



BANCA D'ITALIA
EUROSISTEMA

Mercati, infrastrutture, sistemi di pagamento

(Markets, Infrastructures, Payment Systems)

A practical implementation of a quantum-safe PKI
in a payment systems environment

by Luca Buccella and Stefano Massi



BANCA D'ITALIA
EUROSISTEMA

Mercati, infrastrutture, sistemi di pagamento

(Markets, Infrastructures, Payment Systems)

A practical implementation of a quantum-safe PKI
in a payment systems environment

by Luca Buccella and Stefano Massi

Number 64 – September 2025

The papers published in the 'Markets, Infrastructures, Payment Systems' series provide information and analysis on aspects regarding the institutional duties of the Bank of Italy in relation to the monitoring of financial markets and payment systems and the development and management of the corresponding infrastructures in order to foster a better understanding of these issues and stimulate discussion among institutions, economic actors and citizens.

The views expressed in the papers are those of the authors and do not necessarily reflect those of the Bank of Italy.

The series is available online at www.bancaditalia.it.

*Printed copies can be requested from the Paolo Baffi Library:
richieste.pubblicazioni@bancaditalia.it.*

Editorial Board: STEFANO SIVIERO, PAOLO DEL GIOVANE, MASSIMO DORIA,
GIUSEPPE ZINGRILLO, PAOLO LIBRI, GUERINO ARDIZZI, PAOLO BRAMINI, FRANCESCO COLUMBA,
LUCA FILIDI, TIZIANA PIETRAFORTE, ALFONSO PUORRO, ANTONIO SPARACINO.

Secretariat: YI TERESA WU.

ISSN 2724-6418 (online)
ISSN 2724-640X (print)

Banca d'Italia
Via Nazionale, 91 - 00184 Rome - Italy
+39 06 47921

Designed and printing by the Printing and Publishing Division of the Bank of Italy

A PRACTICAL IMPLEMENTATION OF A QUANTUM-SAFE PKI IN A PAYMENT SYSTEMS ENVIRONMENT

by Luca Buccella* and Stefano Massi*

Abstract

The security of the digital certification services provided by Public Key Infrastructures (PKIs) is essential to ensure the proper functioning of payment systems and financial market infrastructures.

This work aims to identify the main challenges and critical issues involved in the transition towards a quantum-resistant PKI architecture, capable of withstanding attacks carried out by quantum computers with sufficient computational capabilities to break into the classical cryptographic schemes currently used. To this end, a Proof of Concept (PoC) is presented, based on quantum-safe cryptographic algorithms designed to ensure the long-term security and resilience of payment systems in a post-quantum era.

The results of the PoC indicate that, although the performance of the new algorithms proved satisfactory and may facilitate the transition process, current PKI software solutions have not yet achieved full compatibility with the quantum-safe algorithms available on the market. Further efforts are therefore required to adapt existing solutions to the emerging standards, in order to promote the widespread adoption of robust cryptographic algorithms.

Keywords: Quantum computing, Quantum security, PKI, Target Services.

Sintesi

La sicurezza dei servizi di certificazione digitale erogati dalle *Public Key Infrastructure* (PKI) è fondamentale per garantire il corretto funzionamento dei sistemi di pagamento e delle infrastrutture dei mercati finanziari.

L'obiettivo del lavoro è identificare le principali sfide e criticità nella transizione verso un'architettura PKI quantum-resistant, in grado di resistere ad attacchi condotti mediante calcolatori quantistici dotati di capacità computazionali sufficienti a violare gli schemi crittografici classici attualmente in uso. Viene a tal fine presentata una Proof of Concept (PoC) di una PKI basata su algoritmi crittografici quantum-safe, disegnati per garantire la sicurezza e la resilienza a lungo termine dei sistemi di pagamento in un'era post quantistica.

I risultati della PoC indicano che, sebbene le prestazioni dei nuovi algoritmi si siano dimostrati soddisfacenti e potranno quindi facilitare il processo di transizione, le soluzioni software per PKI non hanno ancora raggiunto un livello di compatibilità pienamente soddisfacente con gli algoritmi quantum safe al momento disponibili sul mercato. È pertanto necessario intensificare gli sforzi per l'adeguamento delle soluzioni ai nuovi standard, al fine di promuovere la pervasiva adozione di algoritmi crittografici robusti.

* Banca d'Italia, IT Development Directorate.

CONTENTS

1. Introduction	7
2. Quantum Threats and Quantum Safe Cryptographic Algorithms	9
2.1. <i>Threats</i>	10
2.2. <i>Risk mitigation</i>	11
2.3. <i>Algorithms, size and efficiency</i>	13
3. Public Key Infrastructures and Payment Systems	17
3.1. <i>Description of certificates usage in ESMIG</i>	17
3.2. <i>Certificates automation with ACME protocol in T2</i>	18
3.3. <i>PKI Security Threats</i>	19
4. The Proof-of-Concept conducted in the Bank of Italy	20
4.1. <i>Hybrid Certificates</i>	20
4.2. <i>Certification Authorities</i>	23
4.3. <i>Testing the use of certificates</i>	24
5. Performance Tests	28
5.1. <i>Dimensions</i>	28
5.2. <i>Execution time</i>	29
6. Conclusions	34
6.1. <i>Crypto-agility of payment systems</i>	35
7. Developments in the short and medium term	36
8. Definitions	38
Index of figures	40
Bibliography	41

1. Introduction¹

European payment system infrastructures such as TARGET2 (T2), TARGET2-Securities (T2S), and TARGET Instant Payment Settlement (TIPS) rely on advanced cryptographic mechanisms as the foundation of their IT security. These mechanisms are primarily built upon secure communication protocols, which are currently considered robust. However, the advent of quantum technologies – capable of implementing processors with dramatically increased computational power – poses a significant threat to their long-term security.

T2, T2S, and TIPS are core components of the European financial infrastructure, supporting the processing and settlement of financial transactions across the continent. Encryption is central to safeguarding sensitive information and preventing unauthorized access. These platforms predominantly employ public key infrastructures (PKIs) for authentication and symmetric cryptography for data confidentiality during transmission.

TARGET2-Securities (T2S) leverages end-to-end encryption and implements the Data Exchange Protocol (DEP), which offers advanced mechanisms to secure communication between settlement systems and participants. DEP ensures that data are encrypted and accessible only to authorized recipients, effectively mitigating the risks of eavesdropping and tampering.

TARGET2 (T2) employs protocols such as Transport Layer Security (TLS) to protect communications between central banks and participants. It also utilizes digital signatures based on RSA and SHA-256 algorithms to ensure message integrity and data security during transmission. Additionally, T2 incorporates DEP to further enhance confidentiality by ensuring that encrypted data can only be accessed by designated entities.

TARGET Instant Payment Settlement (TIPS) relies on advanced cryptographic algorithms such as AES-256 for symmetric encryption, enabling secure real-time transaction processing. This ensures that exchanged data remain protected against unauthorized access and manipulation.

The adoption of these cryptographic technologies is essential for maintaining the trust, integrity, and stability of the European financial system. However, the emergence of quantum computing technologies necessitates the evolution of existing cryptographic protocols to preserve high security standards in a rapidly changing threat landscape.

Numerous studies have highlighted the vulnerabilities of *conventional* cryptographic algorithms in the face of quantum computing and have proposed strategies to mitigate these risks in operational settings. In this paper, we adopt a pragmatic approach by implementing a Public Key Infrastructure (PKI) that supports quantum-safe algorithms. We evaluate its practical applicability through a real-world proof-of-concept (PoC), assessing both effectiveness and performance.

Our PoC reveals that compatibility with quantum-safe algorithms remains limited in most current software environments, with the notable exception of hybrid certificates – which are available but not yet widely or properly utilized. Nonetheless, the performance metrics of quantum-safe algorithms indicate a high degree of computational efficiency. Among the algorithms analyzed, *Dilithium* emerges as the most balanced solution, offering an optimal trade-off between memory usage and computational cost.

The remainder of this paper is structured as follows. Section 2 introduces the quantum threat landscape and provides an overview of quantum-safe cryptographic algorithms and their evolution.

¹ We are grateful to Andrea Billet, Alessandro Casale, Damiano Diego De Felice, Simone Dutto, Sergio Polese and Giordano Santilli of the Italian National Cybersecurity Agency.

Section 3 discusses the role of PKIs in payment systems. Section 4 presents the implementation and architecture of the PoC. Section 5 details the performance evaluation results. Section 6 draws the main conclusions. Section 7 outlines potential directions for further development, and Section 8 provides a glossary of technical definitions.

2. Quantum Threats and Quantum Safe Cryptographic Algorithms

From a technical standpoint, the security of communication protocols – specifically with regard to confidentiality and data integrity – relies fundamentally on three core functionalities: encryption, key exchange, and digital signatures. In particular, asymmetric cryptographic schemes, in which two parties communicate using a pair of keys (a public key and a corresponding private key), derive their security from the computational hardness of mathematical problems such as integer factorization and the discrete logarithm problem.

These problems are considered intractable with classical computing, as no known algorithms can solve them efficiently – i.e., in polynomial time – on conventional hardware. However, this security assumption is fundamentally challenged by the advent of quantum computing. Quantum computers, which leverage the quantum mechanical properties of matter for data representation and computation, are capable of executing algorithms that outperform classical approaches for specific problem classes.

A notable example is Shor's algorithm, which can efficiently factor large integers and compute discrete logarithms in polynomial time [1], thereby undermining the foundational security assumptions of widely used asymmetric cryptographic schemes such as RSA and ECC. Consequently, any encrypted communication that relies on these schemes could be decrypted if intercepted and later processed by a sufficiently powerful quantum computer.

Similarly, symmetric encryption is also affected by quantum computing, albeit to a lesser extent. Grover's algorithm offers a quadratic speed-up for searching an unstructured space, which effectively reduces the security strength of symmetric key cryptosystems by half (e.g., AES-256 would provide a quantum equivalent of 128-bit security).

Recognizing these threats, the National Institute of Standards and Technology (NIST) highlighted as early as 2016 [2] that most commonly used cryptographic mechanisms – including those for encryption, key exchange, and digital signatures – will no longer be considered secure once large-scale quantum computers become available. This recognition has since led to a global initiative to develop and standardize post-quantum cryptography capable of resisting attacks from quantum adversaries.

Table 1 - Cryptographic algorithms and impacts of Quantum Computing

Cryptographic algorithm	Type	Purpose	Impacts of Quantum Computing on a large scale
AES	Symmetric key	Encryption	Larger key sizes needed
SHA-2, SHA-3	-	Hash functions	Larger output needed
RSA	Public key	Signature Key establishment	No longer secure
ECDSA, ECDH (ELLIPTIC CURVE CRYPTOGRAPHY)	Public key	Signature Key exchange	No longer secure
DSA (FINITE FIELD CRYPTOGRAPHY)	Public key	Signature Key exchange	No longer secure

Given the progress made in recent years in the area of research on quantum computers, there has been an impetus for studies in the area of so-called Post Quantum Cryptography (PQC), in order to identify algorithms that are robust and resistant, and to this end, NIST has initiated a process of evaluating

and selecting new algorithms, defined as quantum safe or quantum resistant, with the aim of standardizing their use.

To normalize the concept of security and define a yardstick to compare the robustness of cryptographic algorithms, including *quantum safe* ones, NIST defined five *levels of security* [3].

Table 2 - Description of NIST security levels [4]

Level	Security Description
I	At least as hard to break as AES128 (exhaustive key search)
II	At least as hard to break as SHA256 (collision search)
III	At least as hard to break as AES192 (exhaustive key search)
IV	At least as hard to break as SHA384 (collision search)
V	At least as hard to break as AES256 (exhaustive key search)

2.1. Threats

One of the significant threats posed by the advent of quantum computers is the so-called *harvest now, decrypt later* strategy. Also known as *store now, decrypt later* (see Figure 1), this approach involves the acquisition and long-term storage of encrypted data that is currently unreadable, with the anticipation that future advancements in decryption techniques will render it accessible.



Figure 1 - Harvest now, decrypt later

This threat manifests in two primary forms: *Grover's* algorithm and *Shor's* algorithm.

Grover's algorithm [4] significantly reduces the time required to attack symmetric encryption systems, thereby lowering their security level. On the other hand, *Shor's* algorithm compromises the security of modern asymmetric cryptographic algorithms, such as the factoring of large prime numbers and the discrete logarithm problem, by reducing the computational complexity from exponential to polynomial time. This reduction in complexity simplifies the decryption of messages encrypted with these asymmetric schemes.

As described by Grover [5] and Brassard [6], substantial performance improvements can be achieved in solving problems related to 'Searching' and 'Collision Finding and Element Distinctness' respectively:

- $\mathcal{O}(\sqrt{n})$ for search algorithms,² where n is the size of admissible inputs; classical algorithms take a time of $\mathcal{O}(n)$;
- $\mathcal{O}(\sqrt[3]{n})$ for collision search algorithms, where n is the domain size.

Table 3 - Asymptotic notation for evaluating algorithms

	Time	Space
Grover	$\mathcal{O}(\sqrt{n})$	$\mathcal{O}(\log n)$
Brassard <i>et al.</i>	$\mathcal{O}(\sqrt[3]{n})$	$\mathcal{O}(\sqrt[3]{n})$

This leads to a significant reduction in the security level of current symmetric encryption algorithms. The impact of quantum computing on these algorithms is depicted in the table below.

