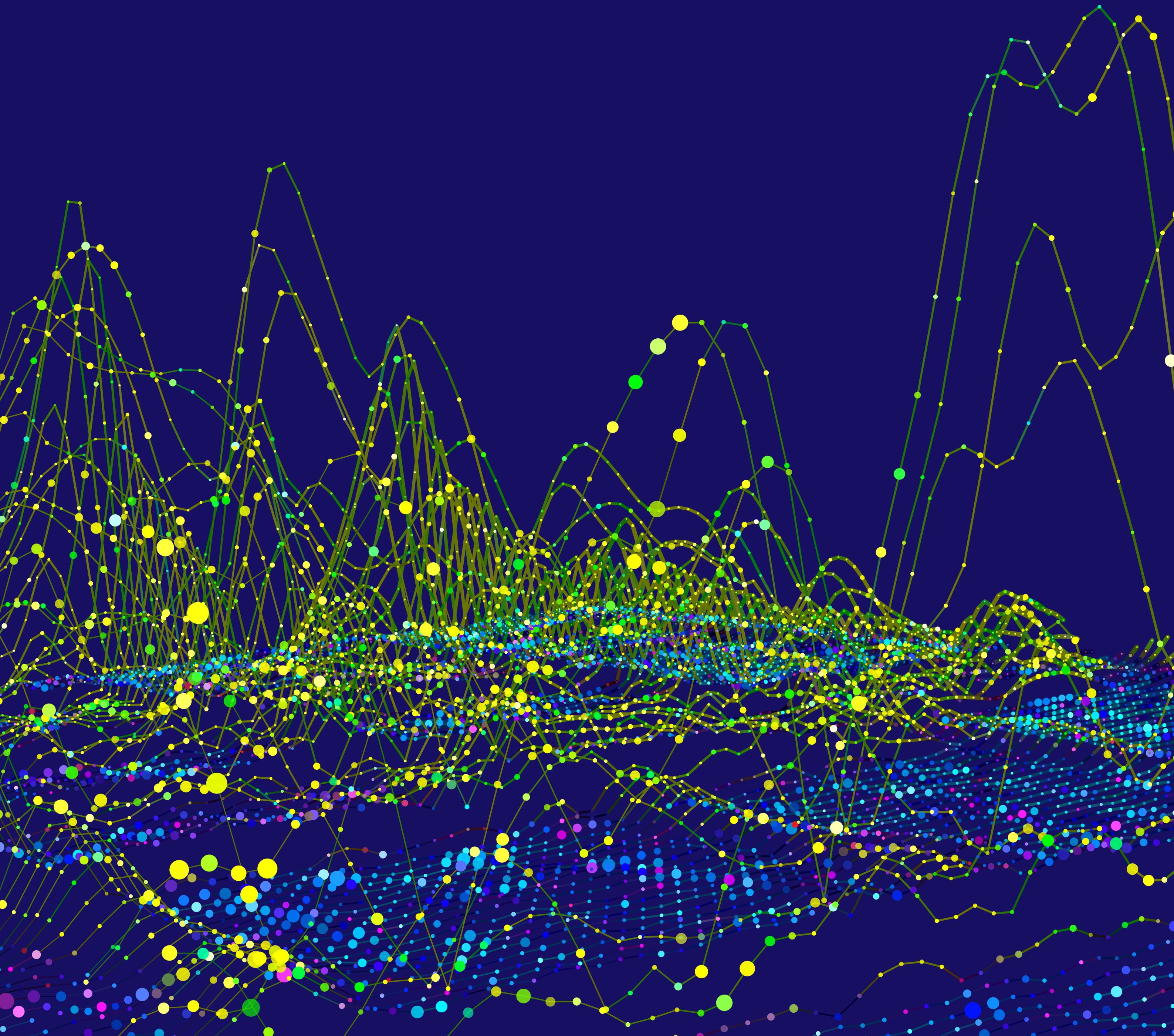
SECURING CENTRAL BANK REGULATORY FILES TRANSFERS WITH POST-QUANTUM CRYPTOGRAPHY (PQC)

BANQUE DE FRANCE

EUROSYSTÈME



1. Sponsors

When we started our experiments on quantum technologies, we were convinced of how strategic post-quantum cryptography (PQC) would become for the financial sector and beyond. The quantum cyber threat is no longer a distant prospect, but a challenge demanding collective anticipation and concrete action.

Since 2022, the Banque de France has conducted several PQC experiments, both internally and with partners such as the Eurosystem centre of the BIS Innovation Hub and the Monetary Authority of Singapore. As part of this continued effort, we were happy to collaborate with Allianz France on a new use case: securing central bank regulatory file transfers with PQC, a core function for financial stability and supervision.

This experiment is at the forefront in exploring the integration of NIST-standardised PQC algorithms into communications between a supervised entity and the central bank. Several scenarios were implemented to test end-to-end protection while preserving existing applications and identity systems.

The results confirm that quantum-safe infrastructures are within reach, and that crypto-agility must become a foundational principle of modern cybersecurity. Building quantum resilience is not only about securing tomorrow's assets; it also starts with protecting today's data.

We thank Allianz for their engagement and look forward to continuing this essential work with all our institutional and industry partners.

Olivier Lantran
Head of the Banque de France Innovation Centre

The arrival of the quantum era will transform the future of insurance and finance, and at Allianz, we are embracing this new horizon with excitement and foresight. This is a remarkable opportunity to innovate, and our commitment to our customers is stronger than ever.

As we lead this transformation, we are pioneering new ways to ensure the trust, data privacy, and operational resilience that form the bedrock of our business.

We are building a quantum-safe future for our clients. By anticipating challenges like "harvest now, decrypt later," we are turning potential vulnerabilities into strengths. As guardians of our customers' most sensitive information, our early adoption of post-quantum cryptography is a powerful statement of our commitment to enduring trust, future-focused regulatory alignment, and our role as a technological leader.

Our landmark pilot with Banque de France has been a resounding success. We have demonstrated that next-generation quantum-safe encryption can be integrated seamlessly into today's real-world infrastructures, enhancing security without any operational disruption. This project confirms that a quantum-ready future is not a distant goal, it is achievable now. By implementing hybrid cryptographic models built on emerging global standards, we are securing our vital communication channels today and building a resilient foundation for tomorrow.

Our path forward is clear: we will continue to secure our most critical data flows and enhance our infrastructure for long-term resilience.

This achievement, made possible through our powerful collaboration with Banque de France and CryptoNext Security, sets a new benchmark for the industry. It showcases how shared vision, and strong partnerships can accelerate innovation. Allianz is proud to be at the forefront of quantum-safe technology, securing our ecosystem, protecting customer data, and shaping the future of our industry.

We are very proud of this collaboration and eagerly anticipate our continued partnership as we confidently lead the insurance industry into the quantum era.

Karim AKHAVAN-HEZAVEI – Head of Group Technology (CIO|CTO)
& Head of COO Transformation Allianz SE
Ralf SCHNEIDER – Senior Fellow and Head of Cybersecurity
and NextGenIT Think Tank (CXTT) Allianz SE
Loic TOURLOURAT – Chief Information Office Allianz FR

2. Executive summary

The Banque de France and Allianz France have collaborated on a joint experiment to **secure regulatory data exchanges using post-quantum cryptography** (PQC), in anticipation of the quantum threats expected to undermine traditional public-key cryptography. The project focused on protecting regulatory file transfers submitted by Allianz through the Banque de France's OneGate platform, a core infrastructure used by supervised entities, to meet their weekly reporting obligations.

The experiment aimed to evaluate the integration of newly standardized PQC algorithms, validated by the National Institute of Standards and Technology in August 2024, into existing client-server communications, without the need for modifications to critical legacy systems. To achieve this, a hybrid approach was implemented that encapsulated existing communication flows secured with traditional cryptography within a quantum-safe layer, without modifying the core applications or authentication systems.

This tunnelling approach enabled end-to-end protection of sensitive data against both immediate and future quantum threats, including store-now-decrypt-later attacks and potential authentication impersonation. The project complied with the latest NIST standards and European cybersecurity guidance, and is in alignment with ongoing standardisation processes. By demonstrating the feasibility of PQC implementations even in constrained environments, **the project offers a concrete transition pathway towards post-quantum readiness**.

This initiative marks a significant milestone in the practical adoption of PQC within financial infrastructures. **It reinforces the need for crypto-agile architectures, and early alignment with international standards while preparing for ambitious migrations strategies at the ecosystemic level.** The Banque de France and Allianz France reaffirm their commitment to accelerating quantum-safe adoption and strengthening the collective resilience of financial systems against future cryptographic disruptions.

3. Introduction

3.1 From anticipating to preparing for quantum threats

The arrival of quantum computing represents a major technological breakthrough whose implications are no longer theoretical. It brings promising prospects for numerous industries such as pharmacology, meteorology, logistics, finance, and others, while also raising new challenges. **Today's most pressing priority remains the mitigation of cybersecurity risks as the advent of quantum computing calls into question the long-term viability of current encryption standards.**

Encryption standards, such as RSA and Elliptic Curve Cryptography (ECC) algorithms are a cornerstone of today's secure digital communication, ensuring the confidentiality of our exchanges, and the protection and integrity of our data. Their robustness is based on the impossibility of solving factorisation problems within a "reasonable" time frame using classical computers. With Shor's algorithm and sufficiently powerful quantum machines, this constraint will no longer hold, exposing current encryption standards to major risks. Once these machines reach sufficient maturity, they will be able to break the encryption standards that underpin most of today's secure communications, such as authentication and digital signatures.

This is not just a concern for the future for the Banque de France and Allianz France. Already today, malicious actors could be intercepting and storing encrypted data with the intention of decrypting them once quantum capabilities become available, even several years later. This "store now, decrypt later" (SNDL) strategy poses a threat to the long-term confidentiality of customer records, sensitive financial data transactions or regulatory filings.

In response to this growing risk, on 13 August 2024 the National Institute of Standards and Technology (NIST) published the first standardized post-quantum cryptography (PQC) algorithms, following research and a selection process initiated in 2016. These new standards, already adopted and implemented by big tech companies, now simplify the testing and integration of PQC into existing infrastructure.

The Banque de France (BDF) – as a national institution committed to the subject of digital sovereignty since it started experimenting in 2022 – partnered with industry leader Allianz France (ALV) and PQC software specialist CryptoNext Security (CNS) to launch a joint testing project to test these validated PQC algorithms in real-life business applications. This project translates international standards into operational practice, making BDF and Allianz early actors in preparing the financial sector for post-quantum resilience and providing practical insights into how PQC can be implemented without disrupting operations.

3.2 Previous Banque de France PQC projects

Since 2022, BDF has pursued a structured and progressive approach to explore the feasibility of PQC integration. This strategy has relied on successive experiments, international cooperation and active engagement with emerging international standards.

The BDF first internal proof of concept, launched in 2022, demonstrated the feasibility of securing communication channels with PQC by implementing a quantum-safe IPSec VPN tunnel (IPsec VPN uses a technique called “tunnelling” to encrypt the data that is being sent between a device and the VPN server). The results were promising, showing that advanced algorithms could be integrated with minimal overheads, thereby maintaining system performance and enhancing security. Beyond the technical validation, this initiative also positioned BDF within the international ecosystem and prepared the ground for subsequent collaborations.

In 2023, BDF engaged in its first international cooperation through the LEAP I project with the Bank for International Settlements Innovation Hub (BISIH) and Deutsche Bundesbank (BBK). This project aimed to secure payments exchanged through an IPSec VPN tunnel using quantum-resistant algorithms. While the technical feasibility was again confirmed, the primary objective was to measure performance impacts in a critical business context. The experiment provided important insights into the practical implications of PQC in financial transactions, ensuring secure and efficient payment processes within a quantum-resilient framework. In parallel, BDF decided to create a team of PQC-dedicated experts to sustain these efforts.

