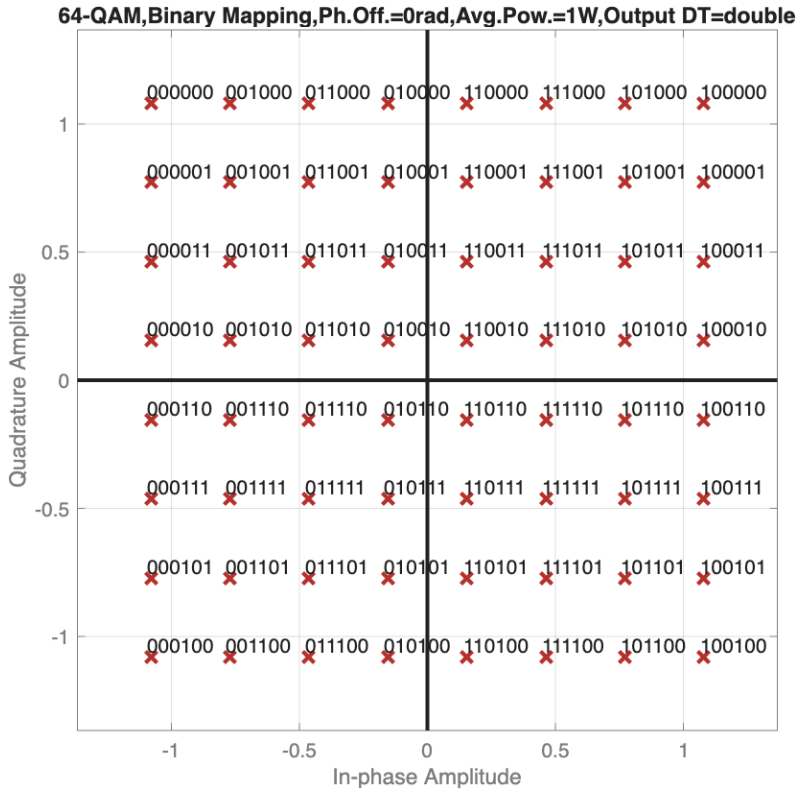

Figure 5.14: Constellation of 64QAM symbols (complex numbers).

An **Orthogonal Frequency Division Multiplexing** (OFDM) symbol represents a block of data transmitted across multiple orthogonal subcarriers within a specific time interval. Each subcarrier in the OFDM system is independently modulated, typically using QPSK or QAM, producing a complex symbol that conveys information in both amplitude and phase. These modulated subcarriers are then combined using the Inverse Fast Fourier Transform (IFFT) to form a composite time-domain signal representing the OFDM symbol. To prevent inter-symbol interference, a copy of the end portion of the OFDM symbol is appended to its beginning before transmission. The cyclic prefix converts the linear convolution of the channel into a circular one, preserving subcarrier orthogonality and enabling simple frequency-domain equalization at the receiver. So,each OFDM symbol consists of two main parts: the useful data and the cyclic prefix. The structure of OFDM symbols allows for parallel data transmission, high spectral efficiency, and resilience against channel distortion, making OFDM the preferred modulation technique in contemporary broadband systems such as LTE, 5G, and Wi-Fi.

In practical OFDM-based systems, **frames** serve as the higher-level structure that organizes multiple OFDM symbols for transmission. A frame typically consists of sev-

eral OFDM symbols transmitted sequentially in time, along with additional components such as preambles, pilot symbols, and control fields. The preamble is used for synchronization and channel estimation, allowing the receiver to correctly detect the start of the frame and estimate the channel characteristics. Pilot symbols are inserted into predefined subcarriers within some OFDM symbols to enable continuous channel tracking and equalization. The remaining symbols within the frame carry user data or payload information. Fig. 5.15 breaks down the several symbols in an OFDM LDACS frame where each of the sub-squares corresponds to the subcarriers used to transfer binary data. According to LDACS specification, apart from the pilot and sync symbols , an OFDM frame can carry up to 2442 data symbols. Finally, no other block processes the amount of the information message towards the communication channel, and thus, the number of frames needed to carry information would be calculated as follows:

$$frames = \frac{Modulator()_{symbols}}{2442} \tag{5.7}$$

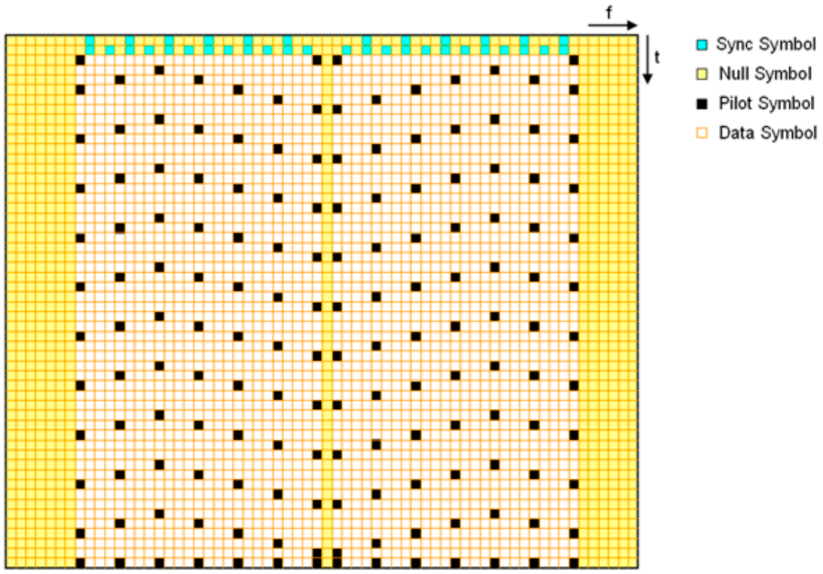

Figure 5.15: LDACS OFDM Frame.

Framing plays a critical role in both wired and wireless communication systems by defining clear boundaries between successive data units, preventing data overlap and loss. Frames also enable retransmission mechanisms in the event of errors, which supports reliable communication. In Fig. 5.16 LDACS physical layer frames are depicted, while in Fig. 5.17 it is organized in batches, followed by their corresponding execution time. For **FL**, which means that the GS transmits data, each frame takes time $t_{frame} = 6.48ms$ plus a broadcast frame (BC) of duration $t_{BC} = 6.72ms$. Therefore, the total time needed for transmission is:

$$t_{FL} = frames \cdot t_{frame} + t_{BC} \tag{5.8}$$

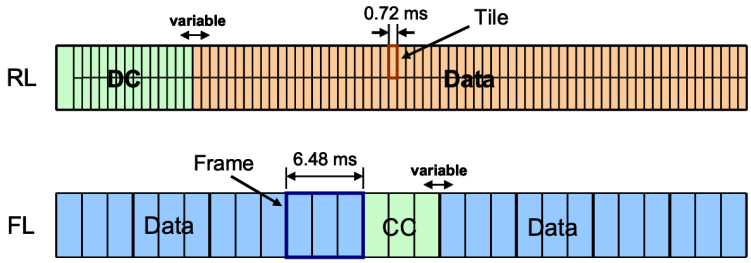

Figure 5.16: LDACS FL and RL single frame timings.

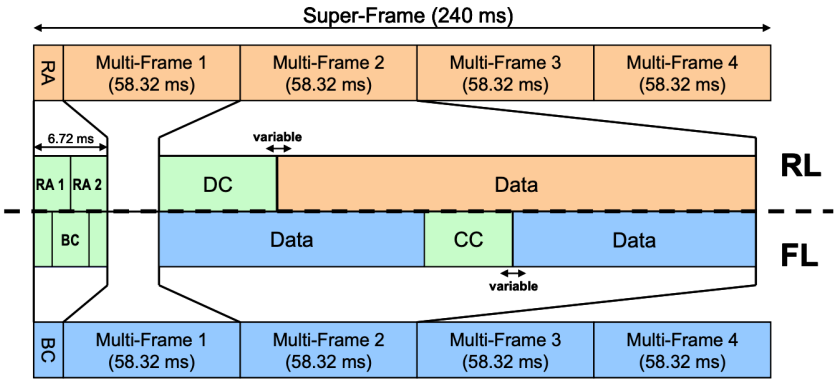

Figure 5.17: LDACS FL and RL super frame timing.

where for the RL is just:

$$t_{RL} = frames \cdot t_{frame} \tag{5.9}$$

5.2.4 LDACS PQC Performance

To calculate the latency of the system, the corresponding payload of messages M1, M2, and M3 must be calculated. The logic of this effort is to count how many OFDM frames are needed to host these messages, corresponding to the authentication procedures. The Tab. 5.2 stores the payload of the two PQC signature algorithms proposed for the LDACS PKI, but also some pre-quantum algorithms, for comparison. In addition, to calculate the total payload for M1, the size of the $Data_{GS}$, which equals to the $|h_{AS}| = 256\ bits$, the static binary string $|context| = 80\ bits$ and the $|M2| = 16\ bits$ must be taken into account. Indicatively, Tab. 5.3 lists these sizes.

Algorithm	Signature (B)
Dilithium-3/5	3293/4595
Falcon-512/1024	666/1280
RSA-4096	512
ECDSA-P521	132

Table 5.2: Signature sizes (in bytes).

Field	Size (bits)	Example contents
GS ID	24	`0xA13F22`
AS ID (Aircraft ICAO address)	24	`0x48A2B3`
Nonce AS	128	Random (challenge)
Nonce GS	128	Random (response)

Table 5.3: Additional data size (in bits).