Table 4 - Robustness comparison between classical algorithms

SECURITY BITS	SYMMETRIC-KEY	HASHING FUNCTION	INTEGER FACTORIZATION (i.e. RSA)	DISCRETE LOGARITHM (i.e. DH)	ELLIPTIC CURVE (I.E. X25519)
80	2TDEA		k=1024	L=1024, N=160	f=160-223 ³
112	3TDEA		k=2048	L=2048, N=224	f=224-255 ⁴
128 (KEY-SEARCH)	AES -128		k=3072	L=3072, N=256	f=256-383
128 (COLLISIONS)		SHA3-256			
192 (KEY SEARCH)	AES -192		k=7680	L=7680, N=384	f=384-511
192 (COLLISIONS)		SHA3-384			
256	AES -256		k=15360	L=15360, N=511	f=512+

Table 5 - Robustness scaling using quantum computer

SECURITY BITS	SYMMETRIC-KEY	HASHING FUNCTION	QUBIT SECURITY
128 (KEY SEARCH)	AES-128		64 $\mathcal{O}(\sqrt{2^{128}}) = \mathcal{O}(2^{64})$
128 (COLLISIONS)		SHA3-256	85 $\mathcal{O}(\sqrt[3]{2^{256}}) \approx \mathcal{O}(2^{85})$
192 (KEY SEARCH)	AES-192		96 $\mathcal{O}(\sqrt{2^{192}}) = \mathcal{O}(2^{96})$
192 (COLLISIONS)		SHA3-384	128 $\mathcal{O}(\sqrt[3]{2^{384}}) = \mathcal{O}(2^{128})$
256	AES-256		128 $\mathcal{O}(\sqrt{2^{256}}) = \mathcal{O}(2^{128})$

2.2. Risk mitigation

To protect *communications*, such as those on the Internet that use 'secure' connections via the HTTPS – TLS protocol, it is possible to maintain the same level of security by doubling the length of the key used, thereby increasing the robustness of the symmetric encryption algorithm. However, it should be noted that the preparatory phase of exchanging the aforementioned encryption keys remains

² See Definitions section.

³ Unsecure.

⁴ Security not guaranteed in the long term.

vulnerable (see Figure 2), whether using typical asymmetric encryption such as RSA or the *Diffie-Hellman*⁵, which is a *key exchange* algorithm that still employs asymmetric encryption.

Therefore, post-quantitative algorithms must also be considered in this context.

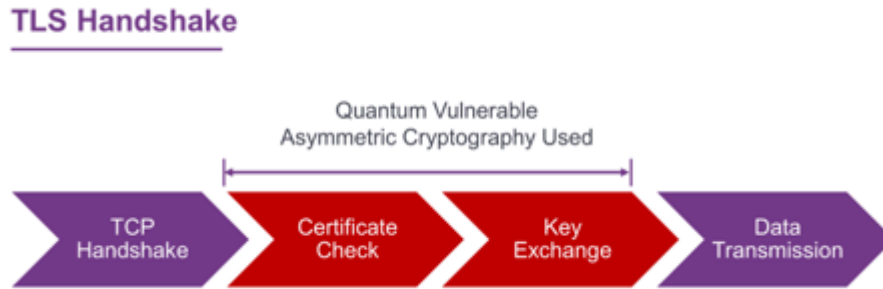


Figure 2 - TSL protocol and its vulnerabilities

Several mathematical techniques have been proposed to build quantum-secure cryptosystems, including:

- *hash* functions and zero-knowledge tests;
- error-corrected codes;
- *lattice*-based cryptography (including Learning with Errors – LWE – and NTRU)⁶;
- multivariate equations;
- isogenies of supersingular elliptic curves.

Indeed, the *new algorithms*, standardized by NIST, relate to *lattice* and *hash* function problems.

The aim of this paper is to examine and evaluate some of these new algorithms, specifically designed to withstand quantum computers. In particular, the key signing and encapsulation algorithms chosen for study are all lattice-based algorithms:

- *Dilithium* signature scheme (in versions 3 and 5);
- *FALCON* signature scheme (in versions 512 and 1024);
- *KYBER* key encapsulation mechanism (in versions 768 and 1024);
- *SPHINCS+* signature scheme, based on *hash* functions (will only be mentioned but not covered in detail in this document).

On *August 13, 2024*, NIST standardized the first three post-Quantum algorithms [7]:

- *FIPS 203, Module-Lattice-Based Key-Encapsulation Mechanism Standard: ML-KEM* is the name given to the standardized algorithm derived from *CRYSTALS-KYBER* [8];
- *FIPS 204, Module-Lattice-Based Digital Signature Standard: ML-DSA* is a digital signature scheme derived from the *CRYSTALS-DILITHIUM* algorithm, with its three main functions: *ML-DSA.KeyGen*, *ML-DSA.Sign* (Algorithm 2) and *ML-DSA.Verify* [9];
- *FIPS 205, Stateless Hash-Based Digital Signature Standard: SLH-DSA* based on version 3.1 of the *SPHINCS+* specification [10];
- *FIPS 206*, the future schema *FN-DSA* derived from *FALCON*.

⁵ It should be noted that the TLS protocol covers a multitude of different cipher *suites*, which define the algorithms used for key exchange and subsequent data encryption.

⁶ See Definitions section.

The PoC described in this paper was performed before the standardization of algorithms by NIST, so we will also use the Kyber, Dilithium and Sphincs+ nomenclature.

2.3. Algorithms, size and efficiency

The implementation of quantum-safe cryptographic algorithms is essential to ensure the security of digital communications in the face of emerging quantum technologies. This section provides an overview of the key algorithms considered in this study, focusing on their size and efficiency.

In the following sections, we will briefly present the winning algorithms of NIST *competition* and verify their *performance* with laboratory tests, as anticipated by Vidaković/Miličević [11, 12]. We will compare the various cryptographic functions (*Key pair generation, signature, verify, etc.*) of the various *quantum-safe* algorithms, also comparing them with those currently in common use in digital signature or encryption scenarios.

2.3.1. Dilithium

The Dilithium signature scheme is a lattice-based cryptographic algorithm (based on the Ring-LWE, *Ring Learning with Errors* problem) designed to provide robust security against quantum attacks. As described in [13], Dilithium gives an option of producing either deterministic or randomized signatures. In the deterministic approach, the randomness used in the signing process is derived from a deterministic function (e.g., a hash) of the message and the secret key. In contrast, the randomized version uses a random seed making each signature unique even for the same message and secret key. The deterministic version is simpler to implement and faster but is more susceptible to attacks that exploit the deterministic nature of signature generation, such as fault attacks (especially in hardware implementations). The randomized version offers better resistance against fault attacks and other attacks that leverage the deterministic nature of the deterministic version, but it can be more complex to implement.

Leaving aside the two implementation versions just described, the Dilithium algorithm is available in three different security versions, Dilithium2, Dilithium3 and Dilithium5, each offering varying levels of security and performance. The algorithm's efficiency is measured in terms of key generation, signature creation, and verification times, as well as the size of the keys and signatures produced (see Figure 3).

This pattern was used by NIST to standardize ML-DSA, a digital signature scheme consisting of three algorithms: ML-DSA.KeyGen, ML-DSA.Sign and ML-DSA.Verify.

Table 6 - Key and Signature Dimensions – Dilithium

	Dilithium 2	Dilithium 3	Dilithium 5
NIST Security Level	2	3	5
Public Key Size (bytes)	1312	1952	2592
Private Key Size (bytes)	2528	4000	4864
Signature Size (bytes)	2420	3293	4595

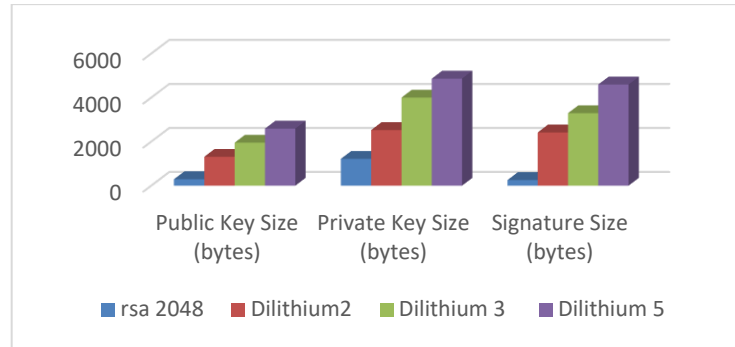


Figure 3 - Key and Signature Size Chart – Dilithium

The main advantages of this algorithm are:

- fast and not *memory-hungry* algorithm, compared to *hash-based* schemes such as Sphincs+;
- simple implementation; does not require complex calculations such as Gaussian sampling;
- it's easy to detect implementation errors.

On the other hand, the size of the signature and public key is approximately 2.3 times larger (*privateKey* + *signature*) than the Falcon algorithm.

2.3.2. Falcon

Falcon ('*FA*st-*FO*urier *LA*tterice-based *CO*mpact signatures over *NTRU*') is another lattice-based signature scheme known for its compact key and signature sizes (see Figure 4). It is available in two versions, Falcon 512 and Falcon 1024, which differ in their security levels and computational requirements. Falcon's efficiency is evaluated based on the same criteria as Dilithium, with a particular emphasis on its suitability for resource-constrained environments.

Table 7 - Key size and signature - Falcon

	FALCON-512	FALCON-1024
NIST Security Level	1	5
Public Key Size (bytes)	897	1793
Private Key Size (bytes)	1281	2305
Signature Size (bytes)	666	1280
Maximum Signature Size ⁷ (bytes)	690	1330

⁷ FALCON may have variable signature sizes.

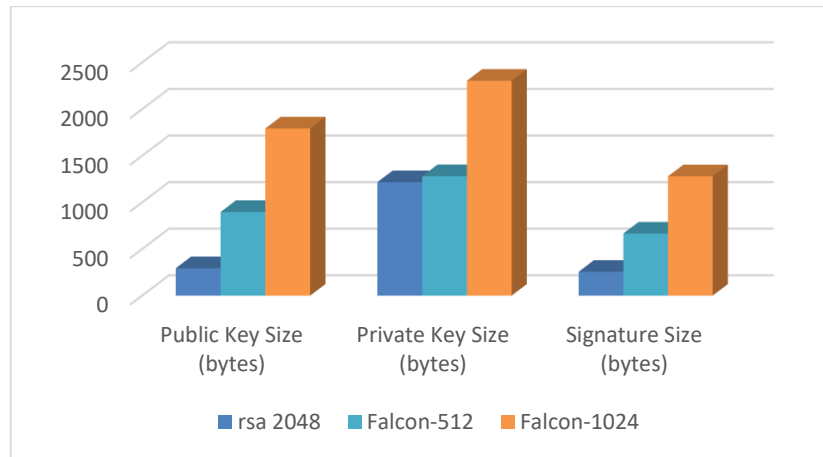


Figure 4 - Key and Signature Size Chart - Falcon

The main advantages of this algorithm are:

- more efficient in terms of bandwidth;
- fast in signature verification;

On the other hand, the disadvantages are:

- difficult to detect implementation errors;
- complex key creation and signing process (*floating point* arithmetic);
- not secure against *side-channel* attacks (masking).

2.3.3. Sphincs+

Let us now introduce another algorithm that emerged from the NIST *competition*: Sphincs+. This is a digital signature algorithm designed to withstand attacks from quantum computers based on a different technology than Dilithium and Falcon: *hash* and not *lattice*; as Vidaković/Miličević elaborated in detail [11, 14], it turns out to be the most computationally demanding signature algorithm: the signatures are large (although the keys are the smallest). Although it is less efficient, compared to others that similarly implement digital signature schemes, it has nevertheless been standardized by NIST. The main reason is the mitigation of the risks associated with the use of *lattice* technology for signature schemes: should this technology be compromised, there would be no alternative; in addition, of course, to the low space requirements for key storage, which is advantageous in specific contexts.

This algorithm was used by NIST to standardize the *Stateless Hash-Based Digital Signature Standard* 'SLH-DSA', based on version 3.1 of the SPHINCS+ specification.

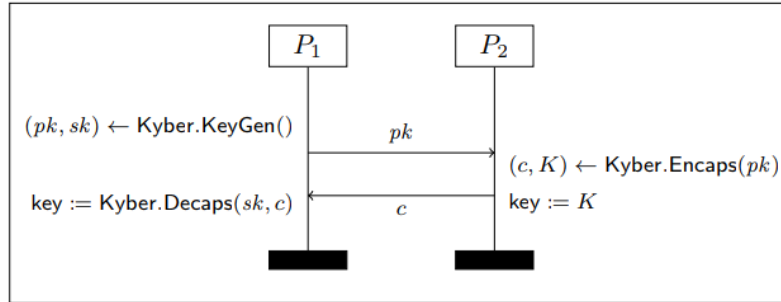
2.3.4. Kyber

Kyber is a lattice-based key encapsulation mechanism (KEM)⁸ designed to provide secure key exchange in a quantum-safe manner. This scheme (see Figure 5) is used to securely establish a shared secret between two parties that cannot be deciphered by a malicious attacker. It is an asymmetric encryption mechanism based on the learning-with-error (LWE) problem in lattice theory, which is assumed to be NP-hard. It is available in three versions, Kyber 512, Kyber 768 and Kyber 1024, each offering various levels of security and performance. Kyber's efficiency is assessed based on key

⁸ See Definitions section for more details - KEM (*key encapsulation mechanism*).

generation, encapsulation, and decapsulation times, as well as the size of the keys and ciphertexts produced.

This algorithm has been used by NIST to standardize Module-Lattice-Based Key-Encapsulation Mechanism Standard - ML-KEM - derived from CRYSTALS-KYBER specifications.



KEYGEN requires no input and generates a private key and a public key;

ENCAPSULATE takes a public key as input and produces a cipher text and a shared secret as output;

DECAPSULATE takes a cipher text and a private key as input and produces a shared secret.