In 2024, BDF partnered with the Monetary Authority of Singapore (MAS) to explore PQC-secured email communications using widely deployed tools such as Microsoft Outlook. Relying on a plugin developed by CNS, the project confirmed the feasibility of integration, and the compatibility of PQC signatures and encryption, into real-world usage scenarios. It also highlighted emerging challenges, related to crypto-agility, compatibility with existing infrastructures, and certificate lifecycle management. These findings reflected a natural progression from earlier experiments, moving from performance considerations to questions of integration and governance.

Through the iteration of increasingly complex use cases and by strengthening its international partnerships, BDF has established a solid foundation. For its part, ALV recognises the importance of addressing the cybersecurity risks associated with quantum computing and this project offered the opportunity to begin concrete, operational implementation of PQC into its information systems and real-life business applications.

4. Description of the experiment

4.1 Objectives of the experiment

In the context of regulatory reporting obligations, bank and insurance companies, including ALV, are required to share sensitive documents and supervisory information with BDF on a weekly basis through BDF's OneGate application. The solution selected for this project aimed at protecting these critical data exchanges from emerging quantum computing threats, without disrupting existing operations.

The initiative was designed to address four key business objectives.

1. Securing an already deployed infrastructure by implementing validated PQC standards.

2. Minimising necessary adaptations within a distributed model to promote interoperability.

3. Demonstrating that gradual migration/backward compatibility is possible and not imposed unilaterally by one system over the other.

4. Maintaining operational continuity with no changes to the core of the OneGate application.

To meet these objectives, the technical solution aimed to integrate the PQC algorithms ML-KEM (encryption) and ML-DSA (which ensures the authenticity of communicating partners) with the TLS protocol. This created a quantum-safe tunnel around standard TLS encrypted data (hybrid architecture). This approach safeguards communications against potential quantum attacks while ensuring compatibility with existing systems, maintaining the as-is infrastructure, and guaranteeing speed of implementation.

The scenarios of our tests were designed based on the trade-off between the projected likelihood of risks and the implementation complexity of the PQC algorithms (see chapter 4.4) while testing was framed to confirm effectiveness both in terms of performance and security. The implementation offers dual security benefits by ensuring both data confidentiality through encryption, and transaction integrity, and authentication through digital signatures, with no significant performance degradation observed.

This helps establish a framework for securing similar data exchanges across both organisations' regulatory reporting systems. Most significantly, this project aims to prove that PQC can be practically implemented in production environments today, addressing a critical business risk without requiring extensive system modifications or operational disruptions.

To carry out this project, ALV and BDF partnered with CNS as their technical partner for the initiative. CNS is a recognised specialist in PQC solutions and supplied, through its Quantum-Safe Crypto Agile Suite, the quantum-safe components that the ALV and BDF teams implemented. CNS also provided technical expertise during the configuration and testing phases.

4.2 Illustration of the current use case

The core infrastructure of the experiment consists of two parts: the ALV network on the client side, and the BDF network on the server side. To establish a secure connection, the client (i) uses an RSA client certificate to be authenticated by the server and (ii) authenticates the server with its TLS RSA server certificate.

From there, the client accesses the OneGate portal via a browser. This set-up will be referred to as the "User-To-Application" (U2A) scenario (figure 1).

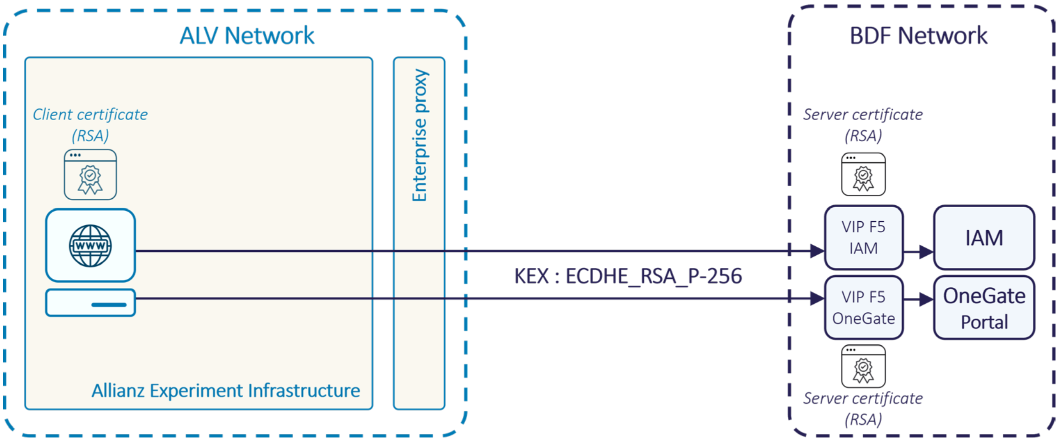


FIGURE 1 – U2A Scenario: Legacy connection from a browser

Alternatively, the client can also call the OneGate web services using a client application. This use case will be referred to as the "Application-To-Application" (A2A) scenario (figure 2).

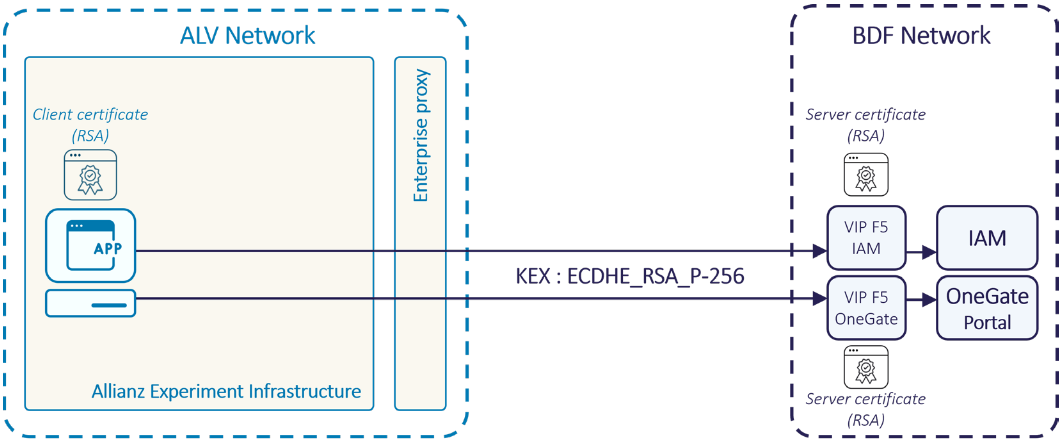


FIGURE 2 – A2A Scenario: Legacy connection from an application

Note that the entry point to the BDF network is a virtual private address (VIP). An Identity Access Management (IAM) component authenticates incoming requests, after which the clients are redirected to a second VIP managed by the F5 Load Balancer for OneGate.

A detailed description of the current components of the use case on the server side and client side, and non-updatable components can be found in appendices A.

4.3 Mitigating anticipated threats

The purpose of this experiment is to address two major classes of threats, each with distinct timelines and technical implications, reflecting both immediate and future risks associated with the advent of quantum computing:

- **Passive interception of encrypted data**, corresponding to the well-documented "*store now, decrypt later*" strategy. It involves intercepting encrypted traffic today with the expectation that it can be decrypted in the future, once quantum computers become powerful enough to break current public-key cryptography. In this use case, this is particularly critical given the sensitivity of the regulatory documents exchanged and the long-term confidentiality they require.

- **Authentication impersonation:** to prevent the use of forged digital signatures or derive private keys that could enable malicious actors to impersonate legitimate actors in secure communications. It can take two forms:

- Client impersonation could facilitate the unauthorised submission of false declarations, retrieval of past submissions, or denial-of-service (DoS) attacks.
- Server impersonation through man-in-the-middle techniques could allow malicious actors to intercept or modify sensitive data exchanges by masquerading as a legitimate endpoint.

By encapsulating classical TLS communications within a quantum-safe tunnel, the proposed solution provides an additional layer of security that mitigates both threat classes. While the hybrid approach does not replace the need for full native PQC integration in the future, it significantly limits the exposure surface during the transition phase, especially for high-value communications transmitted requiring long-term confidentiality.

4.4 Evaluating risks to define priorities

To define the priorities for this experiment, we evaluated each component based on the trade-off between the projected likelihood of risks and implementation complexity, helping us determine where to start and how to progress step by step.

Component	Projected likelihood	Implementation complexity	Priority
TLS endpoint	Immediate	Low	High
Outgoing proxy	Immediate	< Medium	High
Browser	Immediate	Low	High
Client application	Immediate	High	Medium
PKI	< 5 years	Medium	Medium
Web server application	Immediate	> Medium	Low
IAM authentication	> 5 years	High	Low

TABLE 1 – Risk and implementation priority matrix

Based on this analysis, we prioritised securing TLS via a reverse proxy, while considering PKI and client authentication as lower-priority enhancements. Changes to the application server, client, and IAM were intentionally deferred. We did not need to modify the browser, as it already supported KEM, and the implementation of certificate-based authentication was intentionally differed

Having defined the scope, threats and priorities, we then situated the experiment in the broader context of PQC standards and integration strategies.

5. PQC standards and integration strategy

5.1 New PQC algorithm standards

The development of quantum-resistant cryptographic standards has progressed through a global and collaborative process that concluded with the ratification in August 2024 of the first NIST PQC standards. These are now recognised by major institutions and standard settings bodies (SSBs), such as G7, BSI or ANSSI and form the reference framework guiding their own recommendations for cryptographic transitions across all organisations.

The standardisation process began in 2016 when NIST initiated a public competition to identify algorithms capable of withstanding quantum-enabled attacks. Out of 80+initials submissions, and after extensive security analysis and performance testing across multiple evaluation rounds, **three algorithms were established as the new security standards**:

1. **ML-KEM (FIPS 203)** serves as the primary standard for general encryption applications. Originally developed as CRYSTALS-Kyber, this Module-Lattice-based Key-Encapsulation Mechanism creates encryption keys that are notably compact, facilitating efficient exchange between parties while maintaining high operational speed.

2. **ML-DSA (FIPS 204)** functions as the primary standard for digital signature protection in the post-quantum environment. Built on the mathematical foundation of the CRYSTALS-Dilithium algorithm, this Module-Lattice-based Digital Signature Algorithm provides the authentication and non-repudiation capabilities essential for financial communication and transaction integrity.

3. **SLH-DSA (FIPS 205)** provides an alternative post-quantum digital signature standard. Based on the SPHINCS+algorithm, this StateLess-Hash-based Digital Signature Algorithm offers strong security assurances derived solely from the robustness of cryptographic hash functions. While its signatures are generally larger and verification times longer than lattice-based alternatives, it benefits from mature security proofs and avoids reliance on unproven mathematical hardness assumptions, making it particularly relevant for high-assurance or long-term archival use cases.

These standards differ fundamentally from traditional cryptography by relying on mathematical structures resistant to quantum computing attacks. As cybersecurity is a constantly evolving field, the standardisation process at NIST remains ongoing: other KEMs are being investigated such as HQC announced for 2027, and another selection process was started in 2023 for the identification and selection of short signature algorithms (see appendices B). Finally other algorithms such as FrodoKEM have also been published by ISO and promoted in a European context, although current consensus still prioritises the NIST ratified algorithms for practical deployment.

This international standardisation provides a clear technological direction for our BDF-ALV quantum security initiative. It ensures that the experiment is aligned with the trajectory defined by major authorities and SSBs. It also offers the opportunity to engage with PQC algorithms currently considered to be the most relevant for their practical implementation potential in financial infrastructures and high-volume systems.

5.2 Implementation of hybrid principles

5.2.1 Hybridisation standards

The concept of hybridisation in PQC refers to the combination of traditional cryptographic algorithms, such as RSA, ECDSA, or X25519 with newly standardised post-quantum schemes like ML-KEM or ML-DSA. Hybridisation is strongly recommended by recognised SSBs and cybersecurity authorities, like the NIST, ANSSI or BSI, as an essential transitional mechanism to achieve cryptographic resilience during the migration to quantum-safe systems.