Before calculating the latency of the PKI, it should be determined the LDACS transmission characteristics, such as the RS, convolution encoder, and modulator, because they affect the size of the data ending up in the OFDM. Recall that FL (M1 or M3) uses the Falcon PQC scheme. Consequently, a representative paradigm of communication a channel characteristics would have an *RS(255,239)* encoder, a $R_c = 1/2$ convolution encoder, and finally a *64QAM* modulator. The next step is to calculate the frames needed to host the 64QAM output. Using Eq. 5.1, we can see that the payload of the message M1 is equal to:

$$\begin{aligned} Payload(M1) &= |\sigma_{GS}| + |Data_{GS}| \\ &= |\sigma_{GS}| + |context||\texttt{"M1"}||ID_{GS}||ID_{AS}||N_{AS}| \\ &\simeq 666 \cdot 8 + 400 \\ &= 5728\ bits \end{aligned}$$

padding one byte in order to fit the RS(255,239), because $239 \cdot 8 = 717$, and so $Payload(M1) = 717 = 5736$ bits. Then, combining Eq. 5.7, 5.3, 5.5, 5.4 , the number of frames needed to transmit the message M1, via the OFDM, are:

$$\begin{aligned} frames &= \frac{Modulator(64QAM)_{symbols}}{2442} \\ &= \frac{Convolution_{bits}}{2442 \cdot XQAM_{symbols}} \\ &= \frac{RS_{bits}}{2442 \cdot XQAM_{symbols} \cdot R_c} \\ &= \frac{|M1| \cdot n_{RS}}{2442 \cdot XQAM_{symbols} \cdot R_c \cdot k_{RS}} \end{aligned}$$

$$= \frac{2 \cdot 5736 \cdot 255}{2442 \cdot 6 \cdot 239}$$
$$\simeq 0.83$$

This result can be verified using Matlab Simulink with the configuration depicted in Fig. 5.18. In particular, applying a random 5736-bit input sequence corresponding to the message M1, we can see that the output of the 64QAM modulator is a row vector consisting of 2040 symbols. These symbols can be easily placed in the data placeholder subcarriers of the OFDM frame, which has the size of 2442 data symbols. In the same way, interpreting the result of the above equation, the bit-stream of the message M1, sent from GS to AS can be transmitted within one FL frame and can easily host the post-quantum Falcon signature without adding any overhead with respect to a pre-quantum algorithm. In the time domain, this frame spends a total transmission time of up to $t_{M1} = t_{M3} = t_{FL} = t_f + t_{BC} = 6.48ms + 6.72ms = 13.2ms$, according to Eq. 5.9. In this context, Tab. 5.4 and Fig. 5.19 represent the same combinations of transmission blocks and the respective demand for OFDM frames. Taking into account the size of the information and the time required for the frames to be transmitted, the corresponding latency is calculated and depicted in Fig. 5.20.

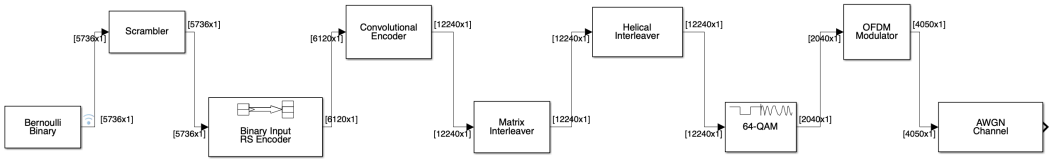

Figure 5.18: Matlab Simulink simulate the FL transmission including both Falcon signature and the additional data (in bits).

Alg.	Payload	RS	Modulation	Frames
Ed448	114	(255,239)/(202,182)	QPSK / 16-QAM / 64-QAM	1 / 1 / 1
ECDSA-P521	132	(255,239)/(202,182)	QPSK / 16-QAM / 64-QAM	1 / 1 / 1
RSA-4096	512	(255,239)/(202,182)	QPSK / 16-QAM / 64-QAM	2 / 2 / 1
Falcon-512	666	(255,239)/(202,182)	QPSK / 16-QAM / 64-QAM	3 / 2 / 1
Falcon-1024	1280	(255,239)/(202,182)	QPSK / 16-QAM / 64-QAM	5 / 3 / 2
Dilithium-3	3293	(255,239)/(202,182)	QPSK / 16-QAM / 64-QAM	11 / 6 / 4
Dilithium-5	4595	(255,239)/(202,182)	QPSK / 16-QAM / 64-QAM	16 / 8 / 6

Table 5.4: Number of OFDM frame needed to transmit the message, based on several channel configurations.

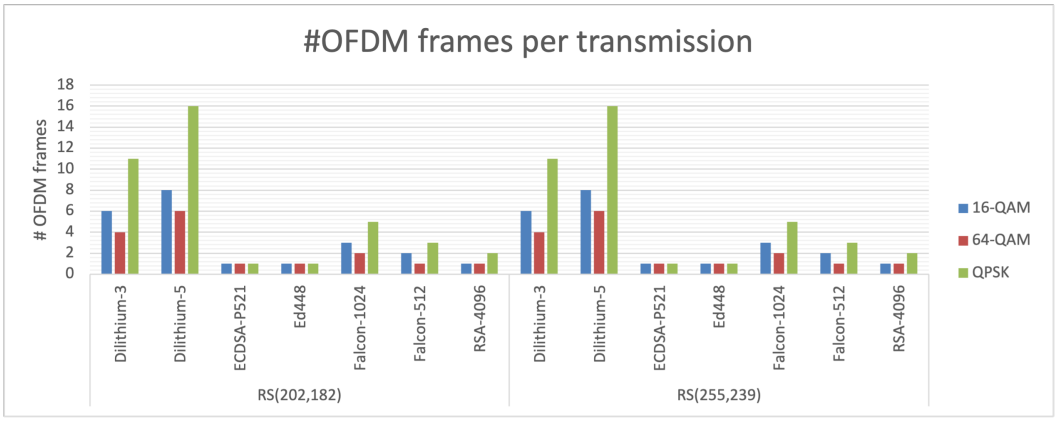

Figure 5.19: Number of OFDM frame needed to transmit the message, based on several channel configurations.

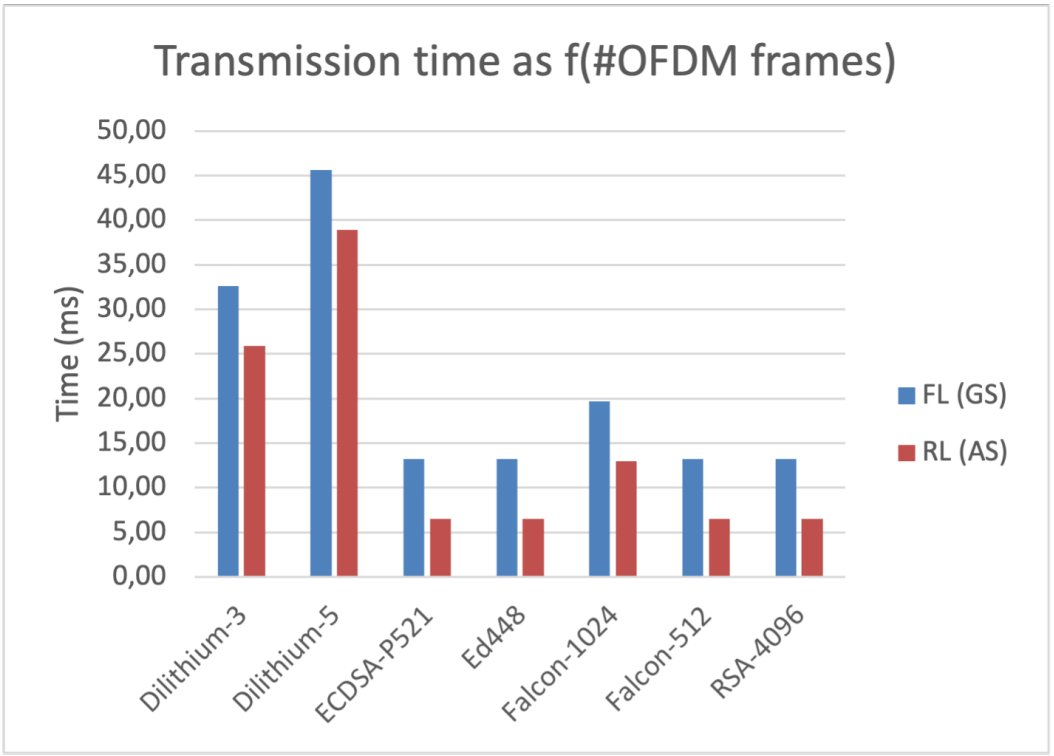

Figure 5.20: Transmission latency with respect to signature algorithm.

Finally, to calculate the total latency of the authentication part of the PQC PKI set up we must perform one additional calculation for the RL, which concerns the transmissions from the AS, as the above algorithm dictates. In addition, the overall system latency includes the execution time of the algorithms when they *sign* and *verify*. Applying the same Eq. 5.7, for all the messages M1, M2 and M3, we are able to calculate the number of frames each of these messages needs to be transmitted. Overall, the total

latency, including *sign* and *verification*, is 28.21ms and 41.37ms for NIST security levels 3 and 5, respectively.

The most impressive observation is that a combination of a PKI based on pre-quantum algorithms spends more time in total to perform mutual authentication. This happens because algorithms such as RSA and ECDSA require more time to *sign* and *verify*, compared to those based on lattices such as Falcon. These observations are depicted in Tab. 5.5 and Fig. 5.21 with respect to the computational capabilities of AS and GS . Then, considering the above, the mega question that arises is whether a PQC PKI is more appropriate to serve LDACS even from the pre-quantum era. This kind of logic may offer convenience with respect to engineering and maintenance. This logic saves man-hours and resources, resulting in less cost, complexity, and incompatibilities. Above all, NIST already consults[6] companies and organizations that should begin migrating their systems to quantum-resistant cryptography.

Algorithm	**Hardware cycles**		**@ 3 GHz(ms)**		**@ 500 MHz (ms)**	
	Sign	**Verify**	**Sign**	**Verify**	**Sign**	**Verify**
Ed25519	113 917	373 258	0.04	0.12	0.23	0.75
Ed448	420 827	1 193 634	0.14	0.40	0.84	2.39
Falcon-512	1 090 897	160 432	0.36	0.05	2.18	0.32
Falcon-1024	2 166 326	313 111	0.72	0.10	4.33	0.63
ECDSA-P521	2 289 063	4 170 636	0.76	1.39	4.58	8.34
Dilithium-3	2 624 981	770 641	0.87	0.26	5.25	1.54
Dilithium-5	3 213 626	1 322 178	1.07	0.44	6.43	2.64
RSA-4096	62 322 370	467 389	20.77	0.16	124.64	0.93

Table 5.5: Signature and verification performance on different hardware platforms. @3Ghz for GS computational capabilities and @500MHz for AS.