Figure 5 - Overview of the KEM protocol - Source: [15]

Compared to the traditional *Diffie-Hellman* key exchange mechanism, in its version on elliptic curves (which is not resistant to quantum attacks), Kyber is about 2-3 times slower and requires a data *overhead* of about 70 times [16].

Table 8 - Key size and cipher text – Kyber

	KYBER512	KYBER768	KYBER1024
NIST Security Level	2	3	5
Public Key Size (bytes)	800	1184	1568
Private Key Size (bytes)	1632	2400	3168
Ciphertext Size	768	1088	1568
Shared Secret Size	32	32	32

Kyber-512 aims for security roughly equivalent to AES-128, Kyber-768 equivalent to AES-192 and Kyber-1024 equivalent to AES-256 [16].

3. Public Key Infrastructures and Payment Systems

Public Key Infrastructures (PKIs) play a crucial role in ensuring the security and integrity of payment systems, such as T2, T2 Securities and TIPS. They provide a framework for managing digital certificates and public-key encryption, which are essential for secure communication and authentication in financial transactions. This section delves into the application of PKIs in payment systems, highlighting their significance and the challenges they face in the context of quantum computing.

In this section we will focus on the Eurosystem Single Market Infrastructure Gateway (ESMIG) and describe how PKI services are implemented [17].

3.1. Description of certificates usage in ESMIG

The Eurosystem Single Market Infrastructure Gateway (ESMIG) is a critical component of the European financial infrastructure, facilitating secure and efficient communication between market participants and the Eurosystem.

The ESMIG infrastructure provides a set of features shared among all the TARGET Services, common components and applications beyond representing a single point of contact with the external networks [18]. These features belong to two main areas:

- Security, for example authentication of the sender and authorisation against a Closed Group of Users.
- Message management, for example message technical validation and forwarding.

The ESMIG provides business continuity measures (e.g. multiple sites, path diversification, etc.) and PKI services.

Digital Certificates, issued by PKI services, are used both by individuals, interacting with ESMIG in U2A mode, and applications, interacting with ESMIG in A2A mode.

The authentication of the "technical sender" (that is, the actor submitting an A2A or U2A request to TARGET services) is performed at network infrastructure level and is based on the certificate used by the actor to establish the technical connection with the network infrastructure itself. In case of successful authentication of the technical sender, the TARGET Services, common components or applications get the certificate Distinguished Name (DN) of the technical sender.

The A2A interaction is achieved through two different protocols: Data Exchange Protocol (DEP), used by the TARGET Services (excluding TIPS), and the Message Exchange Processing for TIPS (MEPT).

In order to guarantee the non-repudiation of emission, A2A messages are signed (see Figure 6) by the Network Service Provider (NSP)'s gateways using private keys stored in a HSM and signed by the CA. The NSP's network gateways of the receiver must check the validity of the certificate and verify the signature, using the public key certificate of the sender.

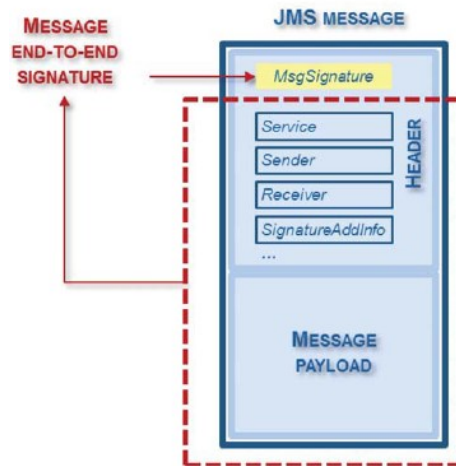


Figure 6 - Message signature

In the ESMIG environment, several types of certificates are used:

- A2A certificates used for digital signature, with the non-repudiation bit set in the 'Key usage' extension.
- U2A certificates used for digital signature and authentication, with the Non-Repudiation and the Digital Signature bit set in the 'Key usage' extension.
- TLS certificates for traffic data flow protection.

In case of compromised certificate, the NSP ensures its immediate revocation. Revocation information is published in the Certificate Revocation List (CRL) and made available in the OCSP services.

As described above, digital certificates issued by the PKI are used to enhance the security of payment systems in several ways:

- ensure *message integrity* (digital signature and non-repudiation);
- perform *authentication* of users (U2A) and applications (A2A);
- ensure *confidentiality* of data across networks (TLS protocol).

In our PoC we will explore all three kinds of certificate usage: digital signature, authentication and cryptography.

3.2. Certificates automation with ACME protocol in T2

In the TARGET2 (T2) payment system, the Automated Certificate Management Environment (ACME - RFC 8555) protocol is employed to streamline the issuance and renewal of digital certificates. ACME automates the process of obtaining and managing certificates, reducing the administrative burden on system operators, and enhancing the overall security of the T2 infrastructure. By leveraging ACME, T2 can ensure that certificates are always up-to-date and compliant with the latest security standard. This Protocol brings the following improvements:

Efficiency: Automated management implies a reduction in wasted time and resources attached.

Improved Security: Automation improves security, ensuring well-issued and managed certificates, with the opportunity to introduce stronger algorithms and big key lengths in a simple way.

Fewer Errors: Human errors will be avoided when automation will be integrated into place. For example, there are no chances for several human errors, even by a man attacker, during the manual issuance of certificates.

Compliance: Automation can become a way to systematically issue and manage certificates in accordance with the latest standards, regulations, and best practices.

Scalability: New server or application certificates may easily be issued and rolled out in parallel with the organisation growing in size.

Cost Reduction: Reduced manual intervention will ensure less cost and less time spent on certificate management.

3.3. PKI Security Threats

Quantum weakness is not the only security threat with PKIs. Indeed, one major issue of PKIs is that most of them are built on a centralised model (Certification Authority, CA), which is a single point of failure. The revocation of keys (certificates) relies on a centralised list (Certificate Revocation Lists, CRL). Certificates are issued by CAs which are assumed to be fully trusted organisations in the PKI system. CAs are expected to operate according to some rules which are announced as Certificate Policy (CP) and Certificate Practice Statement (CPS) documents [19]. Certificate revocation and validation processes can have serious security issues. Certificate owners must rely on CAs which have the full responsibility to revoke the certificates and give accurate revocation services and furthermore, many clients, including web browsers, rarely check whether the certificates are revoked or not [20].

3.3.1. Blockchain based PKIs

The blockchain technology is being widely adopted in trade and finance systems [21]. Blockchain was originally designed to support the implementation of Bitcoin, but now this technology has been successfully applied to many sectors; in the financial area, blockchain technology has great potential due to its decentralisation, safety, and traceability nature.

Moreover, Blockchain technology has recently been proposed by many authors for decentralised key management in the context of PKIs [22]. Instead of relying on trusted centralised validation authorities, the confirmation and revocation of keys is distributed over a multitude of participants. This paradigm may successfully address the above-described security threats.

However, blockchains are affected by other security issues; managing the replacement of cryptographic primitives through *crypto-agility*⁹ may not be very effective for payment systems [23]. In Bitcoin, for example, replacing the ECDSA digital signature scheme with a post-quantum scheme would almost certainly involve making hard forks¹⁰ in the transaction chain [24].

In this study, we will focus only on traditional PKIs, as blockchain ones should still be considered as experimental.

⁹ *Crypto-Agility* is the ability of a security system to switch from a cryptography method to another and is focused on visibility and the dynamic movement of cryptography resources of an organization. A *crypto-agile* company is able to substitute obsolete cryptography resources without significant interruptions to the infrastructure.

¹⁰ A *hard fork*, as it relates to blockchain technology, is a radical change to a network's protocol that makes previously invalid blocks and transactions valid, or vice-versa. A hard fork requires all nodes or users to upgrade to the latest version of the protocol software.

4. The Proof-of-Concept conducted in the Bank of Italy

The Bank of Italy is a *Qualified Trusted Service Provider* (QTSP) under the eIDAS Regulation for the issuance of qualified electronic signature certificates. It is registered in the list of Qualified Service Providers in Italy, maintained by AgID¹¹ and valid at European level. The certificates are issued to employees of the Bank or to representatives of institutional stakeholders and are used exclusively in relations with the Bank. The Bank also issues 'auxiliary' certificates (authentication and encryption, TLS, etc.). The Bank's PKI infrastructure is entirely *on-premises*.

In the test environment of the PKI infrastructure, a *Proof-of-Concept* (PoC) was realised consisting of various *Certification Authorities* instances hosted on a virtual machine equipped with the software packages shown in the following table.

Table 9 - Software components used in the PoC

Component	Software
CERTIFICATION AUTHORITY	EJBCA Community Edition, v. 8
VIRTUAL MACHINE	Operating System: Red Hat Enterprise Linux v. 8.8 Amount of RAM: 4 GB
JAVA VIRTUAL MACHINE	OpenJDK Runtime Environment (Red_Hat-11.0.20.0.8-1, build 11.0.20+8-LTS)
APPLICATION SERVER	jboss-eap-7.4
COMPILER TOOL	apache-ant-1.10.7
DBMS	mysql v. 8.0.32
CRYPTOGRAPHIC API	Bouncy Castle library v1.73

The *Certification Authority* software was customised appropriately in order to also support hybrid certificates, as detailed in the next section.

4.1. Hybrid Certificates

The term *hybrid* refers to various methods of combining classical and post-quantum (PQ) cryptographic algorithms, particularly in the context of digital certificates. The main hybridization strategies can be classified into three categories: **composite**, **multiple** (or *chameleon*) and “real” **hybrid**.

4.1.1. Composite approach

In this scheme, two public keys (one classical, such as RSA, and one post-quantum) belonging to the subscriber are combined and included in a single certificate. This ensures that as long as at least one of the cryptographic algorithms remains secure (i.e., unbroken), the overall security is preserved. However, this approach requires that the software processing the certificates be capable of understanding and handling the new post-quantum algorithms, making it unsuitable for transitional environments or legacy systems.

¹¹ The Agenzia per l'Italia Digitale - Agency for Digital Italy (AgID) - is the technical agency of the Presidency of the Council of Ministers that guarantees the achievement of the objectives of the Italian digital agenda, coordinating all Italian public administrations.

4.1.2. Multiple (or *Chameleon*) approach

This approach uses two distinct but closely linked certificates, one based on a classical algorithm (e.g., RSA) and the other on a PQ algorithm. In a typical TLS connection, for instance, two separate authentication phases would occur: one using the legacy certificate and one using the post-quantum certificate. Although conceptually elegant and flexible, this solution introduces additional operational complexity and may impact performance and connection management.

4.1.3. Hybrid approach

In the PoC, so-called '**real hybrid**' certificates were used, i.e. certificates with two public keys (one traditional and one *quantum-safe*) appropriately defined within them (see Figure 7). There are currently no standards for this type of certificate, only draft implementations. The IETF (*Internet Engineering Task Force*) versions were evaluated in the study [25] and the Technical University of Darmstadt (CROSSINGTUD) [26] which exploits the *hybrid using extension* approach.

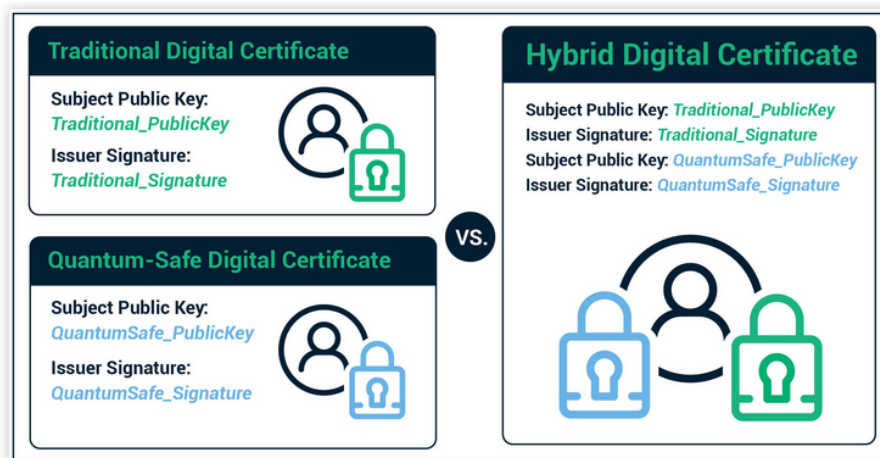


Figure 7 - Representation and comparison of different types of certificates (Source: Sectigo)

➤ IETF coding

In the IETF implementation [25] some non-critical attributes were added in order to define the secondary key type (*OID-2.5.29.72*), the secondary signing algorithm (***OID-2.5.29.73***) and the secondary public key signed by the CA (*OID-2.5.29.74*).