As post-quantum algorithms remain relatively new and evolving, combining them with well-established classical algorithms (like X25519 or ECDSA) provides a layered security approach. This ensures that even if one family of algorithms is later found to be vulnerable, whether to quantum or classical attacks, the other can still protect the system and uphold the confidentiality or authenticity of the communication. In addition, hybridisation allows for the flexible exchange of encryption algorithms or key lengths in the "underlying system" while protecting communication with quantum-safe encryption on the "overlay".

Hybrid schemes are particularly valuable for critical use cases like TLS, digital signatures, and secure communications, in high-security contexts where both confidentiality and integrity are paramount. They ensure that communications and data remain protected against known threats, but also against speculative or emergent weaknesses.

The main idea behind hybrid key agreement is that both the client and server agree on two shared secrets: one using traditional Diffie-Hellman key exchange (where each party combines their private key with the other's public key), and another using a post-quantum KEM (where the client sends a public key to the server, and the server responds with an encapsulated shared secret). These two secrets are then combined to derive a single, secure session key (see appendices C).

5.2.2 Applying hybrid standards for encapsulation

There are several ways to combine PQC and classical cryptography. In this experiment, we aimed to test standard hybrid approaches, as well as an encapsulation-based solution that aligns with the security goals of hybrid cryptography, while remaining compatible with existing systems and workflows.

Standard hybridisation approach

Implemented in test: User-To-Application scenario

This approach combines classical cryptography and PQC through TLS key exchange with x25519 +ML-KEM. Client and server authentication is performed using RSA certificates. It complies with existing standards and is relatively simple to implement, as it is already integrated in various mainstream browsers. However, its applicability remains limited, as it does not address client limitations or support certificate-based authentication.

Encapsulation-based hybridisation approach (known as Tunnelling)

Implemented in tests: Application-To-Application and User-To-Application scenarios

In this model, a quantum-safe TLS tunnel is established between the Client Proxy (ALV) and the Server Proxy (BDF) using a hybrid key exchange (x25519 +ML-KEM). Both client and server are authenticated with post-quantum ML-DSA certificates.

The traditional TLS flow between User and App goes uninterrupted through a quantum safe TLS Tunnel from the ALV network to the BDF network. For this flow, a standard x25519 key exchange is used, along with RSA certificates for client and server authentication, ensuring transparency of the application.

Although this method introduces an additional layer, as it involves a traditional TLS session inside a TLS tunnel, it remains effective, allowing the use of hybrid key exchange for establishing the tunnel or for certificate-based authentication.

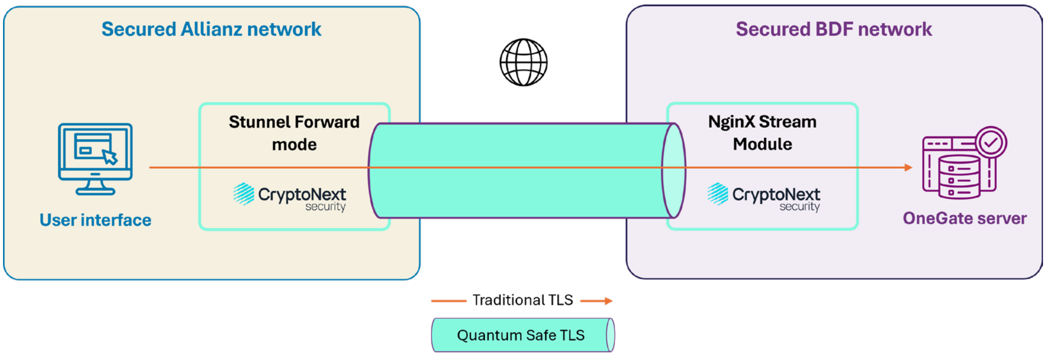


FIGURE 3 – Encapsulation principle

We established an over-encapsulation layer using proxy forwarding with Stunnel (configured in forwarding mode to add encryption) and NGINX (configured in stream mode to handle connection redirection), creating an encrypted tunnel between the client and the server.

This approach offers several key advantages in terms of compatibility:

- Preservation of the OneGate client and server stack: no modifications to the application logic or encryption libraries were required.

- No impact on ALV's existing infrastructure, including proxy and outbound communication flows.

- Alignment with the spirit of emerging standards, by enforcing cryptographic diversity and supporting post-quantum migration in a decoupled, testable manner.

Ultimately, the strategy illustrates how hybridisation principles can be effectively implemented outside of formalised protocol frameworks, especially in constrained or legacy environments.

Encapsulation-based layering represents a viable transitional mechanism that provides tangible security benefits today, while preparing the ground for future adoption of fully standardised hybrid schemes.

6. Experiment

This experiment was designed based on a scenario in which the web application or authentication portal cannot be upgraded, and where multiple clients may be involved. To address this, a reverse proxy capable of handling PQC was implemented in front of the application, a configuration commonly made for load balancing purposes. A similar constraint was addressed on the client side, where the client application cannot be upgraded (browser or Java). In this case, a forward proxy is required to establish a secure connection to a PQC-enabled server.

In addition, we also tested a scenario in which the web client (browser) was already compatible with some PQC key exchange algorithms and connected directly to BDF's PQC reverse proxy, without the use of a PQC forward proxy.

Through this experiment, three critical dimensions were addressed: key exchange (KEX/KEM), server authentication, and client authentication. We also considered the limitations imposed by the organisation's outgoing proxy, which is often mandatory for accessing external services.

Finally, instead of relying on manually generated certificates (e.g. with OpenSSL), we decided to deploy a proper PKI (EJBCA) to verify the feasibility of PQC PKI deployment, as our company's current one does not support PQC.

It should be noted that an external BDF cloud environment was used to host the reverse proxy (RP) to save time during the deployment process. However, in a real production environment, it would be deployed within our BDF network to maintain a secure connection between the PQC RP and the traditional server. Client components were deployed on dedicated virtual test machines hosted in ALV's private cloud.

One potential solution commonly considered in such scenarios is building a VPN tunnel secured with PQC. However, this approach was already explored in a previous experiment (see *Project Leap: quantum-proofing the financial system*). Furthermore, it would not have allowed testing of the U2A use case as the client side could not be updated. In the current experiment, we intentionally shift the focus to securing the TLS layer.

6.1 Technical set-up

The main components used in this experiment were NGINX as a reverse proxy, Stunnel as a forward proxy, and EJBCA Community Edition (CE) as the PKI. To enable PQC compatibility of both NGINX and Stunnel, we used a customised version of OpenSSL provided by CNS, a contributor to the OpenSSL open-source project.

We also used a CNS custom build of EJBCA CE, as the standard was still evolving at the time of our tests, and not all algorithms were supported by the official release. CNS expertise was used on two levels:

1. Cryptography, as they participate in standardisation organisations (NCCOE and IETF) and thus have in-depth knowledge of standards (post-quantum TLS protocol and the various hybrid X509 certificates).

2. Software, through the modification of certain software blocks to introduce post-quantum algorithms or features. See appendices D for detailed technical contributions by CNS)

6.2 Remediation approach by components

BDF acts as the server, hosting the business applications, while ALV acts as the client, accessing the business applications. The experiment was deployed in two steps:

- First, ALV and BDF each deployed all the components in their testing infrastructures, enabling local testing without requiring inter-connection with the other participant.

- Second, ALV only kept the required client component, while BDF kept the server components, and the inter-connection was established in-between.

A synthesis table of the remediation matrix by component is provided in appendices E.

6.2.1 Certificate generation

As the generation of PQC certificates is not supported by our company's PKI, an EJBCA CE PKI provided by CNS was implemented. While we could have used manually generated certificates with OpenSSL, this would not have been a realistic approach for a production use case. This also demonstrated the feasibility of setting up a secondary PKI capable of handling PQC, if required.

6.2.2 BDF side

Web portal (OneGate) U2A

The *OneGate* application and the *Identity and Access Management* (IAM) system cannot be upgraded. To support browser-based access, we deployed a reverse proxy in a TLS termination that handles the KEM part of the PQC.

In our use case, client certificate authentication is managed by the reverse proxy. However, in a typical set-up, the reverse proxy should be trusted to forward the client certificate in a header to the IAM system.

This approach allows any client with a compatible browser to perform PQC KEM with our web portal, without requiring any modification on our VIP host, IAM system or underlying network components.

Webservice portal (OneGate) A2A or B2A

Since we couldn't upgrade the application or the IAM system, we decided to deploy a reverse proxy in stream mode (TCP tunnelling). This allows client applications to connect through their forward proxy using PQC. The advantage of this mode is that it enables both PQC client and server authentication, as well as a post-quantum key encapsulation mechanism (KEM).

A quantum-safe tunnel is established between the forward and reverse proxies, using PQC certificate authentication on both ends and PQC KEM. This mode can also be applied to the web portal. When the client browser (supporting PQC or not) connects via the reverse proxy, standard TLS communication occurs through a PQC encrypted tunnel, making the process completely transparent to the client and quantum safe.

6.2.3 ALV Side

Company proxy (SQUID)

In the ALV infrastructure, to access external networks, any flow must go through the company proxy, which does not support PQC algorithms. However, it allows the use of an additional proxy in *CONNECT* mode. Using this forward proxy, we can enable PQC algorithms and ensure compatibility between our application and PQC servers.

To enable the CONNECT mode, we used Stunnel instead of NGINX as a forward proxy, as the latter does not support this configuration. It is important to note that this configuration is possible because the SQUID proxy does not perform TLS traffic interception, i.e. it allows communications to pass through without requiring the ability to decrypt them.

6.3 Functional tests

Three scenarios were tested to validate PQC integration under different constraints.

6.3.1 Scenario – Reverse proxy HTTP

To support PQC key exchange, we added a NGINX HTTP reverse proxy in front of the application and IAM. RSA certificates are maintained for both client and server authentication, ensuring compatibility with existing client.

Solution	Pro	Con
Reverse proxy HTTP	• Transparent for both the application and the client • Can be deployed in front of most applications that can be proxied (through client certificate forwarding) • RP often already exists in front of application (VIP LB) • No component on client side • When the client browser or application supports PQC certificates, the RP will offer full protection (KEM and authentication) • Allows traditional KEX for non PQC compatible clients	• Client must be compatible with Hybrid PQC KEX/KEM to ensure quantum-safe communication • Authentication is performed using traditional certificates, not post-quantum (PQC) certificates: – Modern browsers and applications do not yet support server authentication using PQC certificates. – Once client-side support for PQC certificates becomes available, the Identity and Access Management (IAM) system must also be capable of handling PQC client certificates directly.

U2A use case – Scenario 1: Connection using a PQC secured browser without a PQC tunnel

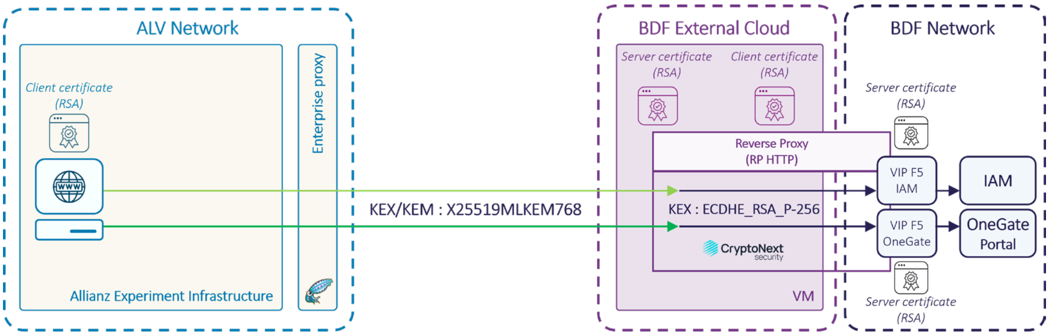


FIGURE 4 – PQC Secured browser without encapsulation (PQC tunnel)

6.3.2 Scenario – Reverse proxy stream with forward proxy

We set up a secure TCP tunnel between a client-side forward proxy and a server-side reverse proxy. The connection was secured using PQC certificates for mutual authentication, along with a hybrid KEX/KEM to ensure quantum-resistant key exchange.