[6]https://csrc.nist.gov/projects/post-quantum-cryptography

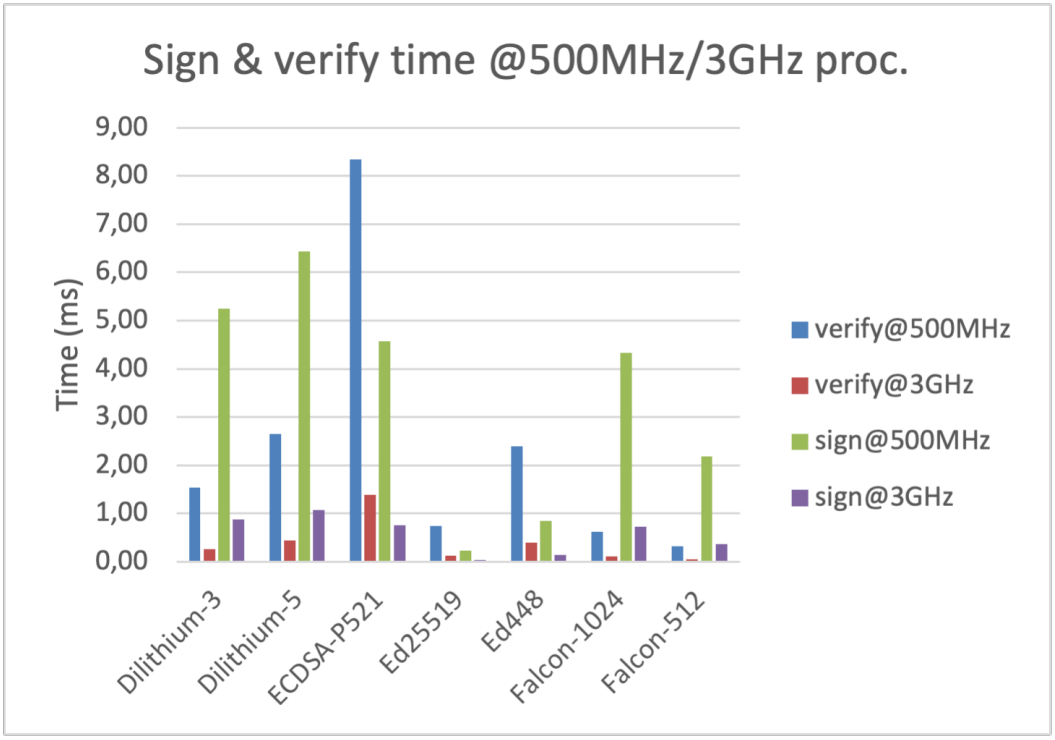

Figure 5.21: Signature and verification performance on different hardware platforms. @3Ghz corresponds to computational capabilities of GS and @500MHz of AS.

Error Rate

With respect to *error correction*, the specification datasheet provides that the Bit Error Rate (BER) can reach the value of 10^{-6}, which is the same as calculated by the authors in [87]. They calculate the BER, the Signal to Noise Ratio (SNR) and other metrics of the LDACS PHYsical layer (PHY) utilizing Matlab to simulate the communication channel and calculated metrics, incorporating all the building blocks. In that study, the authors did not include any interference that Distance Measurement Equipment (DME) may add to LDACS transmissions because these systems occupy almost the same frequency spectrum.

Consequently, while the estimated BER is 10^{-6}, we can utilize this to calculate the overall probability of success in trying to transmit data from the FL and the RL. In the previous subsections, the actual size of the PKI messages exchanged in order to establish authentication between AS and GS, has been calculated. Thus, we denote the probability P of success of the transmitted sequence of bit length L, as :

$$P_{success} = (1 - BER)^L \tag{5.10}$$

Hence, using the size of the M1 message that was calculated in the previous latency example, we can assign it as $L = |M1| = 5736\ bits$. In this case, the actual probability is calculated as follows:

$$P_{success} = (1 - 10^{-6})^{5736} = 0.9943$$

As a result, transmitting the M1 via the FL, incorporating the provisioned Falcon PQC signature, will reach the AS's cryptographic processor with probability equal to 99.45 %. In addition, Fig. 5.22 represents the probability trends regarding the PQC signatures set up based on NIST level 3 and 5, but also a set up that comprises pre-quantum algorithms. In that, we are able to notice that pre-quantum algorithms perform better regarding error rate because their codeword length is smaller than post-quantum signatures. Nevertheless, except for Dillithium-3/5 the other PQC signatures act significantly well with respect to the success of delivery non-erroneous data. In conjunction with the aforementioned latency outputs, it is crystal clear that a PKI based on NIST level 3 is a well trade-off choice for LDACS. Such a choice would be perfectly aligned with NIST's which urge the stakeholders to start migrating to the PQC.

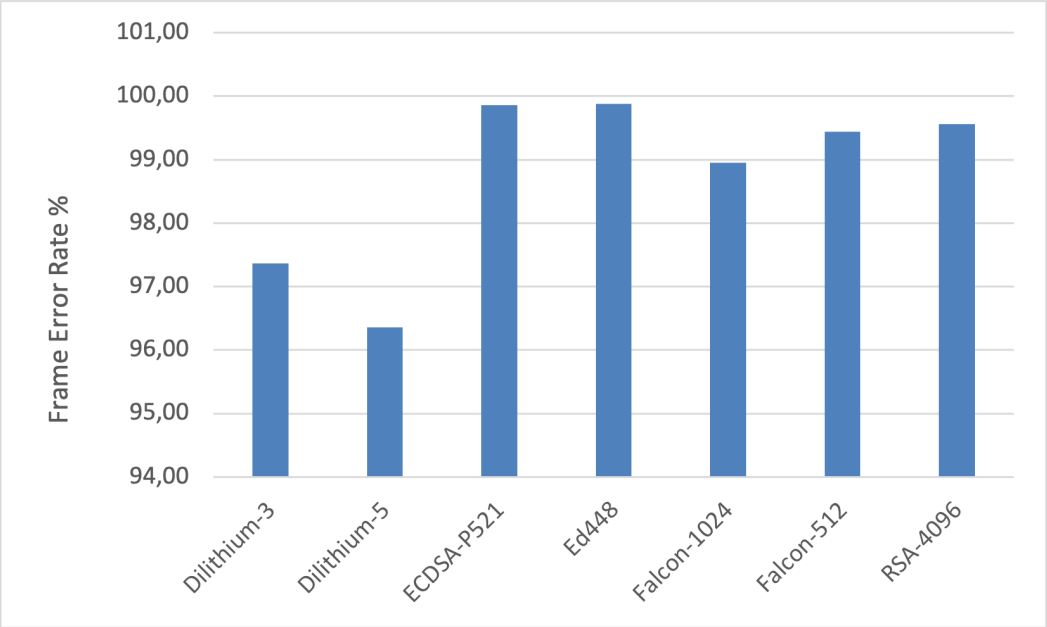

Figure 5.22: Error frame comparison for pre and post-quantum signatures.

5.2.5 A Blockchain Augmented Mutual Authentication for LDACS

In relation to the pre-handover rationale, but aligned to [84], for a PQC PKI LDACS, this study focuses on the ATN/IPS specifications. Although PQC is widely supposed to add more latency, this thesis indicates that a careful setup does not add additional latency. In contrast, pre-quantum algorithms would have worse impact due to computations complexity. Consequently, an alternative solution could impose the authentication to be established during preflight checks only. The question that arises next is how the handovers would take place without re-authentication procedures. To enable secure handovers without repeated authentication, a post-quantum ready blockchain

would be a solid alternative, as the authors analogously propose in [88]. In this case, the LDACS security would state across administrative domains, which is abstractly illustrated in Fig. 5.23.

More specifically, the ledger records the status of the certificate which is stakeholders' responsibility to provide. So, each peer tries to validate the X.509 certificate of the party that wants to establish communication, and if it is valid, it can proceed to a shared secret key exchange. Consequently, all payload and latency remain on ground and after the initial authentication establishes, each handover is authorized by presenting a ledger-backed ciphered ticket that the target ground station verifies against its locally synchronized view. Session keys are then derived via a Key Derivation Function (KDF), avoiding a full re-authentication exchange and consequently the delay is minimized.

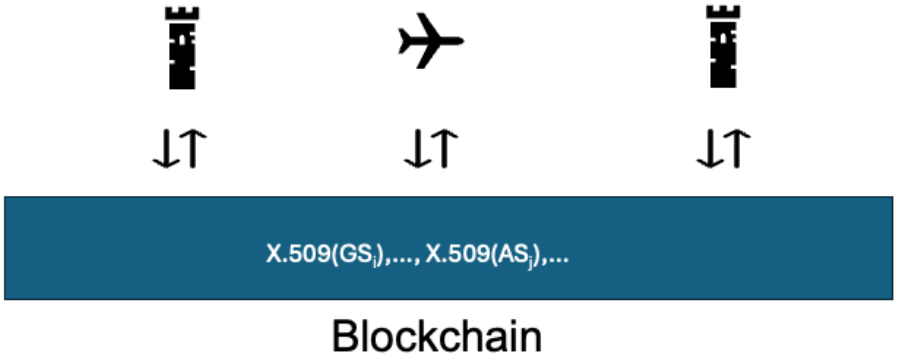

Figure 5.23: A Blockchain based LDACS PKI.

5.3 Smart Cities

A smart city is an urban environment that leverages advanced technologies such as the IoT, Artificial Intelligence (AI), and data analytics to improve the quality of life of its citizens, the efficiency of urban services, and promote a sustainable environment. In a smart city ecosystem, interconnected sensors and devices continuously collect and exchange data related to infrastructure, transportation, energy consumption, environmental conditions, and public safety. This real-time information enables city administrators to make data-driven decisions, optimize resource allocation, and respond proactively to emerging challenges. Key applications include intelligent traffic management, smart grids, waste monitoring, digital healthcare, and automated public services. Integration of communication networks, cloud computing, and edge processing

ensures that data is processed efficiently and securely. By combining technology with urban planning, smart cities aim to create more resilient and environmentally sustainable communities, aligning with the global vision of sustainable and human-centered urbanization.