In the example (see Figure 8), the representation of a certificate in ASN.1 encoding is shown, in which the three attributes with reference to the Dilithium5 scheme is highlighted.

```

SEQUENCE (1 elem)
  [0] (20 byte) C6A403855FBD360CA0A211D62DF4C90691F7510F
Extension SEQUENCE (2 elem)
  extnID OBJECT IDENTIFIER 2.5.29.37 extKeyUsage (X.509 extension)
  extnValue OCTET STRING (22 byte) 301406082B0601050507030206082B06010505070304
    SEQUENCE (2 elem)
      OBJECT IDENTIFIER 1.3.6.1.5.5.7.3.2 clientAuth (PKIX key purpose)
      OBJECT IDENTIFIER 1.3.6.1.5.5.7.3.4 emailProtection (PKIX key purpose)
Extension SEQUENCE (2 elem)
  extnID OBJECT IDENTIFIER 2.5.29.14 subjectKeyIdentifier (X.509 extension)
  extnValue OCTET STRING (22 byte) 0414D9532D0FFA2B7F774162B0E44414B9053793B39A
    OCTET STRING (20 byte) D9532D0FFA2B7F774162B0E44414B9053793B39A
Extension SEQUENCE (3 elem)
  extnID OBJECT IDENTIFIER 2.5.29.15 keyUsage (X.509 extension)
  critical BOOLEAN true
  extnValue OCTET STRING (4 byte) 030205E0
    BIT STRING (3 bit) 111
Extension SEQUENCE (2 elem)
  extnID OBJECT IDENTIFIER 2.5.29.72
  extnValue OCTET STRING (1976 byte) 308207B4300D060B2B0601040102820B070605038207A10001F9AD925D7FCF2385DF...
    SEQUENCE (2 elem)
      SEQUENCE (1 elem)
        OBJECT IDENTIFIER 1.3.6.1.4.1.2.267.7.6.5
        BIT STRING (15616 bit) 0000000111111001101011011001001001011101011111111001111001000111000...
      Extension SEQUENCE (2 elem)
        extnID OBJECT IDENTIFIER 2.5.29.73
        extnValue OCTET STRING (15 byte) 300D060B2B0601040102820B070807
          SEQUENCE (1 elem)
            OBJECT IDENTIFIER 1.3.6.1.4.1.2.267.7.8.7
          Extension SEQUENCE (2 elem)
            extnID OBJECT IDENTIFIER 2.5.29.74
            extnValue OCTET STRING (4600 byte) 038211F4007E00890FA8831AB9E1B8261154469A057BE0C6F43CC5E333D8DE0A9F2B...
              BIT STRING (36760 bit) 01111110000000001000100100001111101010001000001100011010101110011110...
signatureAlgorithm AlgorithmIdentifier SEQUENCE (2 elem)
  algorithm OBJECT IDENTIFIER 1.2.840.113549.1.1.11 sha256WithRSAEncryption (PKCS #1)
  parameters ANY NULL
signature BIT STRING (4096 bit) 0100101010100110110110101010001100111101011001001001110011000100001...

```

Figure 8 - ASN.1 representation of the certificate with evidence on PQ attributes - IETF implementation

➤ CROSSINGTUD coding

In the CROSSINGTUD implementation [26] the non-critical attributes defining the algorithm and secondary key (*OID-2.5.29.211*) and the secondary public key signed by the CA (*OID-2.5.29.212*) were added.

In the example (see Figure 9), a certificate displayed in its ASN.1 encoding shows the 2 attributed with reference to the Dilithium3 for the secondary public key and Dilithium5 scheme for its signature by the CA.

```

SEQUENCE (1 elem)
  [0] (20 byte) C6A403855FBD360CA0A211D62DF4C90691F7510F
Extension SEQUENCE (2 elem)
  extnID OBJECT IDENTIFIER 2.5.29.37 extKeyUsage (X.509 extension)
  extnValue OCTET STRING (22 byte) 301406082B0601050507030206082B06010505070304
    SEQUENCE (2 elem)
      OBJECT IDENTIFIER 1.3.6.1.5.5.7.3.2 clientAuth (PKIX key purpose)
      OBJECT IDENTIFIER 1.3.6.1.5.5.7.3.4 emailProtection (PKIX key purpose)
Extension SEQUENCE (2 elem)
  extnID OBJECT IDENTIFIER 2.5.29.14 subjectKeyIdentifier (X.509 extension)
  extnValue OCTET STRING (22 byte) 0414875601E683D328CF5805CA9CB63D19F37B883DA2
    OCTET STRING (20 byte) 875601E683D328CF5805CA9CB63D19F37B883DA2
Extension SEQUENCE (3 elem)
  extnID OBJECT IDENTIFIER 2.5.29.15 keyUsage (X.509 extension)
  critical BOOLEAN true
  extnValue OCTET STRING (4 byte) 030205E0
    BIT STRING (3 bit) 111
Extension SEQUENCE (2 elem)
  extnID OBJECT IDENTIFIER 2.5.29.211
  extnValue OCTET STRING (1976 byte) 308207B4300D06082B0601040102820B070605038207A1002DCEC562B3AF9D631C66...
    SEQUENCE (2 elem)
      SEQUENCE (1 elem)
        OBJECT IDENTIFIER 1.3.6.1.4.1.2.267.7.6.5
        BIT STRING (15616 bit) 0010110111001110110001010110001010110011101011110011101011000110001...
      Extension SEQUENCE (2 elem)
        extnID OBJECT IDENTIFIER 2.5.29.212
        extnValue OCTET STRING (4619 byte) 30821207300D06082B0601040102820B070807038211F400ACF26CB90E51129053A5...
          SEQUENCE (2 elem)
            SEQUENCE (1 elem)
              OBJECT IDENTIFIER 1.3.6.1.4.1.2.267.7.8.7
              BIT STRING (36760 bit) 10101100111100100110110010111001011000100011001010001000100100100000101...
signatureAlgorithm AlgorithmIdentifier SEQUENCE (2 elem)

```

Figure 9 - ASN.1 representation of the certificate with evidence on PQ attributes - Crossingtut implementation

4.2. Certification Authorities

In the Proof-of-Concept (PoC) conducted by the Bank of Italy, several *Certification Authority* (CAs) instances were created, including *quantum-ready* versions. Due to the unavailability of Hardware Secure Modules (HSMs) implementing post-quantum algorithms¹² on the European market at the time of the tests, software tokens were utilized for storing the root keys of the CAs and for signing the Certificate Revocation Lists (CRLs). The Online Certificate Status Protocol (OCSP) service was not included within the scope of the PoC, as the only impacted aspects would have been the signing times of the individual OCSP responses, a topic discussed in detail in the specific section comparing execution times.

As can be seen from the table below, the following CAs were created:

Table 10 - CA generated in the PoC

CA	Description
ROOTCA_RSA	<i>Certification Authority</i> with classic key algorithm <i>RSA-4096</i>
ROOTCA_HYBRID_RSA_DIL5	Hybrid <i>Certification Authority</i> , according to the crossingtut formalism, with primary key <i>RSA-4096</i> and secondary key <i>Dilithium5</i>
ROOTCA_HYBRID2_RSA_DIL5	Hybrid <i>Certification Authority</i> , according to IETF formalism, with primary key <i>RSA-4096</i> and secondary key <i>Dilithium5</i>
ROOTCA_DILITHIUM5	<i>Certification Authority</i> with <i>quantum-safe</i> key algorithm <i>Dilithium5</i> .

4.2.1. Description of the tools used in the PoC

In the execution of the Proof of Concept (PoC), a variety of sophisticated tools were employed; it was necessary to use third-party libraries and create ad hoc tools for the concrete use of quantum

¹² At the time the tests were carried out, some manufacturers had already produced devices with support for *post-quantum* algorithms (e.g. Thales Luna T-Series), but they were not yet available on the European market or did not yet have the required certifications for use in qualified production environments.

safe algorithms in the various phases of the certificate lifecycle and in typical usage scenarios. The use of Bouncy Castle¹³ libraries was mandatory, as they are the only ones that implement the specifications of the new post-quantum algorithms.

4.2.2. Types of Certificates produced

During the PoC, various types of certificates were issued according to possible combinations of algorithms and keys (hybrid or Quantum-safe for each CA); in addition, for simplicity, all certificates were configured with *Key Usage* extensions (Digital Signature, Non-repudiation, Key encipherment) and *Extended Key Usage* (Client Authentication, Email protection).

4.3. Testing the use of certificates



The evaluation of certificate usage was conducted through a series of rigorous tests designed to assess the functionality and security of the implemented cryptographic algorithms. These tests were essential in verifying the robustness of the quantum-safe Public Key Infrastructure (PKI) within the context of the Proof of Concept (PoC). In the following sections, we will describe the tests for using the certificates produced in commonly used scenarios, such as the encryption of the communication channel in an SSL connection, digital signature scenarios, both for *PAdES* and *CAdES*, and finally we will report the outcome of the compatibility tests of file encryption with the Kyber algorithm.

4.3.1. TLS connection test

The Transport Layer Security (TLS) connection test was performed to ensure the secure transmission of data between clients and servers. This test involved the establishment of TLS connections using certificates that incorporated quantum-safe cryptographic algorithms. The primary objective was to validate the compatibility and performance of these algorithms in real-world scenarios, ensuring that they could effectively replace classical cryptographic methods without compromising security or efficiency.


We assumed the use of certificates with a typical *web server*; for this purpose, one of the most popular was chosen: Apache Tomcat in version 8.5.94. It is worth emphasising that it was necessary to set all CA certificates created in the PoC as trusted *root* certificates in the operating system's *certificate store*.

Table 11 - Hybrid certificate with crossingtud coding and DILITHIUM5 secondary key issued by hybrid CA

Certificate	Condition	Result
Issued by CA_HYBRID_RSA_DIL5 with RSA 2048 primary public key and DILITHIUM5 secondary public key	SSL server keystore PKCS12 with only Dilithium5 private key	The SSL connection fails due to a server-side exception. 
	SSL server keystore PKCS12 with only RSA private key	The SSL connection is established correctly, only showing the <i>backward</i> compatibility in the use of the hybrid certificate. 

¹³ <https://www.bouncycastle.org/>

Table 12 - Dilithium5 PQ key certificate, issued by CA Dilithium5

Certificate	Condition	Result
Issued by CA_DILITHIUM5 with DILITHIUM5 primary public key	SSL server keystore PKCS12 with Dilithium5 private key	In this test, the <i>application server</i> failed to interpret the quantum key algorithm and thus raised an exception at start-up. [1.3.6.1.4.1.2.267.7.6.5 KeyFactory not available] 

4.3.2. Signature and verification test

The signature and verification test aimed to evaluate the integrity and authenticity of digital signatures generated using quantum-safe algorithms. This test involved the creation of digital signatures for various data sets, followed by the verification of these signatures to ensure their validity. To set up a signature scenario, we used the tools commonly available on Bank of Italy's workstations, such as *Acrobat Reader DC*, *Actalis FileProtector* and *DSS (Digital Signature Service)*¹⁴ for the verification phase.

Table 13 – Signature with Dilithium5 certificate, issued by CA Dilithium5


Certificate	Condition	Result
Issued by CA_DILITHIUM5 with DILITHIUM5 primary public key	Configuring digital ID in PKCS#12 format	The test ended even before reaching the signature application phase: when attempting to configure a software digital ID in PKCS#12 format, an exception was raised, showing that the algorithm used was not recognised as valid ("The Windows Cryptographic Service Provider reported an error: ASN1 bad tag value met. Error Code: 2148086027"). 

Table 14 – Signature with hybrid certificate with RSA and Dilithium5 keys, issued by CA DILITHIUM5



Certificate	Condition	Result
Issued by CA_HYBRID_RSA_DIL5 with RSA 2048 primary public key and DILITHIUM5 secondary public key	Configuring digital ID in PKCS#12 format	In this scenario, the configuration of the certificate on a PKCS#12 file as a digital ID occurs correctly. The signing process was successfully completed, demonstrating backward compatibility of hybrid certificates in this context. 

Table 15 – CADES Verification with Dilithium5 certificate, issued by CA DILITHIUM5

Certificate	Condition	Result
Issued by CA_DILITHIUM5 with Dilithium5 primary public key	Verifying File signed with PQ certificate with standalone tool	The verification application <i>File Protector</i> , while recognising the cryptographic envelope, was unable to interpret the <i>hashing</i> algorithm used. As described in [12], SHAKE version 128/256 ¹⁵ can be used as the <i>hashing</i> algorithm in the signature. Each version is a function of the level of security to be achieved. "ERROR: Hashing algorithm:2.16.840.1.101.3.4.2.18" 

¹⁴ DSS - Digital Signature Service .

¹⁵ SHAKE256 is defined by OID 2.16.840.1.101.3.4.2.18.





	Verifying File signed with PQ certificate with online tool	Using the DSS web application, we get the message “ <i>Unsupported algorithm: 1.3.6.1.4.1.2.267.7.8.7</i> ”. In this case, the exception raised is on the signing algorithm used, which turns out not to be supported.	
--	--	---	---

Table 16 – PAdES Verification with Dilithium5 certificate, issued by CA DILITHIUM5

Certificate	Condition	Result	
Issued by CA_DILITHIUM5 with Dilithium5 public key	Validation signed file with adobe tool	Using the tools implemented in the Proof of Concept (PoC) perimeter, we successfully generate PAdES signatures employing post-quantum algorithms. However, while the PDF file is recognised by Adobe software, the signature verification fails. The signature is deemed ' <i>invalid</i> ' and the signer's information is not accurately extracted from the certificate. Repeating the same exercise with <i>File Protector</i> , the result did not change.	

4.3.3. Cryptography tests











For this test, a certificate with a Kyber1024 key issued by a classic RSA-4096 CA was used.

Certificate	Condition	Result	
Issued by CA_RSA with Kyber1024 public key	A certificate with a Kyber1024 key issued by a classic RSA-4096 CA	To decrypt the encrypted file, we tried installing the PKCS#12 file on a smart card, but each time we got import errors.	
		Decryption with the <i>customised tool</i> , on the other hand, was successful.	

4.3.4. Summary of tests performed

The following table summarises the various tests carried out during the PoC.