Solution	Pro	Con
Reverse proxy stream	• Client only has a proxy to configure • Enforced authentication through PQC (server side and client side) • PQC KEM protection from the FP to the RP. • Improved crypto-agility: enables support for post-quantum KEMs not yet implemented in browsers	• Needs component on client side • Needs to be set-up between two entities (Company to Company) • No PQC protection upstream and downstream the tunnel. • Even when the browser supports PQC KEM, this capability is not used in conjunction with the forward proxy. The browser continues to establish traditional KEX connections with VIP endpoints.

U2A use case – Scenario 1(b): Connection using any traditional browser through a PQC tunnel with a forward proxy

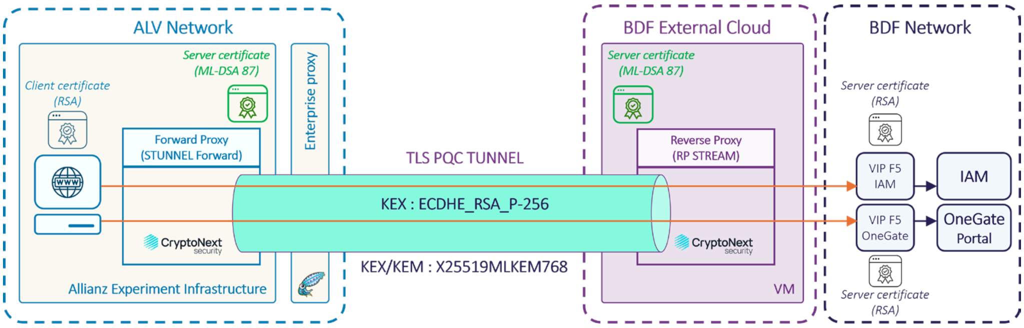


FIGURE 5 – Connection using a browser through a PQC tunnel

A2A use case – Scenario 3: Regulatory file deposit using an application through a PQC tunnel

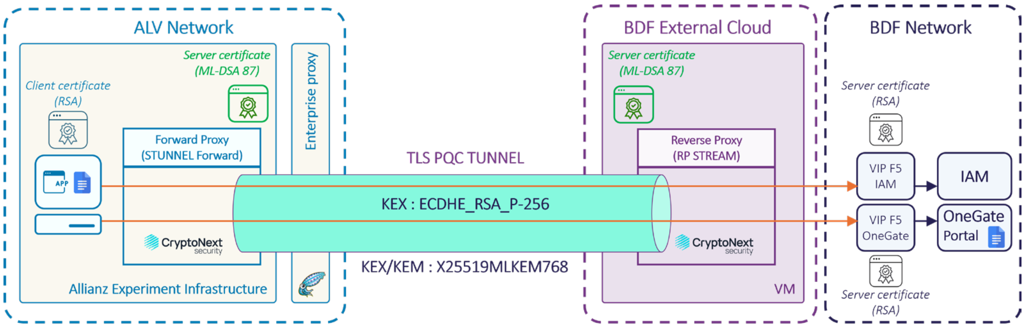


FIGURE 6 – File deposit using an application through PQC channel

6.3.3 Scenario – PKI

To address the incompatibility between our existing corporate PKI and PQC, we deployed a new EJBCA CE PKI dedicated to managing PQC certificate generation.

Using this set-up, we generated the following:

- A Certificate Authority (CA) based on ML-DSA87
- A server authentication certificate (SSL/TLS server)
- A client authentication certificate (SSL/TLS client)

These certificates were used in a scenario involving PQC-based authentication for both the server and the client (see Appendices F for details).

We chose not to use hybrid certificates to keep the use case simple and focused. However, this option remains open for future exploration, particularly if standardised hybrid certificate formats gain broader support among browsers.

7. Project takeaways

This section presents the main findings of the experiment, drawing from empirical observations, technical challenges, and strategic insights gained during the deployment. It aims to highlight both the feasibility and the operational implications of implementing PQC in real-world infrastructures.

7.1 Lessons learned

7.1.1 Operating observations

This experiment demonstrated the technical feasibility of deploying quantum-safe encryption mechanisms on top of existing infrastructures. From a performance standpoint, the use of ML-KEM and ML-DSA within the encapsulation architecture highlighted certain key points:

Computational and network efficiency: PQC key structures present a minimal overhead footprint, while longer key lengths in ML-KEM and ML-DSA introduce marginal computational demands. Network latency remained negligible in our test set-up. CPU consumption shows minimal incremental load. By delegating cryptographic processes to upstream proxies, we preserved application performance integrity without compromising security protocols.

Cryptographic tunnel dynamics: Tests on cryptographic tunnel performance showed a minor but acceptable impact on the TLS handshake time: the initial setup of the TLS connection experienced a slight performance impact due to the added cryptographic layer. However, data transfer performance remained unaffected, regardless of the data size, as encryption relies on conventional AES-256 cryptography.

Load testing revealed that our solution maintains good performance under normal operating conditions. While stress testing showed minor performance variations during peak usage scenarios, these effects remained minimal and were confined to the handshake phase. The enhanced security capabilities provided by this encryption technology deliver substantial long-term protection benefits that far exceed any minor computational adjustments.

These encouraging early results should be validated outside a controlled environment. We recognise that comprehensive real-world validation will be essential to fully demonstrate the system’s operational effectiveness and confirm these promising initial results.

7.1.2 Implementation challenges

Deploying PQC in a real-world enterprise infrastructure revealed several implementation constraints. The PQC migration exposes fundamental limitations in our current network security infrastructure. Existing proxy systems and TLS interception mechanisms face critical incompatibilities with PQC-based TLS key exchange mechanisms.

The core challenges stem from architectural rigidity: traditional network security appliances cannot effectively process, decrypt, or analyse quantum-safe communication protocols using advanced algorithms like ML-KEM and ML-DSA.

This incompatibility extends beyond technical constraints, inducing a structural shift in cybersecurity architecture. PKI frameworks require comprehensive redesigns to support quantum-resistant certificate management, creating a complex technological transition that demands holistic organisational adaptation.

Modern client-side software stacks do not support PQC-based certificates. As a result, implementing a PQC-compatible web service client was not feasible. Modern browsers do not support PQC-based certificates, and neither Java nor Bouncy Castle currently provide support for PQC KEMs or PQC-based certificate handling. As a result, we were unable to develop a web service client capable of using PQC.

Moreover, monitoring the actual use of PQC algorithms remains complex, as it requires specialised tools and network probes (see example in appendices G).

7.2 Insights

The experiment confirmed that post-quantum protections can already be deployed to secure existing communication channels without requiring disruptive modifications to core systems.

However, this approach also highlighted structural complexities. The additional components required on both client and server sides create architectural redundancy that, while effective for experimentation, would not be sustainable at scale. Direct integration of post-quantum encryption into core components presents a more sustainable path forward.

One of the key takeaways is that the experiment confirmed that post-quantum protection remains technically feasible even for non-compatible applications. The most efficient approach targets front-end equipment (SSL endpoints), which typically requires minimal updates while providing maximum protection coverage.

However, the temporary solutions deployed during testing (virtual environments, custom deployment frameworks) are not suitable for operational use or should be limited to critical applications where the implementation of PQC is not going to be available within a reasonable time frame. Critical network infrastructure, particularly enterprise proxies, would benefit from short-term updates to support both current testing and future implementation phases.

This experiment, which involved multiple systems and stakeholders without any substantial modifications to existing components, **validates the industry assessment that a complete transition will require coordinated efforts over a multiple-year time frame**. Early adoption is nonetheless encouraged. Banks and insurers should begin by:

- Adopting quantum-safe algorithms across all cryptographic functions (key exchange, digital signatures, encryption) to build fully resistant communication channels.

- Conducting application-specific testing, as our findings indicate varied compatibility issues when applying post-quantum protocols to different systems. A hybrid approach should involve a detailed assessment of application performance.

The current limited deployment of NIST standards calls for cross-industry coordination to achieve a practical, phased implementation of post-quantum encryption across the financial sector.

7.3 Assessment of the experiment

Through the assessment of the experiment and its objectives (see detailed assessment in Appendices H), valuable experience was gained and findings made into how to upgrade an infrastructure to support PQC, even when some components cannot be immediately upgraded.

It became apparent that it is not necessary to wait for every component in the system to be PQC-ready before starting the migration. For example, PQC certificates are not required to enable PQC-based TLS encryption.

The successful deployment of a PQC-enabled Public Key Infrastructure (PKI) further strengthened confidence in the migration process. By setting up an independent PQC PKI, corporate PKI compatibility was removed from the critical path of the migration. This provided greater flexibility to test new certificate formats and cryptographic algorithms.

Due to the operational uncertainty surrounding the standardisation of hybrid authentication certificates and the fact that cryptographic risk is time-bounded, with the vulnerability window for classical authentication algorithms likely being short enough to remain acceptable in the near term, we decided to use non-hybrid PQC certificates (ML-DSA) for our tests.

In this context, **prioritising crypto agility** appears to be a more pragmatic short- to medium-term strategy. The ability to easily switch between certificate algorithms is more valuable at this stage than attempting to adopt hybrid certificates before full support is available in client browsers and applications.

7.4 Extension of the project

To further enhance and complete the experiment, the following avenues could be pursued:

- **Implementing hybrid authentication mechanisms** in compliance with IETF standards for hybrid certificates, ensuring alignment with emerging best practices in PQC.

- **Developing a web service client capable of supporting authentication using PQC-based certificates**, enabling end-to-end testing of post-quantum secure authentication workflows in real-world scenarios.

- **Investigating and tracking the evolution of hybrid certificate formats** and maintaining active technological watch over standardisation efforts. This is essential given the current uncertainty surrounding which approaches will ultimately be widely adopted across the industry.

- **Investigating hybridisation using two post-quantum algorithms**, as an alternative to the classical/PQC combination traditionally explored. This approach could provide enhanced long-term security in anticipation of a fully quantum-resistant ecosystem and full defence-in-depth protection.

8. Recommendations

8.1 Technical challenges

The experimentation phase has highlighted the complexity of cryptographic configurations. To build upon this work, several areas should be investigated:

- **Implementing crypto-agility governance and tools**, enabling systems to adapt rapidly to changes in cryptographic standards or threat models.

  - **Crypto-agility is not limited** to PQC as it is becoming essential for both traditional and PQC mechanisms to mitigate exploitation risks and meet new short-term certificate policies.
  - It is also already relevant for traditional encryption as it enhances automation and detection of vulnerabilities and incidents and represents a key step for migration to PQC.

- **Exploring hybrid and extended signature schemes**, including their integration into current standards, to enhance the security and longevity of **digital document signing** in a post-quantum context.

- **Verifying the enforcement of security policies through post-hoc analysis** of network activity at multiple points within the architecture, ensuring correct implementation across the system.

- **Conducting comprehensive performance assessment and benchmarking of PQC** implementations with various algorithms, hybrid combinations, and load conditions to evaluate the impact cryptographic configurations on network load. This establishes baseline metrics for computational overhead, latency impacts, resource consumption, and TLS performance including transactions per second (TPS), enabling informed decision-making for production deployment and optimisation strategies.

- **Continuing development and evaluation of hybrid certificate solutions**, with particular attention to identifying compatibility issues and implementation limitations across platforms.

8.2 Implementations

Following this experiment, several key recommendations have been identified to guide a secure and effective transition to PQC.

First and foremost, it is essential to identify the most critical applications and dataflows/ workflows, particularly those involving sensitive or high value information, to prioritise application layers, proxies, PKI and key management systems, ensuring PQC algorithm compatibility throughout.

Furthermore, PQC must be integrated by default and by design into future and currently developed system architectures and security models.

Finally, the success of this transition hinges on raising awareness among all internal stakeholders, both technical and non-technical, to ensure a shared understanding of the risks and implications associated with PQC threats and the mitigation measures to be implemented.