In particular, in modern smart cities, the IoT serves as the foundational technology that allows the monitoring, automation, and optimization of urban systems in real-time. One of the most prominent applications is **smart transportation**, where IoT sensors embedded in vehicles, traffic lights, and road infrastructure provide dynamic traffic management, adaptive signal control, and congestion prediction to reduce delays and emissions. **Smart energy systems** also use IoT in smart meters and distributed sensors to optimize electricity distribution, integrate renewable sources, and monitor energy consumption patterns throughout the grid. **Environmental monitoring** is another key function, employing air quality sensors, weather stations, and water-quality probes to track pollution levels and support data-driven environmental policies. In **public safety**, IoT-based surveillance cameras, connected emergency systems, and predictive analytics improve situational awareness and rapid response to incidents. In addition, **smart waste management** systems use sensor-equipped bins that report fill levels, allowing efficient route planning for collection vehicles and reducing operational costs. These applications demonstrate how IoT connectivity transforms traditional urban infrastructure into intelligent and responsive networks that improve sustainability, safety, and overall urban living.

Consequently, the IoT plays a key role in the fundamental functions of smart cities. Thus, shielding them against cyber-attacks is crucial. Most IoT devices are used to operate in conjunction with other related devices, constantly exchanging information. Hence, the network must guaranty the security and availability because a cyber-attack could cost human life. For example, *traffic lights* and *autonomous vehicles* are a representative aspect of daily operations where any malicious action would have a significant impact. As a result, the integration of PQC is becoming increasingly important, as quantum computing poses a threat to classical cryptographic primitives, which are widely deployed in current IoT devices. However, implementing PQC in IoT environments presents unique challenges due to the constrained nature of many devices in terms of processing power, memory, and energy consumption. To address these limitations, lightweight PQC schemes, particularly lattice-based and hash-based constructions, are being actively studied for embedded applications. Ongoing research focuses on optimizing key sizes, reducing computational overhead, and developing hybrid security frameworks that combine classical and post-quantum algorithms.

5.3.1 Intelligent Transportation Systems (ITS)

An ITS is a significant cell of a smart city, and this thesis analyzes this aspect as it already plays a vital role in modern cities. In relation, the authors in [89] represent the integration of advanced technologies and how to improve the safety, efficiency, and sustainability of transportation networks. In an ITS environment, sensors, connec-

ted vehicles, traffic lights, and road infrastructure continuously collect and exchange real-time data. This information is analyzed to optimize traffic flow, reduce congestion, and minimize travel time. Applications include adaptive traffic signal control, Vehicle-to-Everything (V2X) communication, smart parking systems, and automated incident detection. Machine learning algorithms further enhance ITS performance by enabling predictive traffic management, such as forecasting congestion patterns and optimizing public transportation schedules. Furthermore, an ITS contributes to reducing fuel consumption and emissions by promoting smoother traffic movement and intelligent routing. Through the integration of cloud and edge computing, ITS ensures rapid data processing and reliable communication between vehicles and infrastructure. In general, intelligent transportation systems play a pivotal role in shaping the future of smart mobility, enabling safer, greener, and more connected urban transport environments, and consequently assembling a smart city.

5.3.2 Security Threats of an ITS

The authors in [90] thoroughly analyze the categories of attacks that an ITS can face. In addition, they propose the conjunction of artificial intelligence and blockchain in order to handle related attacks. The majority of attacks are based on manipulating a network, which is classified into already known categories.

More specifically, in a *Man-in-the-Middle* attack scenario, an adversary positions himself between Autonomous Vehicles (AVs) and the devices or infrastructure with which they communicate. These links, commonly based on Wi-Fi or cellular technologies, can be intercepted or altered by an attacker. After accessing the AV through its On-Board Diagnostics (OBD) port, it is possible to intercept communications with the cloud platform, extracting information about the vehicle, and even influence critical components, including the airbag control system.

In an *Impersonation Attack* , the attacker attempts to present himself as a legitimate participant by stealing the identity of a real AV or fabricating a new one. This may include distributing false information, suppressing important messages, or introducing malware into the system. The risk is particularly significant in Vehicle-to-Vehicle (V2V) environments, where nearby AVs routinely exchange safety messages. A malicious vehicle using a spoofed identity can integrate into the network, transmit fabricated data, and obtain sensitive information from surrounding AVs. For example, an attacker could falsely report multiple simultaneous positions, potentially causing artificial traffic congestion. In such attacks, the adversary typically forges or mimics only one identity at a time.

Passcodes and cryptographic keys serve as important security mechanisms in connected AVs. However, numerous studies have shown that these secrets can often be extracted from an AV OBD interface, using methods such as brute force, dictionary attacks, rainbow tables, password theft, or social engineering. For example, the Bluetooth interface of a connected AV can be compromised within seconds by brute force attack, as it relies on a simple four-digit PIN. Another commonly observed attack, applicable

to virtually any vehicle equipped with remote central locking, involves replaying captured key signals. By retransmitting a previously intercepted key signal, an attacker can unlock the vehicle without physically required access to the original key fob.

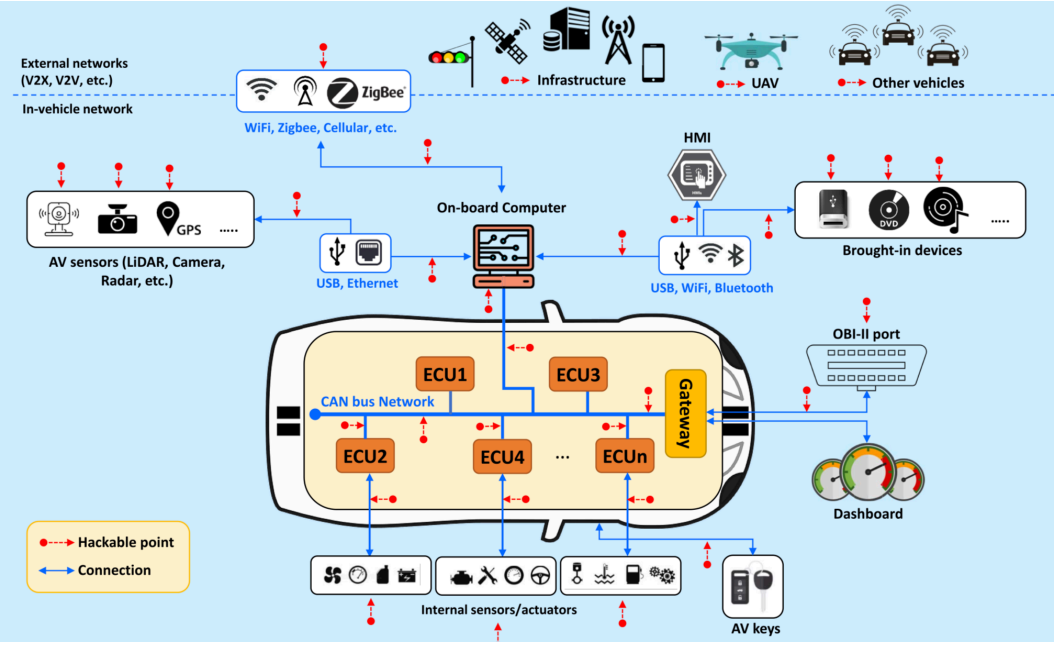

Figure 5.24: An overview of the multiple vulnerability points of an ITS.

5.3.3 PQC in ITS

Concerning the aforementioned, we can see that not only the AV per se, but also the network that relies on to exchange data, is vulnerable. In Fig. 5.24 it is crystal clear that there are several critical points where a malicious action could take place. Non-cryptographic communications should be substituted, and to be proactive looking towards the future, should also be quantum-resistant. Consequently, companies should try to experiment with the appropriate PQC schemes for this kind of IoT in order to achieve a quantum-safe ITS environment.

In particular, [91] surveys various PQC candidates, assessing their practical applicability within ITS contexts, especially in scenarios such as V2V, Vehicle-to-Infrastructure (V2I) and generally to V2X communications. Key considerations include computational load, signature or key sizes, latency, bandwidth overhead, and compatibility with existing communication standards. Through this detailed evaluation, the authors show that while PQC introduces overhead compared to classical schemes, certain PQC algorithms can be feasibly integrated into ITS with careful adaptation, making quantum-resilient security a realistic goal rather than a purely theoretical possibility. Therefore, the paper proposes that securing ITS against quantum threats requires a shift from classical cryptographic primitives to quantum-resistant alternatives, but such a shift must be guided by practical deployment constraints. To this end, the authors systematically

evaluated a range of candidate PQC schemes, including lattice-based, code-based, and hash-based algorithms, with a focus on their suitability for ITS environments. Their analysis carefully considers factors like computational overhead, key size, signature length, bandwidth and latency constraints, and compatibility with existing vehicular communication standards (such as V2X protocols). They argue that although PQC inevitably imposes greater resource demands compared to classical cryptography, certain algorithms and configurations strike a workable balance, making PQC adoption in ITS feasible without excessive performance degradation. The authors recommend that ITS designers integrate PQC through a phased migration rather than an abrupt wholesale replacement, allowing for backward compatibility and incremental deployment. The key components of this strategy include redesigning certificate and key-management processes to handle larger keys and signatures, optimizing protocols for real-time constraints

5.3.3.1 PQC in AV Authentication

Initial Access Authentication (IAA) is the first security step performed when an AV attempts to join a V2X communication network. During this phase, the vehicle must be authenticated before it can exchange safety-critical data with neighboring AVs or the related infrastructure. Cryptographic mechanisms are used to validate the vehicle's identity, establish trust relationships, and distribute temporary verification keys needed for secure communication. By ensuring that only authorized vehicles are admitted and that communication channels are protected from impersonation or message tampering, initial access authentication prevents malicious entities from infiltrating the platoon, disrupting cooperative driving maneuvers or manipulating vehicular behavior. Therefore, it forms the foundations of a secure and reliable decision-making system that coordinates connected AVs. The authors in [92] propose a three-component defense model that protects the IAA phase of autonomous vehicle platoons against quantum-enabled eavesdropping and active cyber-physical attacks. According to this, the defense framework of the IAA combines the following.