Table 17 - summary of tests performed

	CA ISSUER	Certificate primary key	Certificate secondary key	Test scene	Signature format		Outcome
1	DILITHIUM5	Dilithium5		Signing with Adobe	PAdES		Digital ID configuration failed.
2	DILITHIUM5	RSA-2048	Dilithium5	Signing with Adobe	PAdES		Successful configuration and signing.
3	DILITHIUM5	Dilithium5		Signature verification with software verifier	CAdES		The verifier did not recognise the hashing algorithm used
				Signature verification with web software verifier	CAdES		The tool did not recognise the algorithm of Dilithium signature.
4	DILITHIUM5	Dilithium5		Signature verification with Adobe	PAdES		Adobe fails to verify the signature and defines it as invalid.
				Signature verification with software verifier	PAdES		The verifier failed to interpret the certificate
5	RSA4096	Kyber 1024		File encryption	p7e		PKCS#12 could not be imported on device.
6	HYBRID RSA DI L5	RSA-2048	Dilithium5	SSL connection with hybrid certificate			Using the secondary key in PKCS#12 the handshake of the SSL connection does not complete
							With the RSA key, hybrid certificate compatibility is maintained, and the connection is correctly established
7	DILITHIUM5	Dilithium5		SSL connection with PQ certificate			The application server did not recognize the key algorithm in PKCS#12 and crashed on start-up.

5. Performance Tests

The following section describes the evaluation of the various algorithms, both from a computational and memory occupation point of view, and where possible we will make comparisons with classical algorithms.

All evidence was calculated by evaluating the average times over a reasonable number of repetitions, using a virtual server with the following configuration:

- Operating System: CentOS 7.9.2009
- RAM: 7.64 GB
- CPU(s): 4
 - model name: Intel(R) Xeon(R) Gold 6238R CPU @ 2.20GHz
 - cpu family: 6
 - model: 85
 - cpu MHz: 2194.843
 - cache size: 39424 KB

5.1. Dimensions

In the tables below, it is evident that the size of the objects generated with the *Dilithium* algorithm, encompassing both key-pair and *signature* cryptographic objects, are 2-3 times larger compared to *Falcon*, as corroborated by [11, 14]. Classical algorithms demand significantly less memory resources than *quantum-safe* algorithms; however, this was to be expected as robustness to quantum computers is paid for in terms of space and/or computational time (see Figure 10).

Table 18 - Size of cryptographic objects across algorithms

	RSA 2048	Dilithium2	Dilithium3	Dilithium5	Falcon-512	Falcon-1024
Public Key Size (bytes)	294	1312	1952	2592	897	1793
Private Key Size (bytes)	1217	2528	4000	4864	1281	2305
Signature Size (bytes)	256	2420	3293	4595	666	1280

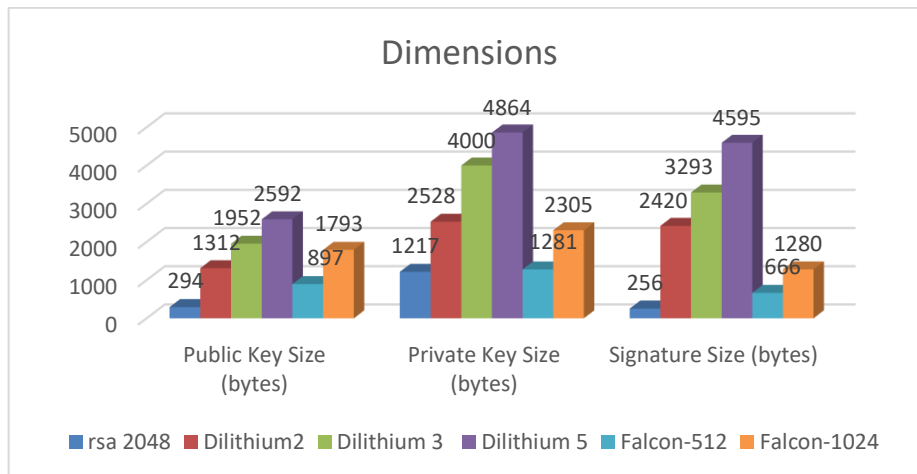


Figure 10 - Size of cryptographic objects across algorithms

The size of cryptographic objects grows as the level of security increases; moreover, for each type, there is at least an order of magnitude more than the current asymmetric encryption algorithm used today (RSA-2048). It is immediately apparent that *Falcon* is the most parsimonious of the *quantum safe* algorithms.

5.2. Execution time

We will now analyze the performance of the various classical and *quantum safe* algorithms, in the various scenarios of key creation, digital signature creation (*signature - CAdES*) and verification.

5.2.1. Key Generation

An initial analysis shows that the time required to define key pairs is much shorter with the new *quantum safe* algorithms, especially when compared to classical algorithms such as RSA with key lengths of 2048, 3072 or 4096 (see Figure 11).

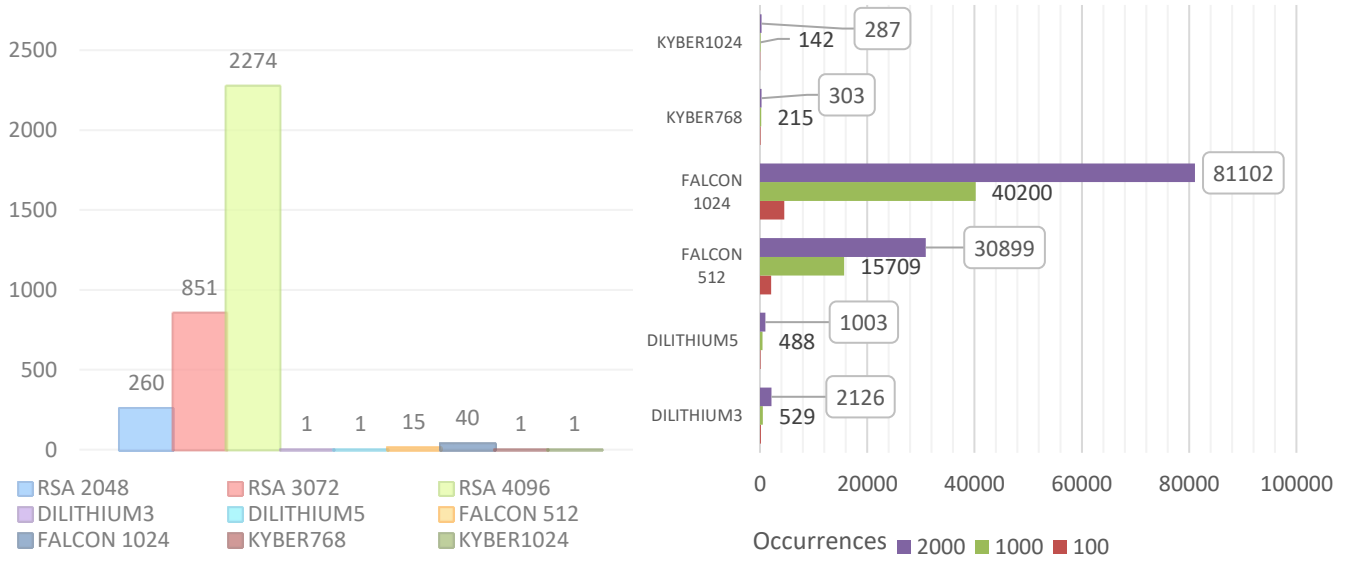


Figure 11 - Comparison of key creation times (ms)

Figure 12 - Sum of times (ms) as the number of occurrences increases

Let us now compare only the *post-quantum* algorithms. *Falcon* turns out to be the least efficient from the point of view of computing resources, and this evidence is more sensitive when we try to increase the number of occurrences (see Figure 12).

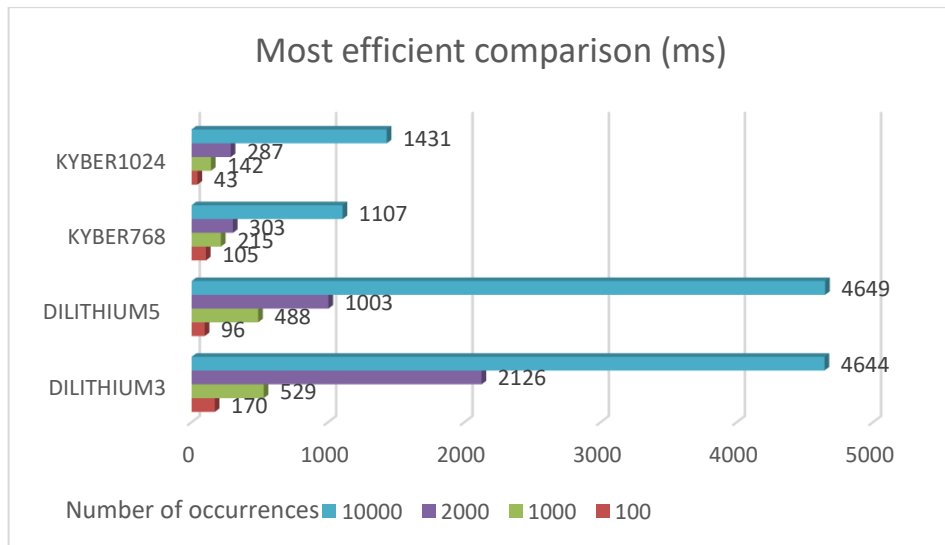


Figure 13 - Comparison among the most efficient

Evidence shows that key creation times are longer for classical algorithms. Among *quantum-safe* algorithms, *Falcon* proves to be very costly, while *Dilithium*, even in its most robust version,

remains within acceptable times. Even though Kyber is more efficient, it has different purposes (see Figure 13).

5.2.2. Digital signature creation and verification

In this scenario, we tested the various algorithms during the *CAdES* digital signature of a PDF test file of approximately *300KB* in size. Timeframes were normalized by not considering the phase of loading the key and reading the document to be signed. The evidence shows that *Dilithium* is the most efficient (even comparing algorithms with the same level of security, see Figure 14).

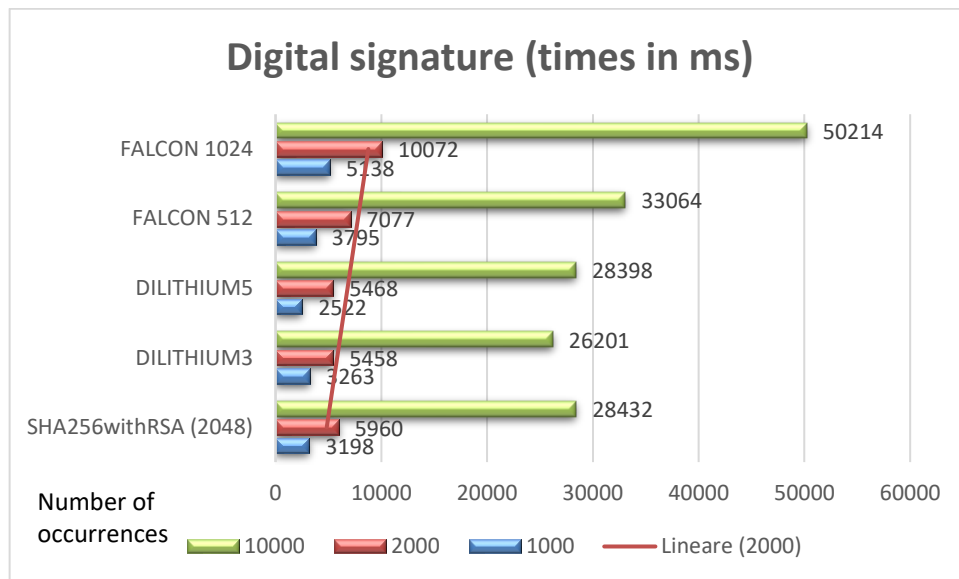


Figure 14 - Digital signature times (ms)

Please note that the *Dilithium signature* scheme is always a winner compared to its main competitor Falcon, even compared to the classic RSA scheme.

Would anything have changed if we had used a *20MB* file for the various signatures?

We reported the evidence below: even by increasing the size of the file to be signed, Dilithium confirms itself as the best performing, among PQ algorithms (see Figure 15).

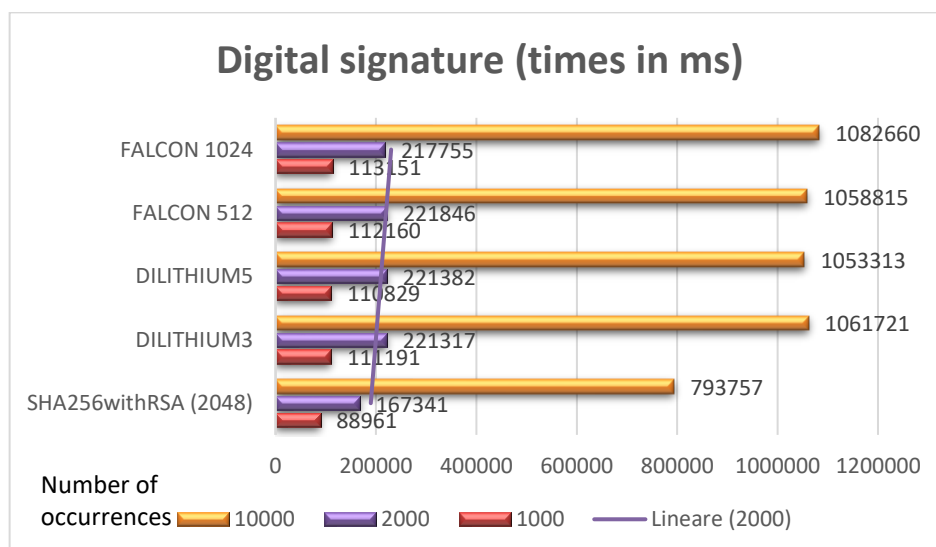


Figure 15 - Digital signature times (ms) for a 20 MB file

By comparing the times between the two scenarios, we obtain the graph below (see Figure 16).