8.3 Organisation

Beyond the technical dimension, the transition to PQC requires a clear organisational framework. Organisations must not only adapt their infrastructures, but also strengthen governance, asset management, and coordination efforts. The following recommendations outline the key organisational enablers to ensure that PQC adoption is structured, auditable, and aligned with international standards

- **Establishing a PQC readiness programme** in partnership with industry stakeholders, technology vendors, and regulatory authorities to strengthen cryptographic practices. Coordinating with EFR, QSFF, and FS-ISAC frameworks for regulatory compliance.

- **Building centralised cryptographic asset management:** creating an automated, enterprise-wide system to track all cryptographic components including keys, certificates, libraries, and modules. This should also metadata management for crypto agility and audit compliance.

- **Leveraging modern cryptographic discovery tools** to analyse and monitor cryptographic usage patterns across the network.

- **Establishing a dedicated governance framework:** developing quantum-resistant cryptographic policies aligned with NIST standards and defining clear ownership and lifecycle responsibilities across DevSecOps, infrastructure, PKI, and third-party management.

- **Classifying all applications and appliances by cryptographic use case, and assessing them based on migration cost, complexity, and time-dependent risk.** For example, the current risk linked to authentication may be low today, but could increase to medium in five years, and high in ten years. In contrast, data-in-transit encryption (e.g., key exchange) may already present a medium risk, which is likely to become high within five years.

8.4 Operational roadmap and milestones

A phased migration to PQC cannot rely solely on ad-hoc initiatives: it **requires a structured roadmap supported by concrete milestones**. Organisations need to anticipate dependencies, prepare fallback options, and synchronise their timelines with international adoption programmes. The following actions provide a practical sequence for testing, prioritising, and progressively integrating PQC into operational environments

- Assess whether a current PKI provider is likely to support PQC certificate issuance in the near future.

  - If this constitutes a critical dependency in the migration path, consider implementing or using a secondary PKI at least for the most critical systems.
  - It may also be necessary to initiate early testing, even before the current provider becomes fully PQC compatible. Some open-source solutions, such as EJBCA and OpenSSL, already offer partial PQC support, making them suitable for early-stage testing.

- Identify the main uses of cryptography within an Information System (IS):

  - SSL exchange: assess key exchange (KEX/KEM) algorithms currently in use.
  - Server authentication: map TLS server certificate deployments.
  - Client authentication: identify client certificate-based identity verification systems.
  - Symmetric encryption: verify algorithm types (e.g., AES) and key lengths against current best practices.
  - Asymmetric encryption: list usage for archival, secure email, and backup purposes.
  - Key pairs: catalogue systems relying on SSH, PGP, or other asymmetric mechanisms.

- From this analysis, develop a structured migration plan and identify dependencies to determine the optimal order of migration.

- Upgrading critical infrastructure components, such as reverse proxies hosting VIPs, to support PQC KEMs can significantly improve TLS security, while requiring limited implementation effort.

- Establish dedicated network monitoring systems to ensure the correct and consistent implementation of PQC-safe security policies.

- Ensure post-quantum readiness and crypto agility are included as evaluation criteria in sourcing and procurement processes. Promote awareness and engagement among all internal stakeholders, both technical and non-technical, to ensure alignment across the organisation.

To this extend, ALV and BDF highly recommend following Europe's PQC adoption timeline:

- 2026 Implementation Threshold: organisations must commence quantum-resistant migration through national frameworks and foundational activities, including security evaluations, stakeholder education, and cryptographic asset cataloguing.

- 2030 Priority System Requirements: high-risk environments including utilities, telecommunications, finance, and government systems must achieve PQC implementation with mandatory completion no later than 2030.

- 2035 Comprehensive Migration Target: European-wide PQC deployment should reach maximum practical coverage across all systems. Legacy platforms may require extended periods but must achieve quantum-safe status to the fullest degree possible.

9. Conclusion

This experiment conducted jointly by the Banque de France and Allianz France constitutes a significant milestone in the transition to quantum-safe cryptographic infrastructures. **It demonstrates that post-quantum technologies, although still emerging, can already be integrated into existing systems** to secure critical communications, while preserving operational continuity and avoiding major architectural changes in the tested configuration.

Throughout this project, **the experiment confirms that hybrid post-quantum encryption schemes can be effectively deployed. Our implementation relied on standardised algorithms** (ML-KEM, ML-DSA, SLH-DSA), **while adopting encapsulation-based hybridisation principles** ensuring both backward compatibility and resilience against future quantum threats.

This approach enabled us to mitigate two major classes of threats: the interception and later decryption of confidential data ("store now, decrypt later"), and the future ability to impersonate clients or servers once quantum computers become powerful enough to break traditional cryptography.

The experiment highlighted two key insights:

- **From a technical standpoint**, it is already feasible to deploy quantum-safe protections, provided that key components (such as proxies, endpoints, or PKI) are compatible or properly isolated. The test set-up showed that post-quantum TLS tunnels can secure file transfers without modifying core applications.

- **From an operational standpoint**, however, current constraints linked to legacy infrastructure, lack of crypto-agility, and dependency of PQC support in standard browsers or proxies remain major elements to consider. Overcoming these hurdles will require both short-term workarounds and long-term redesigns of security architectures.

The experiment also led to the establishment of emergency response capabilities. If current encryption methods fail unexpectedly, both organisations can activate hybrid quantum-resistant systems within days. Deployment relies on minimal additional infrastructure, mainly limited to extra proxy servers.

The results of this experiment reinforce the notion that a successful PQC transition should be structured as a phased, long-term migration programme, combining immediate protections for the most sensitive data with progressive architectural updates.

Moreover, crypto-agility, early testing, and industry-wide standardisation will be essential to ensure secure-by-design infrastructures in the post-quantum era. Given the ecosystemic nature of financial systems, large-scale coordination with standardisation efforts led by financial institutions is required.

With algorithms now standardised, the ecosystem can confidently adopt these solutions. Our experiment proves implementation is achievable without excessive complexity, offering immediate operational benefits and clear action paths. Ongoing monitoring of PQC developments remains essential, particularly for current standardisation processes including NIST selection procedures and authentication protocols.

This work lays a concrete foundation for future efforts. By integrating PQC into live systems and building a reusable implementation pattern, it sets the stage for broader adoption and cross-sector collaboration.

This initiative demonstrates that a future-proof security transition is not only within reach, but is already underway. The Banque de France and Allianz stand ready to lead, together with the ecosystem, the next phase of Europe's quantum readiness.

10. Glossary

Traditional cryptography: Cryptographic algorithms and protocols established before the quantum threat, such as RSA and ECC, relying on classical computational hardness assumptions.

1. Cryptographic primitive: A low-level cryptographic algorithm (e.g., hash function, encryption scheme, signature) that serves as a building block for higher-level cryptographic protocols and systems.

2. Encryption: The process of converting plaintext data into ciphertext to prevent unauthorised access, ensuring confidentiality.

3. Asymmetric encryption: Encryption using a public/private key pair, enabling secure communication without prior key exchange.

4. Public Key Encryption (PKE): A cryptographic scheme allowing direct encryption of data using a public key, with decryption by the private key.

5. Key Exchange (KEX): A mechanism that allows two parties to securely establish a shared secret over an insecure channel.

6. Digital Signature Algorithm (DSA): A family of algorithms used to verify the authenticity and integrity of digital messages.

7. RSA: A widely used public-key cryptographic algorithm based on the mathematical difficulty of factoring large integers, used for encryption and digital signatures.

8. Elliptic Curve Cryptography (ECC): A form of public-key cryptography relying on the algebraic structure of elliptic curves, providing strong security with smaller key sizes.

9. Elliptic Curve Digital Signature Algorithm (ECDSA): A variant of DSA based on elliptic curve cryptography, enabling highly compact digital signatures.

10. Stream: A continuous flow of data sent over a network, typically secured and structured, often in the context of real-time or persistent communication.

11. Connect: The establishment of a secure or authenticated session between systems, users, or applications within an operational workflow.

12. Stunnel: An open-source proxy designed to add TLS encryption to existing clients and servers without changes to the application code. It is commonly used to secure legacy systems or automate tunnelling.

13. TPS (Transactions Per Second): A common performance metric that measures how many individual transactions a system can process every second. In cryptographic systems, TPS is influenced by the algorithm's computational cost and network efficiency.

PQC

14. Post Quantum Cryptography (PQC): A field of cryptography focused on developing algorithms resistant to attacks from quantum computers.

15. Hybridisation: The process of combining classical and post-quantum algorithms to ensure both backward compatibility and forward security.

16. Encapsulation: A technique used in KEMs where a symmetric key is securely transmitted within a ciphertext to establish a secure session.

17. Crypto-agility: The ability of a cryptographic system to quickly adopt new algorithms or configurations without significant redesign, facilitating resilience and adaptability.

18. Key Encapsulation Mechanism (KEM): Unlike public-key encryption schemes, KEMs are not designed to encrypt messages directly, but rather to establish shared keys that are then used in symmetric encryption protocols.

19. ML-KEM (FIPS 203): A NIST-standardised post-quantum key encapsulation mechanism derived from CRYSTALS-Kyber, used for secure key exchange.

20. ML-DSA (FIPS 204): A post-quantum digital signature algorithm standardised by NIST and based on CRYSTALS-Dilithium, ensuring message authenticity and integrity.

21. SLH-DSA (FIPS 205): A post-quantum signature scheme based on hash functions, specifically SPHINCS+, providing a stateless, lattice-free alternative.

22. FrodoKEM: A Key Encapsulation Mechanism based on standard lattice problems, developed as a conservative post-quantum alternative. Unlike ML-KEM, FrodoKEM does not rely on structured lattices, making it attractive for certain high-assurance environments despite larger key sizes.

23. Falcon: A NIST-approved post-quantum digital signature algorithm based on lattice cryptography and NTRU lattices. It offers compact signatures and high efficiency, making it suitable for constrained environments.

24. SPHINCS+: A stateless hash-based post-quantum signature scheme standardised by NIST. It provides strong long-term security assumptions but comes with larger signature sizes and slower signing operations compared to lattice-based alternatives.

Protocols and Standards

25. Client-server communications: communication between a user and a service provider.

26. Transport Layer Security (TLS): A cryptographic protocol that provides secure communication over the internet through encryption and authentication.

27. TLS communications: Secure data exchanges conducted using the TLS protocol, ensuring the confidentiality and integrity of transmitted information.

28. Public Key Infrastructure (PKI): A framework of policies, hardware, software, and procedures used to manage digital certificates and public-key encryption.

29. IPSec VPN tunnel: A virtual private network tunnel secured using the IPsec protocol suite, providing confidentiality and integrity at the network layer.

30. X25519: A Diffie-Hellman function over Curve25519 used in key exchange, offering high performance and strong security in TLS and VPNs.

31. OpenSSL: A widely adopted open-source library implementing SSL/TLS protocols and cryptographic functions.

32. EJBCA: An open-source Certificate Authority (CA) software for managing PKI infrastructure and issuing digital certificates.

33. C-QSL (Crypto-Qualified Security Libraries): A European framework led by ENISA to assess and qualify cryptographic libraries according to post-quantum and classical security requirements. It supports the development of trusted cryptographic implementations.

34. NIST: The US National Institute of Standards and Technology, responsible for developing and standardising cryptographic algorithms, including PQC.

35. ANSSI: The French Cybersecurity Agency, providing guidance and certification on cybersecurity and cryptographic standards.

Infrastructure and network architecture

36. Virtual Private Network (VPN): A secure communication channel established over a public network to protect data and identities.

37. Forward proxy: A server that sits between a client and the internet, often used for filtering, monitoring, or anonymising client traffic.

38. Reverse proxy: A server that handles incoming internet traffic and routes it to internal servers, often used for load balancing and security.

39. Hardware Security Module (HSM): A dedicated, tamper-resistant hardware device designed to securely generate, store, and manage cryptographic keys and perform sensitive cryptographic operations such as signing or encryption.

40. Virtual IP Load Balancer (VIP LB): A networking component that distributes traffic to multiple backend servers using a shared virtual IP address.