1. Vehicle Signal Separation and Identification (VSSI) to extract legitimate signals from mixed wireless transmissions and suppress high-power jamming or modification attacks.

2. Post-quantum encryption using QC-MDPC codes, where both preamble sequences and data bits used during IAA are jointly encoded to prevent interception and fabrication by a quantum-capable attacker.

3. Dynamic key generation using quantum-walk-based quantum encryption, which enables the legitimate receiver and vehicles to share ephemeral secret keys for QC-MDPC encryption with built-in randomness and resistance to quantum attacks.

Together, these components create a security model that filters malicious radio signals, ensures confidentiality of transmitted identifiers, and prevents attackers from forging preambles or data during platoon formation. This design explicitly addresses fabrication, interception, interruption, and modification attacks described in the threat model, and provides a quantum-secure mechanism for reliable and low-latency initial access in V2X-based autonomous vehicle platoons. Simulation results indicate that the designed approach successfully mitigates malicious disruptions and lowers the likelihood of system malfunctions triggered by both deliberate and accidental cyber-physical disturbances. By protecting IAA without compromising the stringent low-latency and high-reliability requirements of communication between autonomous vehicle platoons and infrastructure, the system presents a promising solution to secure access in future connected transportation environments.

5.3.3.2 AI-assisted PQC

Moreover, and in a more abstract manner, Transportation Cyber-Physical Systems (TCPS) are integrated systems that combine physical transportation components such as vehicles, sensors, and infrastructure with cyber elements such as communication networks, computing platforms, and control algorithms. By enabling real-time interaction between the physical and digital domains, TCPS supports intelligent transportation functions, including autonomous driving, traffic management, and safety monitoring. This tight integration improves the overall efficiency, safety, and reliability of modern transportation systems. Recall that sensitive data exchanged between vehicles, infrastructure, and external systems should depend on strong cryptographic mechanisms that ensure confidentiality, integrity, and authenticity. As a result, incorporating PQC into TCPS is becoming critical to maintain long-term security [93]. In addition, the authors argue that artificial intelligence can further enhance security by optimizing the selection of the PQC algorithm, managing computational resources, and dynamically adapting to new attack patterns. AI-assisted PQC solutions offer the potential to improve the resilience and performance of TCPS without imposing excessive overhead. This paper provides an overview of TCPS communication protocols, highlights their exposure to cyber-threats, examines the shortcomings of current cryptographic approaches in the quantum era, and discusses how AI can reinforce PQC-based protections to achieve cyber-resilient TCPS communication.

5.3.3.3 Hybrid AI-assisted PQC

In relation to the rationale for the utilization of an AI-based security augmentation, the authors in [94] address the challenge of securing V2V communications against emerging quantum-capable adversaries, emphasizing that current systems, built around elliptic curve cryptography, will become vulnerable within the operational lifetime of today's vehicles. Because V2V safety messages must be broadcast frequently, processed within milliseconds, and transmitted over very limited radio spectrum, directly

replacing classical signatures with the heavy ones post-quantum schemes is impractical due to their large key and signature sizes. The authors show that existing V2V standards waste substantial bandwidth by repeatedly transmitting certificates that most receivers already possess, with simulations revealing more than 90% redundancy. To overcome these constraints, the paper introduces a two-part solution: 1) An AI-assisted spectrum optimization technique that adaptively reduces redundant certificate transmissions using environmental awareness. 2) A partially hybrid authentication protocol that combines classical and post-quantum signatures in a way tailored for near-term deployment. Hence, instead of fully replacing ECDSA, the hybrid design protects the integrity of short-lived ECDSA signing keys using post-quantum signatures while still allowing basic safety message to be authenticated with classical methods under strict latency requirements. This staged approach supports a phased migration towards quantum-resistant V2V security, aligning with projected timelines for quantum computing advancements. Real-world experiments using software-defined radios, commercial V2V modules, and simulation environments demonstrate that the combined design can be integrated into current IEEE 1609.2 frameworks with negligible delay, approximately 0.39 ms per message, while enabling practical post-quantum protection. In general, the paper provides a feasible roadmap for transitioning V2V ecosystems towards quantum-secure operation without compromising safety, spectrum efficiency, or compatibility with existing vehicles.

5.3.3.4 PQC-Assisted AI systems

Most of the current technological systems are about to incorporate AI with the perspective of being more productive and secure. ITS, in particular, is a sector that is based on AI to a great extent. In addition, there is a lot of discussion about how AI is vulnerable to quantum computers. In the previous Ch. 5.3.3.2 we saw that there are studies on how AI-augmented systems select the appropriate PQC algorithm to be more efficient in constraints environments. In contrast, in [95] the authors highlight the growing importance of incorporating PQC into AI systems, pinpointing where these technologies intersect, how they can be combined, and where they are already proving valuable. By examining practical examples in sectors such as healthcare, finance, edge intelligence, and national defense, we show that adopting quantum-resistant mechanisms is not only technically feasible but essential for safeguarding high-value data and AI-driven decision processes. Nevertheless, moving toward a quantum-secure AI landscape introduces non-trivial obstacles. Increased computational demands, hardware limitations, deployment complexity, and rapidly evolving standards all challenge the path forward. Progress in this area will require collaboration across multiple domains, such as cryptography, AI, systems engineering, regulation, and policy, to design solutions that remain both secure and operationally practical.

Looking towards the future, the resilience of AI ecosystems will rely on continued advancement in efficient PQC primitives, flexible integration architectures, hybrid and transitional cryptographic models, and rigorous threat-aware system design. Equally

crucial will be investments in education, standardization initiatives, and policy frameworks that support the widespread adoption of quantum-safe practices. Ultimately, merging PQC with AI provides a strong foundation for building robust, reliable, and privacy-preserving intelligent systems capable of withstanding future quantum-enabled threats. Acting now to develop and deploy these protections will be vital to ensuring the security and trustworthiness of AI technologies that increasingly underpin modern society.

5.3.3.5 PQC-AI-PQC

The question that arises is which is more important, concerning AV and ITS in general, the AI-assisted PQC or the PQC-Assisted AI. Another point of view is that these levels can be distinguished, and thus the application of AI to PQC and vise versa will not affect the infrastructure in terms of incompatibilities. It is worth thinking that to affect AI is another aspect of attack in vital systems, such as ITS. So, apart from the aforementioned control layers which embody an holistic ITS system, there are internal logical components that are separated into layers.

So, beyond engineering challenges, ethical and social implications become increasingly prominent. AVs may be forced to respond to scenarios that involve unavoidable harm, which underscores the need for transparent and well-defined moral principles within their decision-making systems. Determining who is responsible when an AI-controlled vehicle causes damage is equally complex, as the conventional notion of a human driver being liable no longer applies. Additionally, do not neglect that a safer AI-driven system demands data collection in order to constantly train the agents that are the decision-makers, henceforth. This extensive data collection required for perception and navigation raises questions about **privacy, surveillance**, and potential **misuse** of personal information. Concerns about bias in AI models and the unequal impact of these systems on different user groups further influence how the public perceives and trusts autonomous transportation technologies. In this context, stakeholders should focus on developing a solid product that can serve sensitive systems. This product must be shielded against any kind of attack. In this regard, PQC should protect the data generated during the ITS operation, preserving anonymity and confidentiality, allowing AI to unbiased orchestrate the traffic, and also carefully monitor potential ITS network cyber threats, selecting the appropriate PQC-based protocol, as function of the detected threats, in order to preserve the overall infrastructure's computational cost to tolerable levels.

CHAPTER 6

Conclusions

The systematic studies on quantum mechanics have been evolving the status of quantum computation. Big companies, such as IBM and Google, work on this state-of-the-art project, providing specific future computational capabilities and their corresponding delivery dates. The key factor of this success is the capability to employ as many quantum bits as possible and also to apply fast error correction of the wrong-calculated quantum states. Important sciences, such as medical, will leverage this computational strength, on the other hand, the security of communications is under a clear threat. The Peter Shor and Lov Grover are two significant quantum algorithms that can efficiently threaten well-established cryptographic schemes. So, scientists work upon cryptosystems that can resist against such attacks, an action that born the post-quantum cryptography. Thus, the scope of this thesis is, on one hand, to study the security status of the post-quantum cryprosystems that have been standardized by NIST, on the other, to study how much the employment of these would affect the performance of significant modern environments, such as the blockchain, aviation, and autonomous vehicles as part of the smart cities.

In particular, in Ch. 4 has conducted an experiment to study whether power traces during the decapsulation procedure of two post-quantum algorithms, McEliece and BIKE, can lead to the prediction of secret values. To do so, a low cost apparatus is used to mimic side channel attacks, trading-off the resolution of the traces. Consequently, the null hypothesis of potential leakage could not be declined for both algorithms, providing the potential for machine learning models to predict secret values. More precisely, the models predicted a few bits from the shared secret key generated from the decapsulation process. Among both algorithms, comparing their power waveforms, we can notice that McEliece acts in a more nervous manner, and this behavior may allow the models to extract better knowledge about hidden values. Looking through the literature, we can see that the same behavior exists in the post-quantum digital signature algorithm named Falcon, which leaks significant information. To address this, scientists focus on several techniques in order to hide any potential leakage. So, while post-quantum algorithms have not yet been in service because quantum computers are still in premature stages, there is plenty of time to study post-quantum standards against side channel attacks, and on the other hand to explore alternative algorithms

that can enrich the current post-quantum standards.

Regarding the overhead that post-quantum cryptosystems are supposed to add in several environments, Ch. 5 studied digital signatures and hash functions, revealing that there is indeed such a slight possibility. We must not neglect that these cryptosystems are not a heavier aspect of the pre-quantum systems but systems based on different mathematical problems. Hence, with respect to the blockchain, which is a promising technology that is increasingly employed by several services, the quantum resilient hash functions that work in the Merkle tree, such as SHAKE, add an insignificant, yet predictable overhead. On the contrary, the lattice based digital signatures (Falcon and CRYSTALS-Dilithium) behave in a notable way with respect to the pre-quantum equivalents, allowing us to select them even during the pre-quantum era, as NIST motivates companies to start considering the transition.