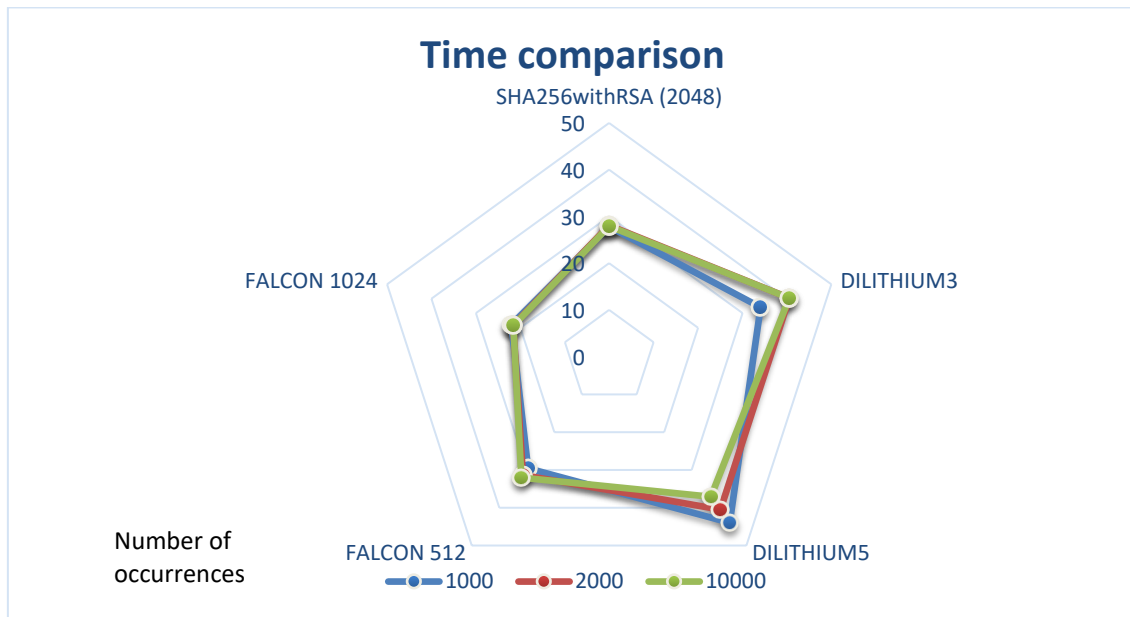


Figure 16 - Digital signature time comparison

It is worth highlighting, however, that Falcon *is less susceptible to this aspect*, although Dilithium recovers as the number of occurrences increases.

On the other hand, Falcon *performs* better in the verification phase, also due to the smaller *signature* size when compared to Dilithium. Its efficiency makes it most suitable in massive signature verification scenarios (see Figure 17).

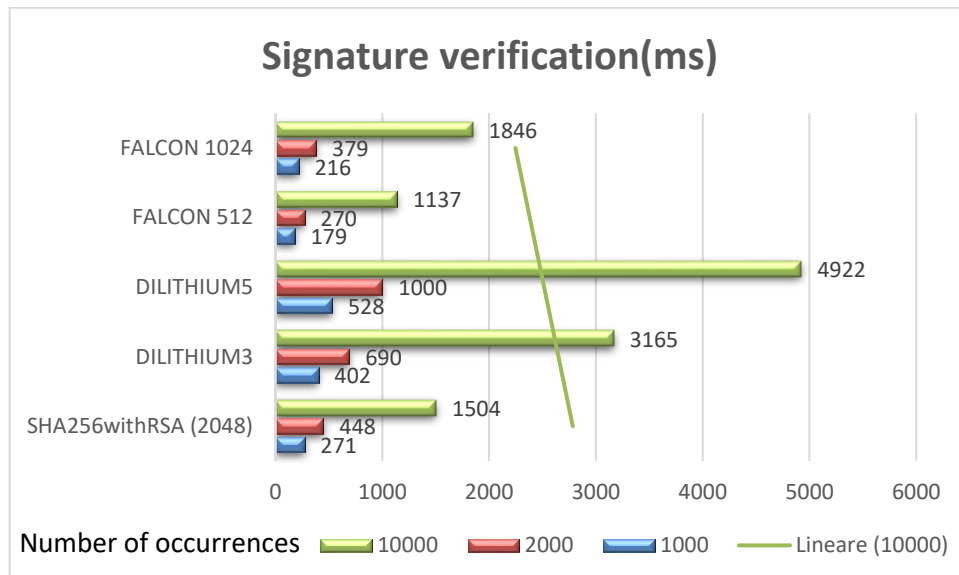


Figure 17 - Signature verification times (ms)

The verification phase is not significantly impacted by the size of the signed file: as the occurrences increase, the timing relationships stabilize (see Figure 18).

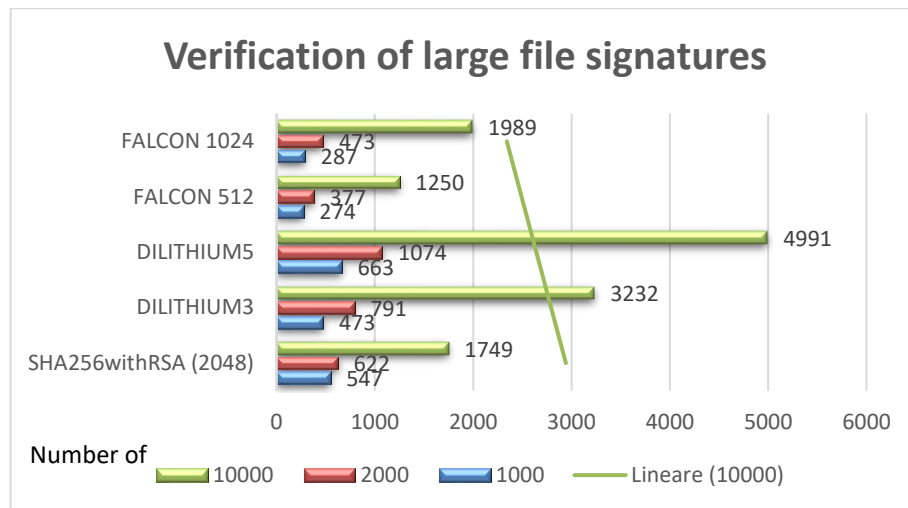


Figure 18 - Signature verification times for large files

Even in this verification scenario, the comparison of times shows us that the verification phase is not impacted by the size of the signed file (see Figure 19).

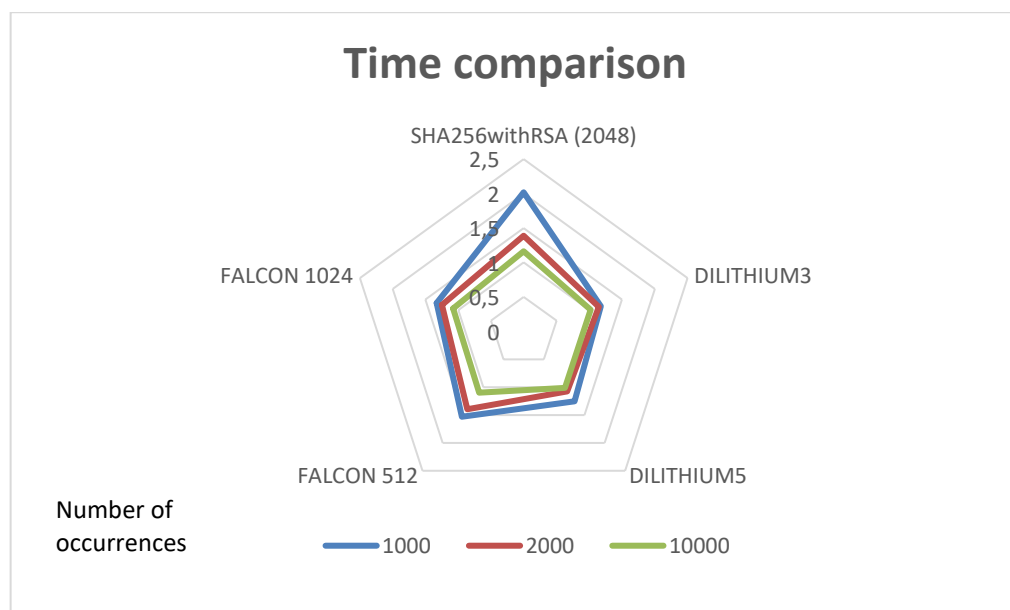


Figure 19 - Signature verification time comparison

This evidence is also reflected in the metrics published by the PQSHIELD portal¹⁶ (see Figure 20).

¹⁶ <https://pqshield.github.io/nist-sigs-zoo/#performance>.


Scheme	Parameterset	NIST level	Sign (cycles)	Verify (cycles)
Falcon	1024	5	2.053.080	160.596
Falcon	512	1	1.009.764	81.036
ML-DSA (Dilithium)	ML-DSA-87	5	642.192	279.936
ML-DSA (Dilithium)	ML-DSA-65	3	529.106	179.424
ML-DSA (Dilithium)	ML-DSA-44	2	333.013	118.412
RSA 	2048	Pre-Q	27.000.000	45.000

Figure 20 - Performance of signature algorithms¹⁷

5.2.3. Generation of a Certificate Signing Request (CSR)

Empirically, one might anticipate that the generation times for Certificate Signing Requests (CSRs) would be shorter for classical algorithms due to the smaller key and signature sizes. However, experimental data indicate the contrary, as illustrated in the accompanying table. Actually, considering PQ algorithms alone, *Dilithium* proves to be more effective than *Falcon* even in this scenario (see Figure 21).

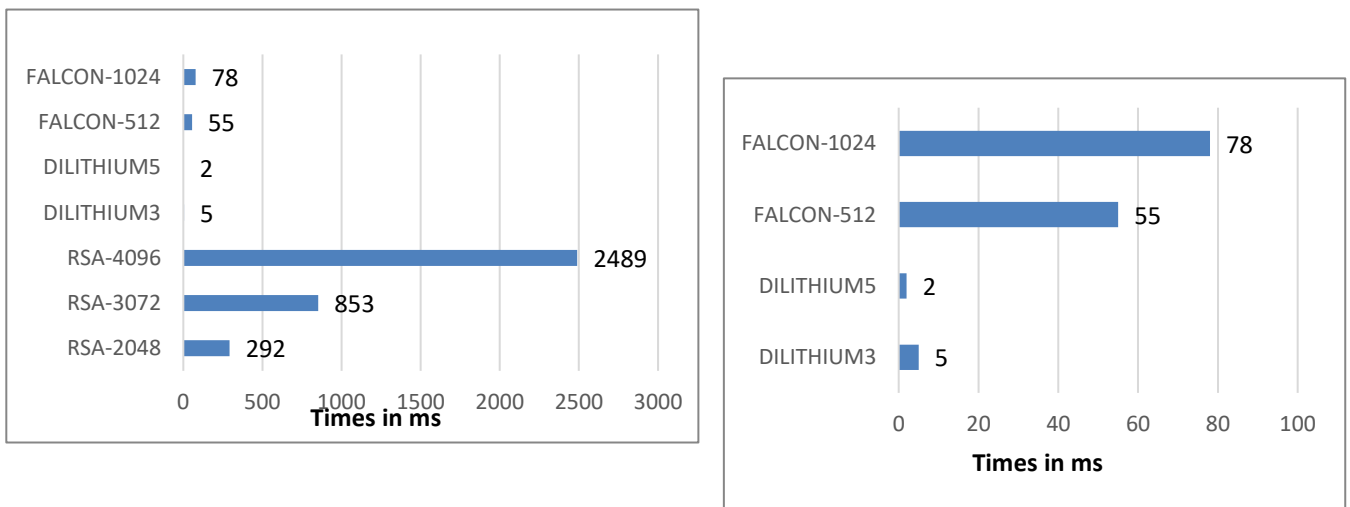


Figure 21 - Comparison of execution times in the creation of a CSR

¹⁷ it is clarified that the 'signing' phase also includes the part of generating the relevant key pair.

6. Conclusions

Dilithium and *Falcon* are both *lattice-based* digital signature algorithms designed to be resistant to quantum attacks. In contrast, *SPHINCS+* employs a fundamentally different approach, relying on *hash-based* cryptographic constructions. These algorithms are leading candidates in the NIST Post-Quantum Cryptography (PQC) standardization process, with differing performance characteristics that make them suitable for distinct application contexts.

Performance evaluations indicate that *Dilithium* consistently demonstrates superior computational efficiency, particularly in key generation and signature creation, making it well-suited for environments with limited processing capabilities. *Falcon*, on the other hand, stands out for its fast signature verification, although it requires more computational resources for key generation and signing.

In terms of key and signature size, *Falcon* benefits from a compact design, offering a smaller public key and signature size compared to other post-quantum algorithms. This makes it advantageous in constrained environments such as embedded systems or networks with limited bandwidth. While *Dilithium* is generally efficient, its relatively larger signature size may introduce performance bottlenecks in bandwidth-restricted scenarios.

The comparative analysis revealed the following insights:

- *Security and Size Efficiency*: *Falcon* offers the smallest public key size among the analyzed schemes, which is particularly beneficial for memory-constrained devices.
- *Computational Performance*: *Dilithium* is efficient in key generation and signing, while *Falcon* excels in signature verification speed.
- *Applicability to Constrained Devices*: in systems such as smart cards, which are characterized by limited memory, slow processors, and low communication bandwidth (typically <100 kB/s), cryptographic performance can often be improved by hardware acceleration. However, dedicated hardware support for *quantum-safe* algorithms is not yet widely available. As hardware implementation and certification processes are lengthy (typically 6–18 months), current efforts rely heavily on software-based implementations.

Moreover, such platforms require *side-channel* resistance, often achieved via masking techniques. These countermeasures, while necessary for secure deployment, result in increased execution time and memory consumption. For instance, the execution time of *Dilithium* increases by a factor of approximately 5.6 when such protections are enabled.