41. NGINX: A high-performance web server and reverse proxy used for load balancing, TLS termination, and application acceleration.

42. F5 Load Balancer: A commercial hardware/software appliance that distributes network traffic across servers and provides advanced security and optimisation features.

43. Identity and Access Management (IAM): A set of tools and policies that manages user identities and control access to systems and data.

44. SQUID proxy: A widely used open-source caching and forwarding web proxy server.

Cybersecurity

45. Store Now, Decrypt Later (SNDL)/Harvest Now, Decrypt Later (HNDL): Attack strategies where encrypted data is collected now and stored for future decryption, assuming advances in cryptanalysis or quantum computing.

46. Denial of Service (DoS) attack: A malicious attempt to disrupt the normal traffic of a service by overwhelming it with a flood of traffic or requests.

Experimentation and methodology

47. Proof of Concept (PoC): A preliminary implementation of a concept or technology to demonstrate its feasibility and validate core assumptions.

48. Application-to-Application (A2A): Communication between software systems without human intervention, often used in automated processes.

49. User-to-Application (U2A): Interaction where a human user communicates directly with a software system via a user interface.

11. Appendices

Appendices A.
Detailed description of the current components of the use case on the server side, and client side, and non-updatable components

a. Server side

- TLS endpoint appliance (F5)
  - VIP F5 IAM: Virtual IP (VIP) address managed by an F5 Load Balancer (LB) that distributes the network traffic across multiple IAM instances to ensure availability and performance.
  - VIP F5 Onegate: A VIP managed by an F5 LB that distributes the network traffic across multiple instances of OneGate

- IAM: Identity and Access Management framework to verify user identities and control the access to the network resources and services. In this experiment, the IAM is a custom development.

- Webserver application (Onegate Wildfly): An internal application developed by the bank for regulatory file deposit.

- PKI: Public Key Infrastructure deliver server certificates

b. Client side

- Outgoing proxy: An enterprise proxy based on an Open Source project (SQUID)

- Browser: In this experiment, the chosen browsers were Google Chrome, version 136.0.7103.25 and Firefox. Both browsers support PQC protocols and NIST Standards.

- Client application: webservice application sending regulatory reporting obligations to OneGate.

- PKI: Public Key Infrastructure deliver client certificates

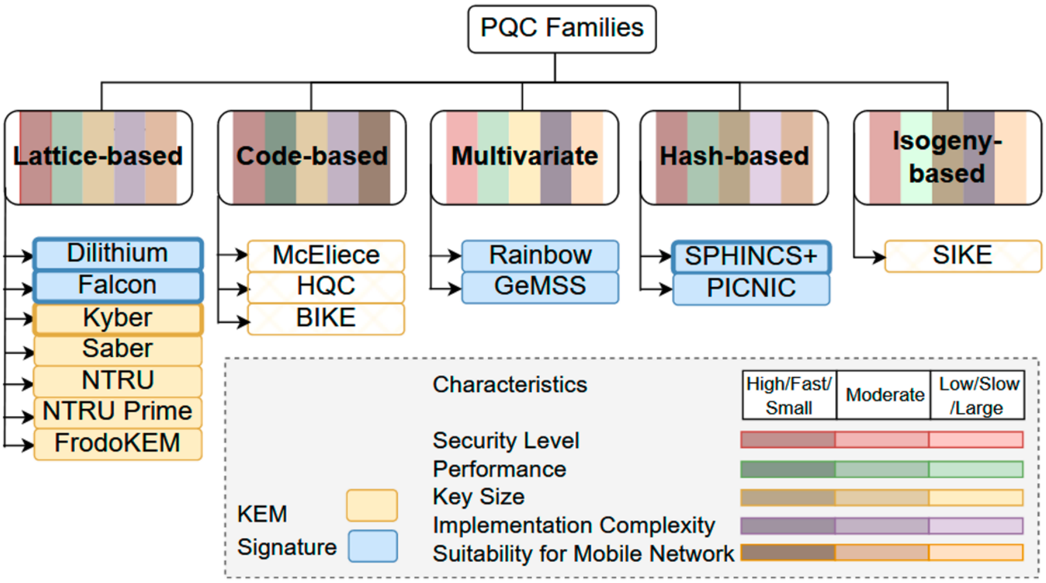
c. Non-updatable components

For precision, it is worth mentioning the fixed and immutable components in the existing infrastructure. This may be due to development cost or administrative constraints, out of this experiment scope. The following components were considered immutable during the experiment:

- Server side
  - TLS endpoint appliance (F5)
  - Webserver application (Onegate Wildfly)
  - IAM (authenfication)
- Client side
  - Outgoing traditional proxy
  - Browsers
  - Client application

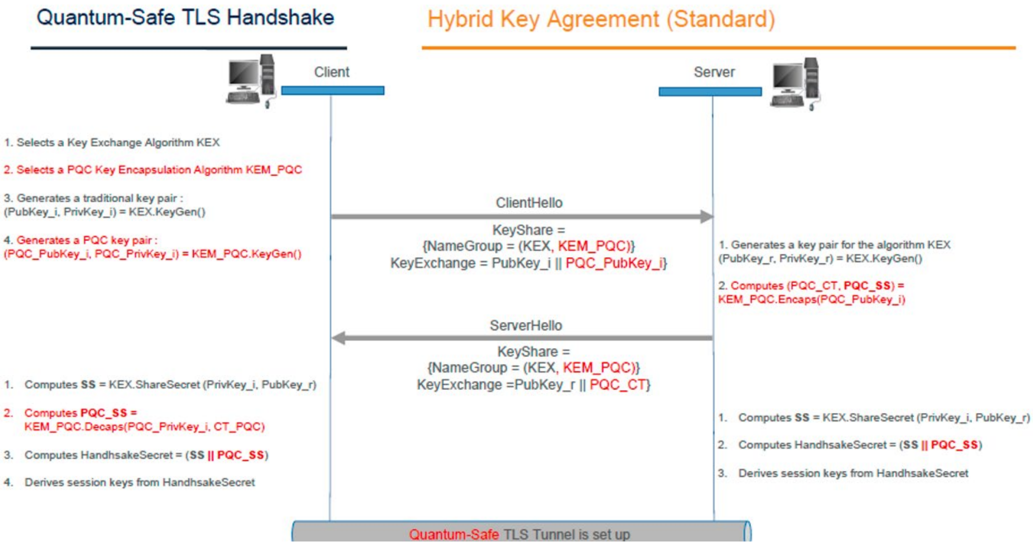
Appendices B.
Algorithms under review for standardisation.

Those candidates can be classified by families, each of those is based on mathematical challenges that are difficult to solve. The algorithms outlined in bold are the four candidates for standardisation, and the boxes with a hashed pattern are the candidates for the KEMs algorithms that will advance to the round 4".



Exploring PQC with Quantum Key Distribution for Sustainable Mobile Network Architecture Design
Source: *http://arxiv.org/abs/2404.10602*

Appendices C.
TLS protocol Handshake



1. Selects a Key Exchange Algorithm KEX
2. Selects a PQC Key Encapsulation Algorithm KEM_PQC
3. Generates a traditional key pair : (PubKey_i, PrivKey_i) = KEX.KeyGen()
4. Generates a PQC key pair : (PQC_PubKey_i, PQC_PrivKey_i) = KEM_PQC.KeyGen()

ClientHello

KeyShare = {NameGroup = (KEX, KEM_PQC)} KeyExchange = PubKey_i || PQC_PubKey_i}

1. Generates a key pair for the algorithm KEX (PubKey_r, PrivKey_r) = KEX.KeyGen()
2. Computes (PQC_CT, **PQC_SS**) = KEM_PQC.Encaps(PQC_PubKey_i)

ServerHello

KeyShare = {NameGroup = (KEX, KEM_PQC)} KeyExchange =PubKey_r || PQC_CT}

1. Computes **SS** = KEX.ShareSecret (PrivKey_i, PubKey_r)
2. Computes **PQC_SS** = KEM_PQC.Decaps(PQC_PrivKey_i, CT_PQC)
3. Computes HandhsakeSecret = (**SS || PQC_SS**)
4. Derives session keys from HandhsakeSecret

1. Computes **SS** = KEX.ShareSecret (PrivKey_i, PubKey_r)
2. Computes HandhsakeSecret = (**SS || PQC_SS**)
3. Derives session keys from HandhsakeSecret

Quantum-Safe TLS Tunnel is set up

Appendices D.
Detailed technical contributions by CryptoNext Securities

- CryptoNext's C-QSL library in which the NIST (ML-KEM, ML-DSA, SLH-DSA, Falcon) or ISO (FrodoKEM) algorithms are implemented.

- Security providers in C (openSSL provider) or Java (compatible with the Java security standard). These components securely implement the post-quantum TLS 1.3 application layer and certificate management, as dynamic software libraries (.so file) that can be used by other programs.

- The software used in the experiments was not directly modified; instead, it systematically relied on CryptoNext Security's dynamic libraries. Hence the tools that we have used as TLS-proxys are NGINX, Apache and sTunnel. They do not require any modifications; only the configuration files of these software are modified to indicate the transition from traditional cryptography to post-quantum (hybrid) crypto.

- The EJBCA CE PKI is an application accessible via a web page written natively in java and which relies on providers for cryptographic security aspects: Bouncy Castle (for the Crypto Software part) and PKCS11 for access to an HSM. We have therefore replaced these providers with CryptoNext post-quantum versions of these providers allowing the transition of the EJBCA CE application. (see appendices F for a summary table of the state of the art on PKI interoperability).

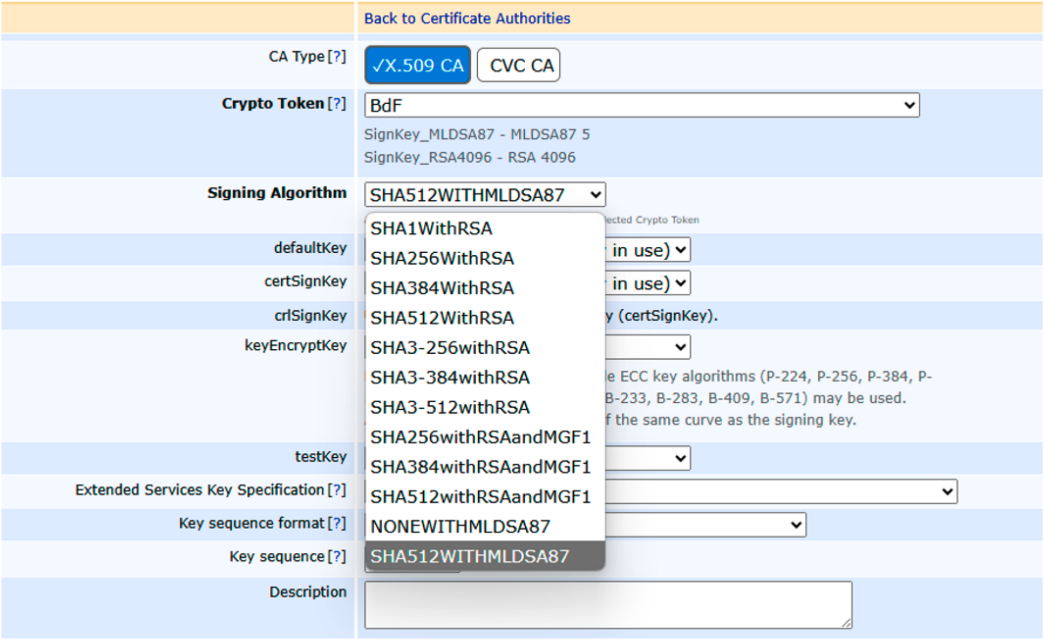
- CryptoNext also made a major change to the SoftHSM software, which simulates a Hardware Security Module (HSM). The modifications mainly involved updating the PKCS#11 protocol to version 3.2 supporting PQC and incorporating CryptoNext's low-level C-QSL library supporting post-quantum algorithms.