Considering though the future aeronautical communications and in particular the specification manual of the LDACS, post-quantum digital signatures, such as the lattice-based, do not add any overhead at all, as they can be transmitted within a single OFDM frame, which is the minimum amount of transmission data. In the same logic, post-quantum cryptography can be currently developed in aircraft radio units, because LDACS is still in heavy production, reducing the maintenance cost of the airliners during the transition to the post-quantum standards.

Finally, a lot of work must also take place in the domain of the ecosystem of future autonomous vehicles. The reason is that this ecosystem is comprised by several layers and services, and thus, to shield communications against quantum attacks, companies should isolate and study all the aspects individually. For that, the use of current technologies, such as 5G or the upcoming 6G, could be a holistic solution as a single mechanism for all types of communication.

APPENDIX A′

Supplementary Material

A′.1 Python Code

In this section is listed the majority parts of the python code, used to extract the knowledge this thesis claims. In order someone to review, use, reproduce, or distribute the code, he may refer to the repository

A′.1.1 Execute KEM Operations On DUT, Store Shared Secret Byte0, Calculate Execution Time

```
1     import RPi.GPIO as GPIO
2 import time
3 import time
4 import datetime
5 import csv
6 import oqs  # Python bindings for liboqs
7
8 # ---------- GPIO CONFIG ----------
9 ENC_PIN = 17   # Raspberry Pi GPIO17 (physical pin 11)
10 DEC_PIN = 27   # Raspberry Pi GPIO27 (physical pin 13)
11
12 # ---------- PQC CONFIG ----------
13 ALG_NAME = "Classic-McEliece-460896"   # change if you want
       other KEM
14 NUM_ITERS = 250   # number of ENC+DEC pairs
15
16 # ---------- Hamming weight ----------
17 def hw8(x: int) -> int:
18     x &= 0xFF
19     return bin(x).count("1")
20
21
```

```
22 def main():
23     # ----- Setup GPIO -----
24     GPIO.setmode(GPIO.BCM)
25     GPIO.setup(ENC_PIN, GPIO.OUT, initial=GPIO.LOW)
26     GPIO.setup(DEC_PIN, GPIO.OUT, initial=GPIO.LOW)
27
28     # ----- Open log file -----
29     leak_file = open(f"{datetime.datetime.now().strftime("%
           Y-%m-%d_t_%H.%M")}-{ALG_NAME}_leak_.csv", "w",
           newline="")
30     writer = csv.writer(leak_file)
31     writer.writerow(["op_id", "phase", "byte0", "hw_byte0",
           "decaps_time"])
32
33
34     print("[*] Initializing KEM:", ALG_NAME)
35
36     # ----- Init KEM -----
37     with oqs.KeyEncapsulation(ALG_NAME) as kem:
38         pk = kem.generate_keypair()
39         print("[*] Keypair generated")
40
41         for op_id in range(NUM_ITERS):
42             # ===== ENC PHASE =====
43             #GPIO.output(ENC_PIN, GPIO.LOW)
44             #time.sleep(1)
45             #GPIO.output(ENC_PIN, GPIO.HIGH)   # trigger
                   high → Pico records ENC trace
46             ct, ss_enc = kem.encap_secret(pk)       # C
                   code inside liboqs
47             #GPIO.output(ENC_PIN, GPIO.LOW)    # trigger
                   low → Pico stops ENC trace
48             #print (ss_enc, ss_enc[0])
49             #b0_enc = ss_enc[0]
50             #hw_enc = hw8(b0_enc)
51             #writer.writerow([op_id, "ENC", b0_enc, hw_enc
                   ])
52             #leak_file.flush()
53
54             # small pause (optional)
55             time.sleep(1)
56
```

```
57              # ===== DEC PHASE =====
58              GPIO.output(DEC_PIN, GPIO.LOW)
59              time.sleep(1)
60              t1 = time.time()
61              GPIO.output(DEC_PIN, GPIO.HIGH)   # trigger
                   high → Pico records DEC trace
62              ss_dec = kem.decap_secret(ct)       # C code
                   inside liboqs
63              GPIO.output(DEC_PIN, GPIO.LOW)    # trigger low
                    → Pico stops DEC trace
64              t2 = time.time()
65              #print (f"decaps time is {t2-t1}")
66              #print (ss_dec, ss_dec[0])
67              b0_dec = ss_dec[0]
68              hw_dec = hw8(b0_dec)
69              writer.writerow([op_id, "DEC", b0_dec, hw_dec,
                   t2-t1])
70              leak_file.flush()
71
72              # optional pause between iterations
73              time.sleep(3)
74
75              print(f"[*] Completed iteration {op_id}/{
                   NUM_ITERS}")
76              #if op_id % 50 == 0:
77              #    print(f"[*] Completed iteration {op_id}/{
                   NUM_ITERS}")
78
79      leak_file.close()
80      GPIO.output(ENC_PIN, GPIO.LOW)
81      GPIO.output(DEC_PIN, GPIO.LOW)
82      GPIO.cleanup()
83      print(f"[*] Done, log file written.")
84
85
86  if __name__ == "__main__":
87      main()
```

A′.1.2 Compute Correlation Between Leakage and Traces

```
1 import pandas as pd
2 import numpy as np
```

```
import matplotlib.pyplot as plt

# Load your uploaded file
df = pd.read_excel("PQC_ML_Analysis.xlsx", sheet_name="BIKE
    -L3")

# Extract trace samples (s0..s1499)
sample_cols = [c for c in df.columns if c.startswith("s")]
X = df[sample_cols].to_numpy(dtype=float)
#y = df["hw_byte0"].to_numpy()

y = (df["byte0"] & bits ).to_numpy()
N = min(200, len(y))
X = X[:N, :]
y = y[:N]

# Compute correlation between each sample and HW label
corrs = [np.corrcoef(X[:, i], y)[0, 1] for i in range(X.
    shape[1] )] #

# Plot
plt.figure(figsize=(12,4))
plt.plot(corrs)
plt.title(f"{SHEET_NAME}: Correlation between {bin(bits).
    count("1")} bits of shared secret byte0 and {N} leak
    traces")
plt.xlabel(f"Samples ({len(sample_cols)}/trace) ")
plt.ylabel("Correlation ρ")
plt.grid(True)
plt.tight_layout()
plt.show()
```

A′.1.3 Machine Learning: Predicts Secret Key Bits

```
print("[*] Building models...")

models = []

models.append(("LogisticRegression",
               LogisticRegression(max_iter=2000)))

models.append(("SVM-RBF",
```

```
9                 SVC(kernel='rbf', C=2, gamma='scale')))
10
11 models.append(("RandomForest",
12                 RandomForestClassifier(n_estimators=400,
                      max_depth=12)))
13
14 models.append(("kNN",
15                 KNeighborsClassifier(n_neighbors=3)))
16
17 models.append(("MLP",
18                 MLPClassifier(hidden_layer_sizes=(100, 50),
                      max_iter=500)))
19 '''
20 models.append(("Lightgbm",
21                     LGBMClassifier(
22                         num_leaves=31,
23                         learning_rate=0.05,
24                         n_estimators=600
25                     )))
26 '''
27 models.append(("LinearSVM", SVC(kernel='linear')))
28 models.append(("AdaBoost", AdaBoostClassifier()))
29
30 # ====================================
31 #  LOOP THROUGH ALL MODELS
32 # ====================================
33 file = open (f"{SHEET_NAME}_ML_Results.pdf", "w")
34 for name, clf in models:
35     print("\n==============================")
36     print(f"Training classifier: {name}")
37     print("==============================")
38
39     clf.fit(X_train, y_train)
40     pred = clf.predict(X_test)
41
42     print(f"\n>>> Accuracy ({name}): {accuracy_score(y_test
          , pred):.3f}")
43     file.write (f"\n>>> Accuracy ({name}): {accuracy_score(
          y_test, pred):.3f}" )
44     print(classification_report(y_test, pred))
45     file.write(f" {classification_report(y_test, pred)}")
46     print("Confusion matrix:")
```

```
47     print(confusion_matrix(y_test, pred))
48     file.write(f"\nConfusion matrix:")
49     file.write(f"\n{confusion_matrix(y_test, pred)}")
50 file.close()
```

A′.1.4 MicoPython (On Microcontroller Pico) Script Collecting and Sending Traces to Computer

```
1  from machine import ADC, Pin
2  #import utime
3  #import time
4  import sys
5
6  ADC_PIN = 26 #Analog to Digital converter
7  TRIG_PIN = 15 #Trigger pin
8  N_SAMPLES = 1500
9  VREF = 3.3
10
11 adc = ADC(ADC_PIN)
12 trigger = Pin(TRIG_PIN, Pin.IN, Pin.PULL_DOWN)
13
14
15 trace_id = 0
16
17 while True:
18
19     while trigger.value() == 0:
20         pass
21
22     trace_buf = [0] * N_SAMPLES #Reinitiate trace buffer
23
24     #t_start = utime.ticks_us()
25
26     i = 0
27     while trigger.value() == 1 and i < N_SAMPLES:
28         trace_buf[i] = adc.read_u16()
29         i += 1
30
31     #t_end = utime.ticks_us()
32
33     volts = [f"{(v * VREF)/65535.0:.5f}" for v in trace_buf
           ]
```

```
34      row = [str(trace_id)] + volts
35
36      sys.stdout.write(",".join(row) + "\n")
37      #sys.stdout.flush()
38
39      trace_id += 1
40
41      while trigger.value() == 1:
42          pass
```

A′.1.5 Bash Script Synchronizing Communication Between Equipment

```
1 #!/usr/bin/env bash
2
3 ALG="\$1"SAMPLES=\$2
4
5 OUTFILE="\$(date +\%Y-\%m-\%d\_%H-\%M-\%S).\$ALG.traces.csv
    "header="trace\_id"for ((i=0; i<\$SAMPLES; i++));
    doheader="\$header,s\$i"doneecho "\$header" > \$OUTFILE
6
7 echo "Waiting pico to connect.."while [ -z "\$(ls /dev/cu.
    usbmodem* 2>/dev/null | head -n 1)" ]dosleep 0.5sdone
8
9 PORT=\$(ls /dev/cu.usbmodem* 2>/dev/null | head -n 1)
10
11 echo ""echo "Connected... Press Ctrl+C to stop or
    disconnect pico.." echo ""
12
13 \# Set serial port parameters (macOS uses -f)stty -f "\
    $PORT" 115200 cs8 -cstopb -parenb -ixon -ixoff -crtscts
14
15 \# Read everything from Pico and save to file
16
17 cat "\$PORT" | tee "\$OUTFILE"
18
19 echo "Saving to : \$OUTFILE"echo ""
```

Bibliography

[1] Daemen, Joan, and Vincent Rijmen. "AES proposal: Rijndael." (1999).