The evaluation also confirmed the limited compatibility of existing software and infrastructure with post-quantum algorithms. Most tools and servers not yet updated to support *quantum-safe* standards failed to complete signing or encryption operations, underscoring the need for broader support and integration across ecosystems.

In conclusion, both *Dilithium* and *Falcon* demonstrate suitability for deployment on constrained devices such as smart cards. However, *Dilithium* offers the best trade-off between memory usage and computational efficiency across the various scenarios evaluated, making it a more balanced choice for general-purpose use in post-quantum public key infrastructures.

6.1. Crypto-agility of payment systems

Enhancing the cryptographic resilience of payment systems in preparation for the post-quantum era requires several parallel interventions. Key areas include:

- the deployment of Quantum Key Distribution (QKD) systems for secure key exchange;
- the adoption of true random number generators with quantum entropy sources;
- the implementation of *crypto-agile* architectures, capable of adapting rapidly to evolving cryptographic standards.

This study highlights the critical dependence of payment infrastructures on cryptographic services, many of which are provided by third-party entities, such as Public Key Infrastructures (PKIs). Core components of payment systems—such as access gateways (e.g., ESMIG) and digital transaction platforms—may rely on certificates issued by external Network Service Providers (NSPs).

To mitigate risks stemming from third-party cryptographic weaknesses in the face of quantum threats, it is essential to include explicit security requirements in technical specifications and procurement contracts. At a minimum, such contracts should mandate that service providers implement a roadmap toward crypto-agility, ensuring a timely transition to quantum-safe algorithms as they are standardized and become commercially available for use in Hardware Security Modules (HSMs), smart cards, and supporting software.

7. Developments in the short and medium term

As outlined in [27] and further emphasized in the *2023 Quantum Threat Timeline Report*¹⁸ (see Figure 22), the quantum threat is expected to reach its peak within the next decade. One of the most concerning scenarios involves the so-called *store now, decrypt later* attack strategy. In this model, adversaries intercept and store encrypted data today – despite its current inaccessibility – anticipating that future quantum computing advancements will enable its decryption. This threat undermines the long-term confidentiality guarantees of both symmetric and asymmetric cryptographic schemes.

To mitigate this risk, it is imperative to introduce post-quantum key exchange mechanisms – such as *Kyber*, a lattice-based Key Encapsulation Mechanism (KEM) – as early as possible. Given the significant time and complexity involved in transitioning entire infrastructures, proactive deployment of quantum-resistant key exchange in existing systems is a practical and urgent first step.

Simultaneously, the National Institute of Standards and Technology (NIST) recommends the development of a Quantum-Readiness Roadmap. This roadmap should prioritize the identification and migration of critical systems, assets, and cryptographic processes to quantum-safe alternatives, with the goal of achieving *crypto-agility* – the ability to rapidly adapt cryptographic algorithms in response to evolving threats.

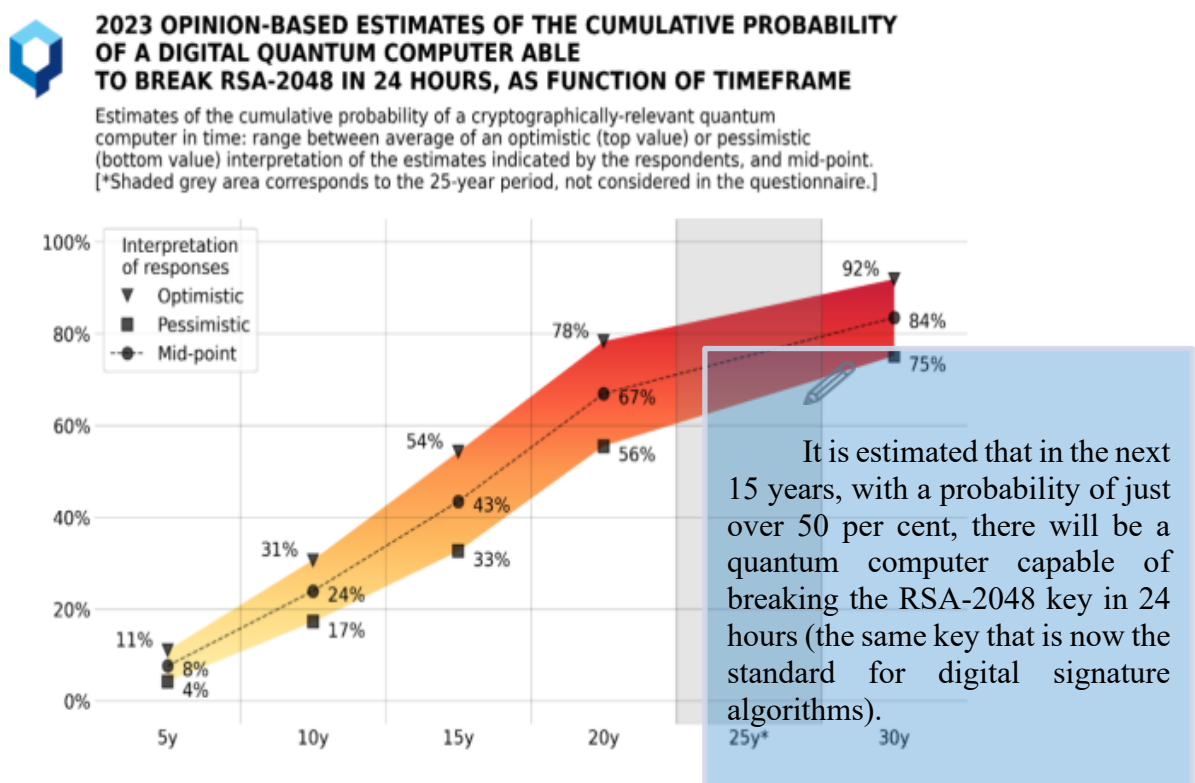


Figure 22 - Source EvolutionQ Inc¹⁹

From the perspective of digital signatures, however, market adoption of post-quantum algorithms remains limited. This hesitation is partly due to the ongoing "Additional Digital Signature Candidates" process within the NIST PQC standardization effort, which is actively seeking to expand the set of signature schemes to be formally standardized.

As shown in Figure 23, several Key Encapsulation Mechanisms (KEMs) have been selected for further evaluation in Round 4 of the NIST PQC process and may be standardized in the near future.

¹⁸ [2023 Quantum Threat Timeline Report - Global Risk Institute.](#)

¹⁹ <https://globalriskinstitute.org/publication/2023-quantum-threat-timeline-report/>

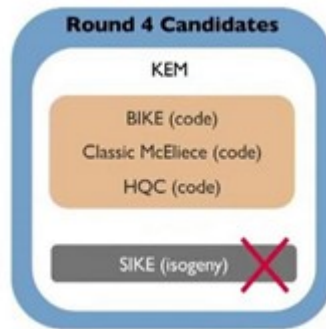


Figure 23 - Candidate Round4 [28]

The Proof of Concept (PoC) conducted as part of this study enabled a preliminary assessment of currently available quantum-safe algorithms. Moving forward, we plan to expand the scope of our testing activities, positioning ourselves to be among the first to evaluate and integrate the forthcoming algorithms that will be formally recognized and standardized by NIST.

8. Definitions

➤ *Grover Algorithm*

Grover's algorithm is a quantum algorithm that significantly accelerates the search process in an unordered database, reducing the search time from $\mathcal{O}(n)$ in classical algorithms to $\mathcal{O}(\sqrt{n})$ in the quantum model. Although originally designed for database search, the algorithm serves as a general-purpose technique to accelerate various search-related computational tasks.

➤ *SHOR algorithm*

Shor's algorithm is a quantum algorithm capable of efficiently solving the integer factorization and discrete logarithm problems. When implemented on a quantum computer, it reduces the computational complexity of these problems from exponential to polynomial time, thereby undermining the security of widely used classical public-key cryptographic systems such as RSA and ECC.

➤ *KEM (key encapsulation mechanism)*

A Key Encapsulation Mechanism (KEM) is a cryptographic primitive used to securely establish a shared secret between parties prior to initiating an encrypted communication session. It consists of three main functions:

- **KeyGen():** Generates a key pair consisting of a public key and a private key.
- **Encapsulate(public key):** Outputs a ciphertext and a shared secret.
- **Decapsulate(ciphertext, private key):** Recovers the shared secret from the ciphertext using the private key.

Although KEMs resemble public-key encryption (PKE) schemes, they are optimized for efficiency and better suited for hybrid encryption systems. Unlike traditional key exchange methods such as Diffie-Hellman – which are vulnerable to quantum attacks and susceptible to man-in-the-middle (MITM) attacks unless certificates are used – KEMs offer a post-quantum secure alternative.

➤ *Security level*

The security level of a cryptographic primitive quantifies its resistance against brute-force or algorithmic attacks. It is typically expressed in terms of security bits, where an n -bit security level implies that 2^n operations would be required to compromise the algorithm. For example, AES-128 provides 128-bit security, which is considered roughly equivalent in strength to RSA with a 3072-bit key.

➤ *Lattices*

In cryptography, a lattice is a discrete set of points in an n -dimensional space formed by all integer linear combinations of a set of linearly independent basis vectors. Lattice-based cryptography leverages the computational hardness of problems such as the Shortest Vector Problem (SVP) and Ring Learning With Errors (Ring-LWE), which are believed to be resistant to quantum attacks.

In a typical lattice-based cryptosystem, the private key consists of a well-structured basis for the lattice, while the public key is a "noisy" version of the lattice. The sender encodes a message by mapping it onto a lattice point and adding noise, and the receiver – knowing the original basis – can recover the original message. An attacker, lacking this basis, finds it computationally infeasible to retrieve the message due to the difficulty of lattice problems.

➤ *NTRU*

NTRU is an open-source, lattice-based public-key cryptographic system designed to provide secure encryption and digital signatures. It comprises two algorithms:

- NTRUEncrypt for encryption,
- NTRUSign for digital signatures.

NTRU is designed to offer strong post-quantum security and high performance, making it suitable for resource-constrained environments.

➤ *FileProtector*

FileProtector is a Java-based software tool developed by Actalis that is compliant with the eIDAS regulation. It enables users to digitally sign documents, encrypt files, and verify signed content, supporting batch processing for high-volume use cases.

➤ *DSS - Digital Signature Service*

The Digital Signature Service (DSS) is an open-source software library developed to support the creation and validation of electronic signatures in compliance with European eIDAS regulations. DSS is freely available on the European Commission's website²⁰ and supports multiple signature formats and validation policies.

➤ *Side Channel attack*

A side-channel attack exploits physical characteristics of a cryptographic device – such as power consumption, electromagnetic emissions, or timing information – to extract secret data. In the context of cryptography, these attacks may be used to reconstruct private keys or other sensitive cryptographic material. Countermeasures typically involve algorithmic masking or noise introduction to obfuscate exploitable patterns, though these techniques can increase execution time and memory usage.

➤ *Computational complexity*

Computational complexity provides a framework for analyzing the efficiency of algorithms, independent of implementation-specific variables such as programming language or hardware. It is often expressed using Big-O notation, which describes the upper bound of an algorithm's running time as a function of input size n .

Formally, if $T(n)$ represents the execution time of an algorithm with input size n , we say that

$$T(n) = O(f(n))$$

if there exist constants $c > 0$ and n_0 such that for all $n > n_0$:

$$T(n) \leq c \cdot f(n)$$

This formalism allows objective comparison of algorithmic efficiency across different computational problems.

²⁰ <https://ec.europa.eu/digital-building-blocks/sites/display/DIGITAL/Digital+Signature+Service+-++DSS>

Index of figures

Figure 1 - Harvest now, decrypt later	10
Figure 2 - TLS protocol and its vulnerabilities	12
Figure 3 - Key and Signature Size Chart – Dilithium	14
Figure 4 - Key and Signature Size Chart - Falcon	15
Figure 5 - Overview of the KEM protocol - Source: [15]	16
Figure 6 - Message signature	18
Figure 7 - Representation and comparison of different types of certificates (Source: Sectigo)	21
Figure 8 - ASN.1 representation of the certificate with evidence on PQ attributes - IETF implementation	22
Figure 9 - ASN.1 representation of the certificate with evidence on PQ attributes - Crossingtud implementation	23
Figure 10 - Size of cryptographic objects across algorithms	28
Figure 11 - Comparison of key creation times (ms)	29
Figure 12 - Sum of times (ms) as the number of occurrences increases	29
Figure 13 - Comparison among the most efficient	29
Figure 14 - Digital signature times (ms)	30
Figure 15 - Digital signature times (ms) for a 20 MB file	30
Figure 16 - Digital signature time comparison	31
Figure 17 - Signature verification times (ms)	31
Figure 18 - Signature verification times for large files	32
Figure 19 - Signature verification time comparison	32
Figure 20 - Performance of signature algorithms	33
Figure 21 - Comparison of execution times in the creation of a CSR	33
Figure 22 - Source EvolutionQ Inc	36
Figure 23 - Candidate Round4 [28]	37