Appendices E.
Remediation matrix by components

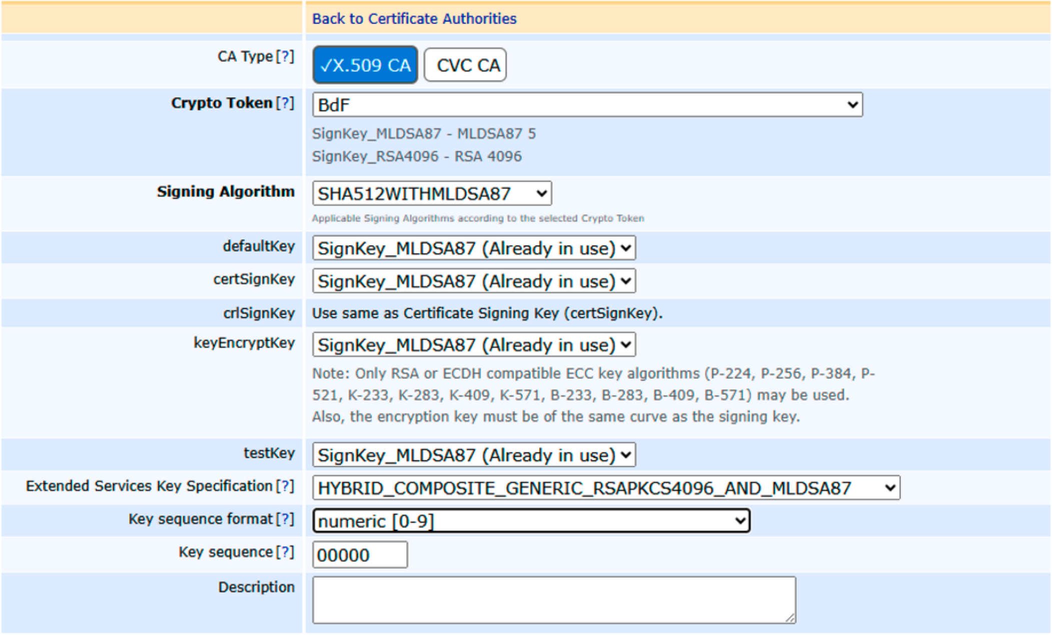
Component	Problem	Remediation
Enterprise proxy (Squid)	Only supports CONNECT mode; incompatible with standard NGINX frontend.	Use a stunnel-based forward proxy (FP).
Apache server	Only supports encapsulating HTTP, not HTTPS.	Use a reverse proxy with NGINX.
VIP Onegate	Cannot handle PQC KEM.	Use a reverse proxy (HTTP or Stream) that performs PQC KEM downstream and traditional KEX upstream.
VIP IAM (PQC KEM)	Cannot handle PQC KEM.	Use a reverse proxy (HTTP or Stream) that performs PQC KEM downstream and traditional KEX upstream.
VIP IAM (RSA header auth)	Does not trust client RSA certificates passed via HTTP headers (test environment limitation).	Use an HTTP reverse proxy to map the downstream RSA certificate to an upstream RSA certificate. Use a Stream reverse proxy to forward the certificate to VIP IAM.
VIP IAM (client PQC cert)	Cannot handle client PQC authentication certificates.	Use a Stream reverse proxy to verify PQC client authentication at the FP and forward the certificate. Use an HTTP reverse proxy to map the PQC cert to an RSA cert upstream.
Java client (PQC auth)	Cannot handle PQC server authentication.	Use a Stream frontend (STUNNEL) to manage PQC server authentication. Client continues to process RSA server certs for VIP IAM and Onegate.
Java client (PQC KEM)	Cannot handle PQC KEM.	Use a Stream frontend (STUNNEL) to manage PQC KEM. Client continues to use traditional KEX with VIP IAM and Onegate.
Browser client (PQC auth)	Cannot handle PQC server authentication.	Use a Stream frontend (STUNNEL) for PQC server authentication. Browser still handles RSA server authentication for VIP IAM and Onegate. Note: KEX inside the tunnel remains legacy.
Browser client (PQC KEM)	Can handle PQC KEM.	The HTTP reverse proxy setup meets this requirement.
Network routing	Uses ad-hoc servers and ports, not identified via URLs/domains.	Hardcoded IP/URL matching and port redirection in Allianz configuration files.

Appendices F.
PKI Interoperability

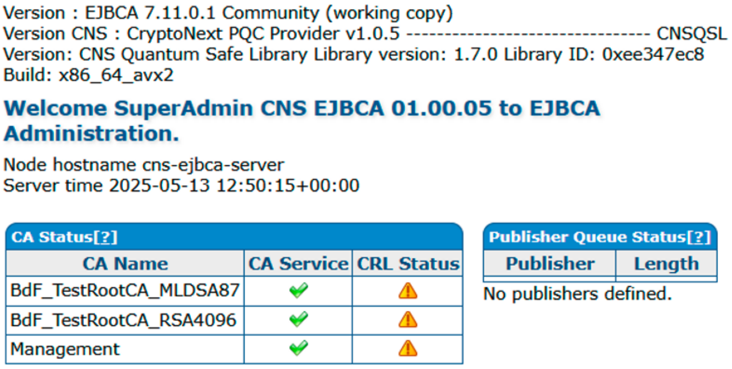
	EJBCA Algorithm	OID Key	Supported by EJBCA CNS	Supported by EJBCA CE	Interop
ML-DSA	NONEWITHMLDSA44	2.16.840.1.101.3.4.3.17	YES	YES	YES
	NONEWITHMLDSA65	2.16.840.1.101.3.4.3.18	YES	YES	YES
	NONEWITHMLDSA87	2.16.840.1.101.3.4.3.19	YES	YES	YES
SLH-DSA	NONEWITHSLHDSA_SHA2_128F	2.16.840.1.101.3.4.3.21	YES	NO	NO
	NONEWITHSLHDSA_SHA2_192F	2.16.840.1.101.3.4.3.23	YES	NO	NO
	NONEWITHSLHDSA_SHA2_256F	2.16.840.1.101.3.4.3.25	YES	NO	NO
	NONEWITHSLHDSA_SHA2_128S	2.16.840.1.101.3.4.3.20	YES	NO	NO
	NONEWITHSLHDSA_SHA2_192S	2.16.840.1.101.3.4.3.22	YES	NO	NO
	NONEWITHSLHDSA_SHA2_256S	2.16.840.1.101.3.4.3.24	YES	NO	NO
	NONEWITHSLHDSA_SHAKE_128F	2.16.840.1.101.3.4.3.27	YES	NO	NO
	NONEWITHSLHDSA_SHAKE_192F	2.16.840.1.101.3.4.3.29	YES	NO	NO
	NONEWITHSLHDSA_SHAKE_256F	2.16.840.1.101.3.4.3.31	YES	NO	NO
	NONEWITHSLHDSA_SHAKE_128S	2.16.840.1.101.3.4.3.26	YES	NO	NO
	NONEWITHSLHDSA_SHAKE_192S	2.16.840.1.101.3.4.3.28	YES	NO	NO
	NONEWITHSLHDSA_SHAKE_256S	2.16.840.1.101.3.4.3.30	YES	NO	NO
Dilithium	NONEWITHDILITHIUM2	1.3.6.1.4.1.2.267.7.4.4	YES	NO	NO
	NONEWITHDILITHIUM3	1.3.6.1.4.1.2.267.7.6.5	YES	NO	NO
	NONEWITHDILITHIUM5	1.3.6.1.4.1.2.267.7.8.7	YES	NO	NO
Falcon	NONEWITHFALCON512	1.3.9999.3.6	YES	YES	YES
	NONEWITHFALCON1024	1.3.9999.3.9	YES	YES	YES
Sphincs	NONEWITHSPHINCS_SHAKE_SIMPLE_128F	1.3.9999.6.7.4	YES	NO	NO
	NONEWITHSPHINCS_SHAKE_SIMPLE_192F	1.3.9999.6.8.3	YES	NO	NO
	NONEWITHSPHINCS_SHAKE_SIMPLE_256F	1.3.9999.6.9.3	YES	NO	NO
HYBRID_CATALYST_RSA	ML-DSA-44 +RSA3072	1.2.840.113549.1.1.1 (= OID RSA)	YES	YES (CSR only)	YES
HYBRID_CATALYST_ECDSA	ML-DSA-87 +ECDSA_P521	1.2.840.10045.2.1 (= OID ECDSA)	YES	YES (CSR only)	YES
HYBRID_COMPOSITE_EXPLICIT_RSA	ML-DSA-44 +RSA3072	1.3.9999.8888.0.1.256.123	YES	NO	NO
HYBRID_COMPOSITE_EXPLICIT_ECDSA	ML-DSA-87 +ECDSA_P521	1.3.9999.8888.0.1.258.148	YES	NO	NO



	Back to Certificate Authorities
CA Type [?]	✓X.509 CA CVC CA
Crypto Token [?]	BdF SignKey_MLDSA87 - MLDSA87 5 SignKey_RSA4096 - RSA 4096
Signing Algorithm	SHA512WITHMLDSA87 SHA1WithRSA SHA256WithRSA SHA384WithRSA SHA512WithRSA SHA3-256withRSA SHA3-384withRSA SHA3-512withRSA SHA256withRSAandMGF1 SHA384withRSAandMGF1 SHA512withRSAandMGF1 NONEWITHMLDSA87 SHA512WITHMLDSA87 lected Crypto Token
defaultKey	in use)
certSignKey	in use)
crlSignKey	y (certSignKey).
keyEncryptKey	le ECC key algorithms (P-224, P-256, P-384, P- B-233, B-283, B-409, B-571) may be used. f the same curve as the signing key.
testKey	
Extended Services Key Specification [?]	
Key sequence format [?]	
Key sequence [?]	
Description	



	Back to Certificate Authorities
CA Type [?]	✓X.509 CA CVC CA
Crypto Token [?]	BdF SignKey_MLDSA87 - MLDSA87 5 SignKey_RSA4096 - RSA 4096
Signing Algorithm	SHA512WITHMLDSA87 Applicable Signing Algorithms according to the selected Crypto Token
defaultKey	SignKey_MLDSA87 (Already in use)
certSignKey	SignKey_MLDSA87 (Already in use)
crlSignKey	Use same as Certificate Signing Key (certSignKey).
keyEncryptKey	SignKey_MLDSA87 (Already in use) Note: Only RSA or ECDH compatible ECC key algorithms (P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409, B-571) may be used. Also, the encryption key must be of the same curve as the signing key.
testKey	SignKey_MLDSA87 (Already in use)
Extended Services Key Specification [?]	HYBRID_COMPOSITE_GENERIC_RSAPKCS4096_AND_MLDSA87
Key sequence format [?]	numeric [0-9]
Key sequence [?]	00000
Description	



Version : EJBCA 7.11.0.1 Community (working copy)
Version CNS : CryptoNext PQC Provider v1.0.5 ------------------------------- CNSQSL
Version: CNS Quantum Safe Library Library version: 1.7.0 Library ID: 0xee347ec8
Build: x86_64_avx2

Welcome SuperAdmin CNS EJBCA 01.00.05 to EJBCA Administration.