[2] Jindal, Poonam, and Brahmjit Singh. "RC4 encryption-A literature survey." Procedia Computer Science 46 (2015): 697-705.

[3] Milanov, Evgeny. "The RSA algorithm." *RSA laboratories* 1.11 (2009).

[4] Levy, Sharon. "Performance and Security of ECDSA." Comput. Sci (2015).

[5] Johnson, Don, Alfred Menezes, and Scott Vanstone. "The elliptic curve digital signature algorithm (ECDSA)." International journal of information security 1.1 (2001): 36-63.

[6] Ladd, Thaddeus D., et al. "Quantum computers." nature 464.7285 (2010): 45-53.

[7] LaPierre, Ray. "Shor algorithm." Introduction to quantum computing. Cham: Springer International Publishing, 2021. 177-192.

[8] Jozsa, Richard. "Searching in Grover's algorithm." arXiv preprint quant-ph/9901021 (1999).

[9] Dam, Duc-Thuan, et al. "A survey of post-quantum cryptography: Start of a new race." Cryptography 7.3 (2023): 40.

[10] Ekerå, Martin, and Johan Håstad. "Quantum algorithms for computing short discrete logarithms and factoring RSA integers." International Workshop on Post-Quantum Cryptography. Cham: Springer International Publishing, 2017.

[11] Silverman, Joseph H., and Joe Suzuki. "Elliptic curve discrete logarithms and the index calculus." International conference on the theory and application of cryptology and information security. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998.

[12] Sadeghi-Nasab, A., Rafe, V. A comprehensive review of the security flaws of hashing algorithms. J Comput Virol Hack Tech 19, 287–302 (2023). https://doi.org/10.1007/s11416-022-00447-w

[13] Wang, Ping, et al. "Quantum algorithms for hash preimage attacks." *Quantum Engineering* 2.2 (2020): e36.

[14] Moody, Dustin. "Nist pqc standardization update." National Institute of Standards and Technology (2021): 2021-10.

[15] Peikert, Chris. "A decade of lattice cryptography." Foundations and trends® in theoretical computer science 10.4 (2016): 283-424.

[16] Avanzi, Roberto, et al. "Crystals-kyber." NIST, Tech. Rep (2017).

[17] Lyubashevsky, Vadim, et al. "Crystals-dilithium." Algorithm Specifications and Supporting Documentation 2 (2020).

[18] Prest, Thomas, et al. "Falcon." Post-Quantum Cryptography Project of NIST (2020).

[19] Weger, Violetta, Niklas Gassner, and Joachim Rosenthal. "A survey on code-based cryptography." arXiv preprint arXiv:2201.07119 (2022).

[20] Biswas, Bhaskar, and Nicolas Sendrier. "McEliece cryptosystem implementation: Theory and practice." International Workshop on Post-Quantum Cryptography. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[21] Nosouhi, Mohammad Reza, et al. "Bit flipping key encapsulation for the post-quantum era." IEEE Access 11 (2023): 56181-56195.

[22] Melchor, Carlos Aguilar, et al. "Hamming quasi-cyclic (HQC)." NIST PQC Round 2.4 (2018): 13.

[23] Bernstein, Daniel J., et al. "The SPHINCS+ signature framework." Proceedings of the 2019 ACM SIGSAC conference on computer and communications security. 2019.

[24] Huang, Yun-Ju, Feng-Hao Liu, and Bo-Yin Yang. "Public-key cryptography from new multivariate quadratic assumptions." *International Workshop on Public Key Cryptography*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[25] De Feo, Luca. "Mathematics of isogeny based cryptography." *arXiv preprint arXiv:1711.04062* (2017).

[26] Wang, Anyu, Dianyan Xiao, and Yang Yu. "Lattice-based cryptosystems in standardisation processes: A survey." *IET Information Security* 17.2 (2023): 227-243.

[27] Peters, Christiane. "Information-set decoding for linear codes over F q." *International Workshop on Post-Quantum Cryptography*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

[28] Kachigar, Ghazal, and Jean-Pierre Tillich. "Quantum information set decoding algorithms." *International Workshop on Post-Quantum Cryptography*. Cham: Springer International Publishing, 2017.

[29] Jayashree Dey and Ratna Dutta. 2023. Progress in Multivariate Cryptography: Systematic Review, Challenges, and Research Directions. ACM Comput. Surv. 55, 12, Article 246 (December 2023), 34 pages. https://doi.org/10.1145/3571071

[30] Mishra, Sweta, Bhaskar Mondal, and Rishi Kumar Jha. "A survey on isogeny-based cryptographic protocols." *Wireless Networks*31.3 (2025): 2993-3024.

[31] Le, Thanh-Ha, Cécile Canovas, and Jessy Clédiere. "An overview of side channel analysis attacks." Proceedings of the 2008 ACM symposium on Information, computer and communications security. 2008.

[32] Lahr, Norman, et al. "Side channel information set decoding using iterative chunking: Plaintext recovery from the "Classic McEliece" hardware reference implementation." *International Conference on the Theory and Application of Cryptology and Information Security*. Cham: Springer International Publishing, 2020.

[33] Zhang, Jiliang, et al. "Timing side-channel attacks and countermeasures in CPU microarchitectures." *ACM Computing Surveys* 56.7 (2024): 1-40.

[34] Molter, H. Gregor, et al. "A simple power analysis attack on a McEliece cryptoprocessor." *Journal of Cryptographic Engineering* 1.1 (2011): 29-36.

[35] Bottinelli, Paul, and Joppe W. Bos. "Computational aspects of correlation power analysis." Journal of Cryptographic Engineering 7.3 (2017): 167-181.

[36] E. Karabulut and A. Aysu, "FALCON Down: Breaking FALCON Post-Quantum Signature Scheme through Side-Channel Attacks," 2021 58th ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 2021, pp. 691-696, doi: 10.1109/DAC18074.2021.9586131.

[37] Barthe, Gilles, Benjamin Grégoire, and Vincent Laporte. "Secure compilation of side-channel countermeasures: the case of cryptographic "constant-time"." 2018 IEEE 31st Computer Security Foundations Symposium (CSF). IEEE, 2018.

[38] Carlet, Claude, et al. "A masking method based on orthonormal spaces, protecting several bytes against both SCA and FIA with a reduced cost." Journal of Cryptographic Engineering 14.2 (2024): 223-240.

[39] Dobias, Patrik, et al. "SoK: Reassessing Side-Channel Vulnerabilities and Countermeasures in PQC Implementations." Cryptology ePrint Archive (2025).

[40] Gan, Peizhou, et al. "Classic McEliece hardware implementation with enhanced side-channel and fault resistance." Cryptology ePrint Archive (2024).

[41] Zheng, Zibin, et al. "Blockchain challenges and opportunities: A survey." International journal of web and grid services 14.4 (2018): 352-375.

[42] Gervais, Arthur, et al. "On the security and performance of proof of work blockchains." Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. 2016.

[43] Artem Barger, Vladimir Gorgadze, and Anastasiia Sanina. 2025. Enhancing State Integrity and Validation in Hyperledger Fabric with Certification Blocks and Patricia Merkle Tries. In Proceedings of the 2024 7th International Conference on Blockchain Technology and Applications (ICBTA '24). Association for Computing Machinery, New York, NY, USA, 11–18. https://doi.org/10.1145/3708622.3708624

[44] H. Liu, X. Luo, H. Liu and X. Xia, "Merkle Tree: A Fundamental Component of Blockchains," 2021 International Conference on Electronic Information Engineering and Computer Science (EIECS), Changchun, China, 2021, pp. 556-561, doi: 10.1109/EIECS53707.2021.9588047.

[45] E. F. Codd. 1970. A relational model of data for large shared data banks. Commun. ACM 13, 6 (June 1970), 377–387. https://doi.org/10.1145/362384.362685

[46] Khan, W.; Kumar, T.; Zhang, C.; Raj, K.; Roy, A.M.; Luo, B. SQL and NoSQL Database Software Architecture Performance Analysis and Assessments—A Systematic Literature Review. Big Data Cogn. Comput. 2023, 7, 97. https://doi.org/10.3390/bdcc7020097

[47] Chauhan, Anjali. "A review on various aspects of mongodb databases." International Journal of Engineering Research & Technology (IJERT) 8.05 (2019): 90-92.

[48] Sankagiri, Suryanarayana, et al. "Blockchain cap theorem allows user-dependent adaptivity and finality." International Conference on Financial Cryptography and Data Security. Berlin, Heidelberg: Springer Berlin Heidelberg, 2021.

[49] Androulaki, Elli, et al. "Hyperledger fabric: a distributed operating system for permissioned blockchains." Proceedings of the thirteenth EuroSys conference. 2018.

[50] Cachin, Christian. "Architecture of the hyperledger blockchain fabric." Workshop on distributed cryptocurrencies and consensus ledgers. Vol. 310. No. 4. 2016.

[51] Yli-Huumo, Jesse, et al. "Where is current research on blockchain technology?—a systematic review." PloS one 11.10 (2016): e0163477.

[52] Singhal, Saurabh. "The Rise of "Big Data" on Cloud Computing." Big Data Analysis for Green Computing. CRC Press, 2021. 39-51.