Bibliography

- [1] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer" [Online]. Available: <https://arxiv.org/pdf/quant-ph/9508027>. [Accessed 11 Feb 2025].
- [2] L. Chen, S. Jordan, D. Moody, R. Peralta, R. Perlner, D. Smith-Tone and L. Yi-Kai, "Report on Post-Quantum Cryptography".
- [3] E. Barker, "Recommendation for Key Management: Part 1 – General, p. 30" [Online]. Available: <https://doi.org/10.6028/NIST.SP.800-57pt1r5>. [Accessed 11 Feb 2025].
- [4] NIST, "Let's Get Ready to Rumble The NIST PQC Competition" [Online]. Available: https://csrc.nist.gov/CSRC/media/Presentations/Let-s-Get-Ready-to-Rumble-The-NIST-PQC-Competiti/images-media/PQCrypto-April2018_Moody.pdf. [Accessed 11 Feb 2025].
- [5] L. Grover, "A fast quantum mechanical algorithm for database search. In: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, pp. 212–219 (1996)" [Online]. Available: <https://dl.acm.org/doi/pdf/10.1145/237814.237866>. [Accessed 11 Feb 2025].
- [6] G. Brassard, P. Høyer and A. Tapp, "Quantum cryptanalysis of hash and claw-free functions. ACM Sigact News 28(2), 14–19", 1997. [Online]. Available: <http://arxiv.org/abs/quant-ph/9705002>. [Accessed 11 Feb 2025].
- [7] NIST, "NIST Releases First 3 Finalized Post-Quantum Encryption Standards", 13 August 2024. [Online]. Available: <https://www.nist.gov/news-events/news/2024/08/nist-releases-first-3-finalized-post-quantum-encryption-standards>. [Accessed 11 Feb 2025].
- [8] Federal Information Processing Standards Publication, "FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard", 13 August 2024. [Online]. Available: <https://doi.org/10.6028/NIST.FIPS.203>. [Accessed 11 Feb 2025].
- [9] Federal Information Processing Standards Publication, "FIPS 204: Module-Lattice-Based Digital Signature Standard", 13 August 2024. [Online]. Available: <https://doi.org/10.6028/NIST.FIPS.204>. [Accessed 11 Feb 2025].
- [10] Federal Information Processing Standards Publication, "FIPS 205: Stateless Hash-Based Digital Signature Standard", 13 August 2024. [Online]. Available: <https://doi.org/10.6028/NIST.FIPS.205>. [Accessed 11 Feb 2025].
- [11] M. Vidaković and K. Miličević, "Performance and Applicability of Post-Quantum Digital Signature Algorithms in Resource-Constrained Environments", 13 November 2023. [Online]. Available: <https://www.mdpi.com/1999-4893/16/11/518>. [Accessed 11 Feb 2025].
- [12] ETSI, "TR 103 616 V1.1.1 (2021-09) CYBER; Quantum-Safe Signatures" [Online]. Available: https://www.etsi.org/deliver/etsi_tr/103600_103699/103616/01.01.01_60/tr_103616v010101p.pdf. [Accessed 11 Feb 2025].

- [13] S. Bai, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler and D. Stehlé, "CRYSTALS-Dilithium - Algorithm Specifications and Supporting Documentation" [Online]. Available: <https://pq-crystals.org/dilithium/data/dilithium-specification-round3.pdf>.
- [14] P. Chandramouli, M. Raavi, C. Sang-Yoon, S. Wuthier and Z. Xiaobo, "Performance Characterization of Post-Quantum Digital Certificates" [Online]. Available: <https://par.nsf.gov/servlets/purl/10324007>. [Accessed 11 Feb 2025].
- [15] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. Schanck, P. Schwabe and D. Stehle, *CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM*.
- [16] "CRYSTALS - Cryptographic Suite for Algebraic Lattices" [Online]. Available: <https://pq-crystals.org/kyber/>. [Accessed 11 Feb 2025].
- [17] 4CB, "Eurosystem Single Market Infrastructure Gateway", March 2023. [Online]. Available: https://www.ecb.europa.eu/paym/target/consolidation/profuse/shared/pdf/ESMIG_UDFS_R2_023.JUN_clean_20230303.en.pdf. [Accessed 11 Feb 2025].
- [18] Banca d'Italia, "Concession contracts for the provision of connectivity", 8 July 2019. [Online]. Available: https://www.bancaditalia.it/media/comunicati/documenti/2019-02/CS-Stipula_contratto_ESMIG_Eng.pdf?language_id=1. [Accessed 11 Feb 2025].
- [19] H. A. Mantar, M. S. Kiraz and M. Y. Kubilay, in *A new PKI model with Certificate Transparency based on blockchain*, Computers & Security, 2019, pp. Volume 85, Pages 333-352.
- [20] D. Choffnes, D. Levin, B. Maggs, A. Mislove, A. Schulman, W. Tome, C. Wilson, L. Yabing and L. Zhang, "An End-to-End Measurement of Certificate Revocation in the Web's PKI", 2015.
- [21] B. Huang, E. Liu, H. Tang, H. Wu, L. Zhenguang and Y. Zhuan, "Blockchain for Finance: A Survey", *IET Research Journals*, 2024.
- [22] C. Brunner, D. Engel, F. Knirsch and A. Unterweger, "A Comparison of Blockchain-based PKI Implementations", Center for Secure Energy Informatics, Salzburg University of Applied Sciences.
- [23] E. Buccioli and P. Tiberi, "Quantum safe payment systems", Banca d'Italia: Markets, Infrastructures, Payment Systems - N. 35, 2023.
- [24] L. Tessler and T. Byrnes, "Bitcoin and quantum computing", 2018. [Online]. Available: <https://arxiv.org/abs/1711.04235>. [Accessed 11 Feb 2025].
- [25] A. Truskovsky, "Multiple Public-Key Algorithm X.509 Certificates", Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks; itu-t-x509-2019. [Online]. Available: <https://datatracker.ietf.org/doc/html/draft-truskovsky-lamps-pq-hybrid-x509-02>, superseded by itu-t-x509-2019 (P 14.3), https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-X.509-201910-I!!PDF-E&type=items. [Accessed 11 Feb 2025].

- [26] (OQS), "Open Quantum Safe project, openssl-hybrid-certificates, Technical University of Darmstadt (Germany)" [Online]. Available: <https://github.com/CROSSINGTUD/openssl-hybrid-certificates>. [Accessed 11 Feb 2025].
- [27] Google, "Google's Threat model for Post-Quantum Cryptography" [Online]. Available: <https://bughunters.google.com/blog/5108747984306176/google-s-threat-model-for-post-quantum-cryptography>. [Accessed 11 Feb 2025].
- [28] NIST, "PQC Standardization Process: Announcing Four Candidates to be Standardized, Plus Fourth Round Candidates" [Online]. Available: <https://csrc.nist.gov/News/2022/pqc-candidates-to-be-standardized-and-round-4>. [Accessed 11 Feb 2025].

RECENTLY PUBLISHED PAPERS IN THE 'MARKETS, INFRASTRUCTURES, PAYMENT SYSTEMS' SERIES

- n. 27 Statistical and forecasting use of electronic payment transactions: collaboration between Bank of Italy and Istat, *by Guerino Ardizzi and Alessandra Righi* (INSTITUTIONAL ISSUES) (in Italian)
- n. 28 TIPS: a zero-downtime platform powered by automation, *by Gianluca Caricato, Marco Capotosto, Silvio Orsini and Pietro Tiberi* (RESEARCH PAPERS)
- n. 29 TARGET2 analytical tools for regulatory compliance, *by Marc Glowka, Alexander Müller, Livia Polo Friz, Sara Testi, Massimo Valentini and Stefano Vespucci* (INSTITUTIONAL ISSUES)
- n. 30 The security of retail payment instruments: evidence from supervisory data, *by Massimiliano Cologgi* (RESEARCH PAPERS)
- n. 31 Open Banking in the payment system: infrastructural evolution, innovation and security, supervisory and oversight practices, *by Roberto Pellitteri, Ravenio Parrini, Carlo Cafarotti and Benedetto Andrea De Vendictis* (INSTITUTIONAL ISSUES) (in Italian)
- n. 32 Banks' liquidity transformation rate: determinants and impact on lending, *by Raffaele Lenzi, Stefano Nobili, Filippo Perazzoli and Rosario Romeo* (RESEARCH PAPERS)
- n. 33 Investor behavior under market stress: evidence from the Italian sovereign bond market, *by Onofrio Panzarino* (RESEARCH PAPERS)
- n. 34 Siamese neural networks for detecting banknote printing defects, *by Katia Boria, Andrea Luciani, Sabina Marchetti and Marco Viticoli* (RESEARCH PAPERS) (in Italian)
- n. 35 Quantum safe payment systems, *by Elena Buccioli and Pietro Tiberi*
- n. 36 Investigating the determinants of corporate bond credit spreads in the euro area, *by Simone Letta and Pasquale Mirante*
- n. 37 Smart Derivative Contracts in DatalogMTL, *by Andrea Colombo, Luigi Bellomarini, Stefano Ceri and Eleonora Laurenza*
- n. 38 Making it through the (crypto) winter: facts, figures and policy issues, *by Guerino Ardizzi, Marco Bevilacqua, Emanuela Cerrato and Alberto Di Iorio*
- n. 39 The Emissions Trading System of the European Union (EU ETS), *by Mauro Bufano, Fabio Capasso, Johnny Di Giampaolo and Nicola Pellegrini* (in Italian)
- n. 40 Banknote migration and the estimation of circulation in euro area countries: the Italian case, *by Claudio Doria, Gianluca Maddaloni, Giuseppina Marocchi, Ferdinando Sasso, Luca Serrai and Simonetta Zappa* (in Italian)
- n. 41 Assessing credit risk sensitivity to climate and energy shocks, *by Stefano Di Virgilio, Ivan Faiella, Alessandro Mistretta and Simone Narizzano*
- n. 42 Report on the payment attitudes of consumers in Italy: results from the ECB Space 2022 survey, *by Gabriele Coletti, Alberto Di Iorio, Emanuele Pimpini and Giorgia Rocco*
- n. 43 A service architecture for an enhanced Cyber Threat Intelligence capability and its value for the cyber resilience of Financial Market Infrastructures, *by Giuseppe Amato, Simone Ciccarone, Pasquale Digregorio and Giuseppe Natalucci*
- n. 44 Fine-tuning large language models for financial markets via ontological reasoning, *by Teodoro Baldazzi, Luigi Bellomarini, Stefano Ceri, Andrea Colombo, Andrea Gentili and Emanuel Sallinger*
- n. 45 Sustainability at shareholder meetings in France, Germany and Italy, *by Tiziana De Stefano, Giuseppe Buscemi and Marco Fanari* (in Italian)

- n. 46 Money market rate stabilization systems over the last 20 years: the role of the minimum reserve requirement, *by Patrizia Ceccacci, Barbara Mazzetta, Stefano Nobili, Filippo Perazzoli and Mattia Persico*
- n. 47 Technology providers in the payment sector: market and regulatory developments, *by Emanuela Cerrato, Enrica Detto, Daniele Natalizi, Federico Semorile and Fabio Zuffranieri*
- n. 48 The fundamental role of the repo market and central clearing, *by Cristina Di Luigi, Antonio Perrella and Alessio Ruggieri*
- n. 49 From Public to Internal Capital Markets: The Effects of Affiliated IPOs on Group Firms, *by Luana Zaccaria, Simone Narizzano, Francesco Savino and Antonio Scalia*
- n. 50 Byzantine Fault Tolerant consensus with confidential quorum certificate for a Central Bank DLT, *by Marco Benedetti, Francesco De Sclavis, Marco Favorito, Giuseppe Galano, Sara Giammusso, Antonio Muci and Matteo Nardelli*
- n. 51 Environmental data and scores: lost in translation, *by Enrico Bernardini, Marco Fanari, Enrico Foscolo and Francesco Ruggiero*
- n. 52 How important are ESG factors for banks' cost of debt? An empirical investigation, *by Stefano Nobili, Mattia Persico and Rosario Romeo*
- n. 53 The Bank of Italy's statistical model for the credit assessment of non-financial firms, *by Simone Narizzano, Marco Orlandi and Antonio Scalia*
- n. 54 The revision of PSD2 and the interplay with MiCAR in the rules governing payment services: evolution or revolution?, *by Mattia Suardi*
- n. 55 Rating the Raters. A Central Bank Perspective, *by Francesco Columba, Federica Orsini and Stefano Tranquillo*
- n. 56 A general framework to assess the smooth implementation of monetary policy: an application to the introduction of the digital euro, *by Annalisa De Nicola and Michelina Lo Russo*
- n. 57 The German and Italian Government Bond Markets: The Role of Banks versus Non-Banks. A joint study by Banca d'Italia and Bundesbank, *by Puriya Abbassi, Michele Leonardo Bianchi, Daniela Della Gatta, Raffaele Gallo, Hanna Gohlke, Daniel Krause, Arianna Miglietta, Luca Moller, Jens Orben, Onofrio Panzarino, Dario Ruzzi, Willy Scherrieble and Michael Schmidt*
- n. 58 Chat Bankman-Fried? An Exploration of LLM Alignment in Finance, *by Claudia Biancotti, Carolina Camassa, Andrea Coletta, Oliver Giudice and Aldo Glielmo*
- n. 59 Modelling transition risk-adjusted probability of default, *by Manuel Cugliari, Alessandra Iannamorelli and Federica Vassalli*
- n. 60 The use of Banca d'Italia's credit assessment system for Italian non-financial firms within the Eurosystem's collateral framework, *by Stefano Di Virgilio, Alessandra Iannamorelli, Francesco Monterisi and Simone Narizzano*
- n. 61 Fintech Classification Methodology, *by Alessandro Lentini, Daniela Elena Munteanu and Fabrizio Zennaro*
- n. 62 The Rise of Climate Risks: Evidence from Expected Default Frequencies for Firms, *by Matilde Faralli and Francesco Ruggiero*
- n. 63 Exploratory survey of the Italian market for cybersecurity testing services, *by Anna Barcheri, Luca Bastianelli, Tommaso Curcio, Luca De Angelis, Paolo De Joannon, Gianluca Ralli and Diego Ruggeri*