[53] N. Banothu, S. Bhukya and K. V. Sharma, "Big-data: Acid versus base for database transactions," 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, India, 2016, pp. 3704-3709, doi: 10.1109/ICEEOT.2016.7755401.

[54] Sai, A. R., Das, A., & Chen, G. (2021). *Taxonomy of Centralization in Public Blockchain Systems: A Systematic Literature Review. Information Processing & Management,* 58(4), 102584. https://doi.org/10.1016/j.ipm.2021.102584

[55] Homoliak, I., & Szalachowski, P. (2020). *Aquareum: A Centralized Ledger Enhanced with Blockchain and Trusted Computing. arXiv preprint* arXiv:2005.13339. https://arxiv.org/abs/2005.13339

[56] Islam, Siful, and Kutub Uddin Apu. "Decentralized vs. Centralized database solutions in blockchain: advantages, challenges, and use cases." Global Mainstream Journal of Innovation, Engineering & Emerging Technology 3.4 (2024): 58-68.

[57] Atzori, M. (2017). Blockchain Technology and Decentralized Governance: Is the State Still Necessary? Journal of Governance and Regulation, 6(1), 45–62. https://virtusinterpress.org/IMG/pdf/10.22495_jgr_v6_i1_p5.pdf

[58] Zerui Ge, Dumitrel Loghin, Beng Chin Ooi, Pingcheng Ruan, and Tianwen Wang. 2022. Hybrid blockchain database systems: design and performance. Proc. VLDB Endow. 15, 5 (January 2022), 1092–1104. https://doi.org/10.14778/3510397.3510406

[59] Kreps, Jay, Neha Narkhede, and Jun Rao. "Kafka: A distributed messaging system for log processing." Proceedings of the NetDB. Vol. 11. No. 2011. 2011.

[60] Fabian, Velucian I., et al. "Investigating Suitable Consensus Protocols for Secured Blockchain Based System." Journal of Computer and Communications 13.7 (2025): 204-222.

[61] Xu, R., Yin, S., & Zhang, J. (2020). *Hybrid Blockchain-Enabled Secure Microservices Fabric for Decentralized Multi-Domain Avionics Systems. arXiv preprint* arXiv:2004.10674. https://arxiv.org/abs/2004.10674

[62] Alkhateeb, A.; Catal, C.; Kar, G.; Mishra, A. Hybrid Blockchain Platforms for the Internet of Things (IoT): A Systematic Literature Review. Sensors 2022, 22, 1304. https://doi.org/10.3390/s22041304

[63] H. Liu, X. Luo, H. Liu and X. Xia, "Merkle Tree: A Fundamental Component of Blockchains," 2021 International Conference on Electronic Information Engineering and Computer Science (EIECS), Changchun, China, 2021, pp. 556-561, doi: 10.1109/EIECS53707.2021.9588047.

[64] Merkle, R.C. (1988). A Digital Signature Based on a Conventional Encryption Function. In: Pomerance, C. (eds) Advances in Cryptology — CRYPTO '87. CRYPTO 1987. Lecture Notes in Computer Science, vol 293. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-48184-2_32

[65] Gracy, M. & Jeyavadhanam, B.. (2022). MTTBA- A Key Contributor for Sustainable Energy Consumption Time and Space Utility for Highly Secured Crypto Transactions in Blockchain Technology. 10.48550/arXiv.2209.13431.

[66] Kermezis, G., Limniotis, K., Kolokotronis, N. (2021). User-Generated Pseudonyms Through Merkle Trees. In: Gruschka, N., Antunes, L.F.C., Rannenberg, K., Drogkaris, P. (eds) Privacy Technologies and Policy. APF 2021. Lecture Notes in Computer Science(), vol 12703. Springer, Cham. https://doi.org/10.1007/978-3-030-76663-4_5

[67] Zeyad A. Al-Odat, Samee U. Khan, Eman Al-Qtiemat, A modified secure hash design to circumvent collision and length extension attacks, Journal of Information Security and Applications, Volume 71,2022,103376,ISSN 2214-2126,https://doi.org/10.1016/j.jisa.2022.103376.

[68] Pan, Shimin, et al. "Multi-signatures for ECDSA and Its Applications in Blockchain." Australasian Conference on Information Security and Privacy. Cham: Springer International Publishing, 2022.

[69] N. Kolokotronis, S. Brotsis, G. Germanos, C. Vassilakis and S. Shiaeles, "On Blockchain Architectures for Trust-Based Collaborative Intrusion Detection," 2019 IEEE World Congress on Services (SERVICES), Milan, Italy, 2019, pp. 21-28, doi: 10.1109/SERVICES.2019.00019.

[70] W. Y. Leong, Y. Z. Leong and W. S. Leong, "Enhancing Blockchain Security," 2024 IEEE Symposium on Wireless Technology & Applications (ISWTA), Kuala Lumpur, Malaysia, 2024, pp. 108-112, doi: 10.1109/ISWTA62130.2024.10651753.

[71] Coron, Jean-Sébastien, et al. "Merkle-Damgård revisited: How to construct a hash function." Annual International Cryptology Conference. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.

[72] Chang, Shu-jen, et al. "Third-round report of the SHA-3 cryptographic hash algorithm competition." NIST Interagency Report 7896 (2012): 121.

[73] Preneel, Bart. "The state of hash functions and the NIST SHA-3 competition." *International Conference on Information Security and Cryptology*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[74] D. S. Crescenzo *et al.*, "Hardware Accelerator for FIPS 202 Hash Functions in Post-Quantum Ready SoCs," *2024 IEEE 30th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, Rennes, France, 2024, pp. 1-6, doi: 10.1109/IOLTS60994.2024.10616067.

[75] Lindell, Y. Fast Secure Two-Party ECDSA Signing. J Cryptol 34, 44 (2021). https://doi.org/10.1007/s00145-021-09409-9

[76] Proos, John, and Christof Zalka. "Shor's discrete logarithm quantum algorithm for elliptic curves." arXiv preprint quant-ph/0301141 (2003).

[77] Brotsis, S., N. Kolokotronis, and K. Limniotis. "Towards post-quantum blockchain platforms." Security Technologies and Methods for Advanced Cyber Threat Intelligence, Detection and Mitigation (2022): 106-130.

[78] Roy, Aloke. "Secure aircraft communications addressing and reporting system (ACARS)." 20th DASC. 20th Digital Avionics Systems Conference (Cat. No. 01CH37219). Vol. 2. IEEE, 2001.

[79] Kamali, Behnam. "AeroMACS: An IEEE 802.16 Standard-based Technology for the Next Generation of Air Transportation Systems." (2018).

[80] Schnell, Michael, et al. "LDACS: Future aeronautical communications for air-traffic management." IEEE Communications Magazine 52.5 (2014): 104-110.

[81] Musa, Sarhan M., and Zhijun Wu, eds. Aeronautical Telecommunications Network: Advances, Challenges, and Modeling. CRC Press, 2015.

[82] Van Houtte, Ben. "The Single European Sky–EU Reform of ATM." European Air Traffic Management. Routledge, 2016. 200-217.

[83] Khan, Suleman, et al. "Enhancing cybersecurity for LDACS: A secure and lightweight mutual authentication and key agreement protocol." 2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC). IEEE, 2023.

[84] Spalas, Konstantinos, and Nicholas Kolokotronis. "Post-Quantum Security Evaluation of Aeronautical Communications." 2025 IEEE International Conference on Cyber Security and Resilience (CSR). IEEE, 2025.

[85] Liu, Pengtao, et al. "A Secure and Low-Latency Design for Ldacs Ground Station Handover." 2025 Integrated Communications, Navigation and Surveillance Conference (ICNS). IEEE, 2025.

[86] https://www.ldacs.com/wp-content/uploads/2023/03/SESAR2020_PJ14-W2-60_TRL6_D3_1_230_3rd_LDACS_AG_Specification_v1.0.0.pdf

[87] Orasch, Sergio, et al. "Development of a L-Band Digital Aeronautical Communications System (LDACS) Framework." 2024 47th MIPRO ICT and Electronics Convention (MIPRO). IEEE, 2024.

[88] Ryu, Jongseok, et al. "Design of secure mutual authentication scheme for metaverse environments using blockchain." Ieee Access 10 (2022): 98944-98958.

[89] Biswas, Anushka, and Hwang-Cheng Wang. 2023. "Autonomous Vehicles Enabled by the Integration of IoT, Edge Intelligence, 5G, and Blockchain" *Sensors* 23, no. 4: 1963. https://doi.org/10.3390/s23041963

[90] G. Bendiab, A. Hameurlaine, G. Germanos, N. Kolokotronis and S. Shiaeles, "Autonomous Vehicles Security: Challenges and Solutions Using Blockchain and Artificial Intelligence," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 4, pp. 3614-3637, April 2023, doi: 10.1109/TITS.2023.3236274

[91] Abdullah Al Mamun, Akid Abrar, Mizanur Rahman, et al. Post-Quantum Cryptography for Intelligent Transportation Systems: An Implementation-Focused Review.*TechRxiv.* November 21, 2025.

[92] Xu, Dongyang, et al. "Post-quantum authentication against cyber-physical attacks in v2x-based autonomous vehicle platoon." IEEE Transactions on Intelligent Transportation Systems 25.6 (2023): 5034-5044.

[93] Abrar, Akid, et al. "AI-Driven Post-Quantum Cryptography for Cyber-Resilient V2X Communication in Transportation Cyber-Physical Systems." arXiv preprint arXiv:2510.08496 (2025).

[94] Twardokus, Geoff, Bindel, Nina, Rahbari, Hanif, and McCarthy, Sarah. When Cryptography Needs a Hand: Practical Post-Quantum Authentication for V2V Communications. Retrieved from https://par.nsf.gov/biblio/10492326. Network and Distributed System Security Symposium (NDSS 2024) . Web. doi:10.14722/ndss.2024.24267.

[95] Danny Smith, Akinniyi James Samuel, Post-Quantum Cryptography: Securing AI Systems against Quantum Threats, DOI: https://doi.org/10.64206/snz0jq38