Node hostname cns-ejbca-server
Server time 2025-05-13 12:50:15+00:00

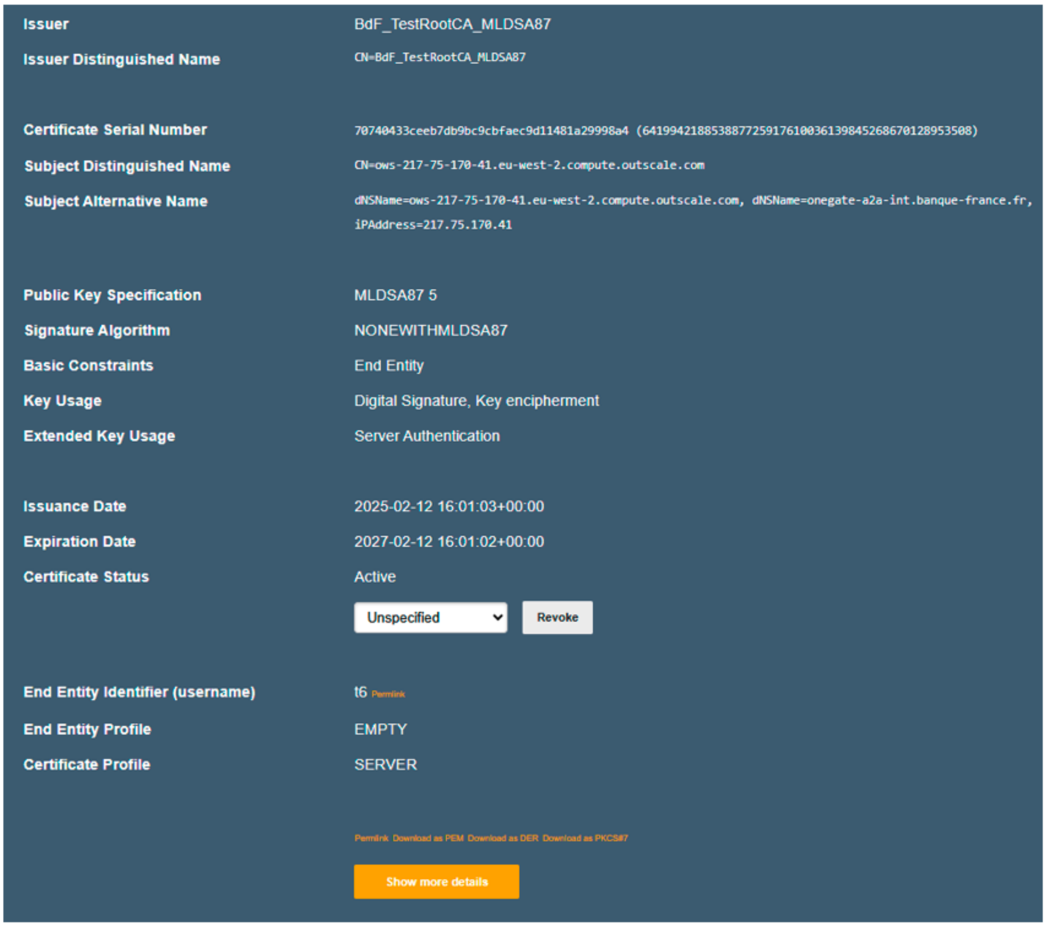
CA Status[?]

CA Name	CA Service	CRL Status
BdF_TestRootCA_MLDSA87	✓	⚠
BdF_TestRootCA_RSA4096	✓	⚠
Management	✓	⚠

Publisher Queue Status[?]

Publisher	Length

No publishers defined.



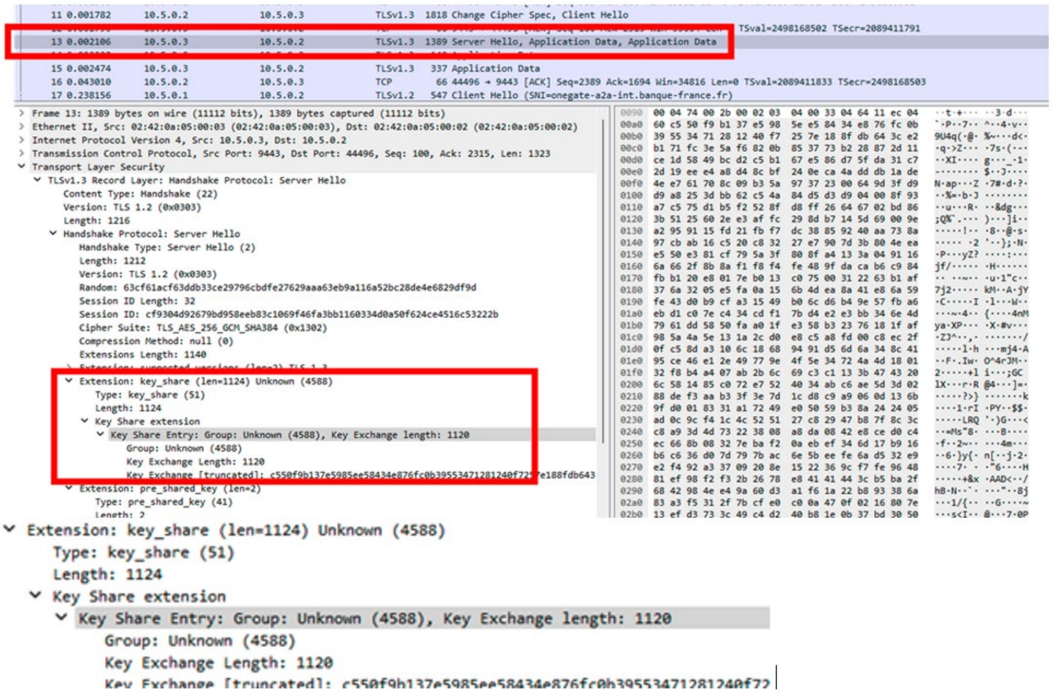
Issuer	BdF_TestRootCA_MLDSA87
Issuer Distinguished Name	CN=BdF_TestRootCA_MLDSA87
Certificate Serial Number	70740433ceeb7db9bc9cbfaec9d11481a29998a4 (641994218853887725917610036139845268670128953508)
Subject Distinguished Name	CN=ows-217-75-170-41.eu-west-2.compute.outscale.com
Subject Alternative Name	dNSName=ows-217-75-170-41.eu-west-2.compute.outscale.com, dNSName=onegate-a2a-int.banque-france.fr, iPAddress=217.75.170.41
Public Key Specification	MLDSA87 5
Signature Algorithm	NONEWITHMLDSA87
Basic Constraints	End Entity
Key Usage	Digital Signature, Key encipherment
Extended Key Usage	Server Authentication
Issuance Date	2025-02-12 16:01:03+00:00
Expiration Date	2027-02-12 16:01:02+00:00
Certificate Status	Active
	Unspecified ⌄ Revoke
End Entity Identifier (username)	t6 Permlink
End Entity Profile	EMPTY
Certificate Profile	SERVER

Notes:

- We used the mode None and not SHA because Post-quantum signature can be used to sign a message of any length; hence there is no need to compute a hash of the message before computing the signature. The meaning of NONEWITHMLDS87 is that we sign the certificate directly with ML-DSA at level 5.
- We did not use hybrid mode in our implementation. However, if you generate a hybrid Certificate Authority (CA), the resulting CA certificate will contain two public keys: a traditional one and an ML-DSA key. Each of these keys will be used to sign the corresponding part of the hybrid key in leaf certificates – the traditional key signs the classical component, and the ML-DSA key signs the post-quantum component.

Appendices G.
Illustration of the difficulty to verify whether a PQC KEM algorithm is being used

With TLS 1.3, most of the handshake, including the cipher suite negotiation, is encrypted, making it impossible to determine the specific algorithm used from packet captures alone. However, it is still possible to identify the key exchange algorithm from the unencrypted part of the handshake. For example, the codepoint 4588 indicates the use of the hybrid key exchange algorithm x25519-mlkem768.



```
   11 0.001782    10.5.0.2     10.5.0.3     TLSv1.3  1818 Change Cipher Spec, Client Hello
                                                                                 TSval=2498168502 TSecr=2089411791
   13 0.002106    10.5.0.3     10.5.0.2     TLSv1.3  1389 Server Hello, Application Data, Application Data

   15 0.002474    10.5.0.3     10.5.0.2     TLSv1.3   337 Application Data
   16 0.043010    10.5.0.2     10.5.0.3     TCP        66 44496 → 9443 [ACK] Seq=2389 Ack=1694 Win=34816 Len=0 TSval=2089411833 TSecr=2498168503
   17 0.238156    10.5.0.1     10.5.0.2     TLSv1.2   547 Client Hello (SNI=onegate-a2a-int.banque-france.fr)
```

```
> Frame 13: 1389 bytes on wire (11112 bits), 1389 bytes captured (11112 bits)
> Ethernet II, Src: 02:42:0a:05:00:03 (02:42:0a:05:00:03), Dst: 02:42:0a:05:00:02 (02:42:0a:05:00:02)
> Internet Protocol Version 4, Src: 10.5.0.3, Dst: 10.5.0.2
> Transmission Control Protocol, Src Port: 9443, Dst Port: 44496, Seq: 100, Ack: 2315, Len: 1323
v Transport Layer Security
  v TLSv1.3 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 1216
    v Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 1212
        Version: TLS 1.2 (0x0303)
        Random: 63cf61acf63ddb33ce29796cbdfe27629aaa63eb9a116a52bc28de4e6829df9d
        Session ID Length: 32
        Session ID: cf9304d92679bd958eeb83c1069f46fa3bb1160334d0a50f624ce4516c53222b
        Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
        Compression Method: null (0)
        Extensions Length: 1140
      > Extension: supported_versions (len=2) TLS 1.3
      v Extension: key_share (len=1124) Unknown (4588)
          Type: key_share (51)
          Length: 1124
        v Key Share extension
          v Key Share Entry: Group: Unknown (4588), Key Exchange length: 1120
              Group: Unknown (4588)
              Key Exchange Length: 1120
              Key Exchange [truncated]: c550f9b137e5985ee58434e876fc0b39553471281240f72 e188fdb643
      v Extension: pre_shared_key (len=2)
          Type: pre_shared_key (41)
          Length: 2
```

```
0090  00 04 74 00 2b 00 02 03  04 00 33 04 64 11 ec 04   ··t·+···  ··3·d···
00a0  60 c5 50 f9 b1 37 e5 98  5e e5 84 34 e8 76 fc 0b   `·P··7··  ^··4·v··
00b0  39 55 34 71 28 12 40 f7  25 7e 18 8f db 64 3c e2   9U4q(·@·  %~···d<·
00c0  b1 71 fc 3e 5a f6 82 0b  85 37 73 b2 28 87 2d 11   ·q·>Z···  ·7s·(·-·
00d0  ce 1d 58 49 bc d2 c5 b1  67 e5 86 d7 5f da 31 c7   ··XI····  g···_·1·
00e0  2d 19 ee e4 a8 d4 8c bf  24 0e ca 4a dd db 1a de   -·······  $··J····
00f0  4e e7 61 70 8c 09 b3 5a  97 37 23 00 64 9d 3f d9   N·ap···Z  ·7#·d·?·
0100  d9 a8 25 3d bb 62 c5 4a  84 d5 d3 d9 04 00 8f 93   ··%=·b·J  ········
0110  a7 c5 75 d1 b5 f2 52 8f  d8 ff 26 64 67 02 bd 86   ··u···R·  ··&dg···
0120  3b 51 25 60 2e e3 af fc  29 8d b7 14 5d 69 00 9e   ;Q%`.···  )···]i··
0130  a2 95 91 15 fd 21 fb f7  dc 38 85 92 40 aa 73 8a   ·····!··  ·8··@·s·
0140  97 cb ab 16 c5 20 c8 32  27 e7 90 7d 3b 80 4e ea   ····· ·2  '··};·N·
0150  e5 50 e3 81 cf 79 5a 3f  80 8f a4 13 3a 04 91 16   ·P···yZ?  ····:···
0160  6a 66 2f 8b 8a f1 f8 f4  fe 48 9f da ca b6 c9 84   jf/·····  ·H······
0170  fb b1 20 e8 01 7e b0 13  c0 75 00 31 22 63 b1 af   ·· ··~··  ·u·1"c··
0180  37 6a 32 05 e5 fa 0a 15  6b 4d ea 8a 41 e8 6a 59   7j2·····  kM··A·jY
0190  fe 43 d0 b9 cf a3 15 49  b0 6c d6 b4 9e 57 fb a6   ·C·····I  ·l···W··
01a0  eb d1 c0 7e c4 34 cd f1  7b d4 e2 e3 bb 34 6e 4d   ···~·4··  {····4nM
01b0  79 61 dd 58 50 fa a0 1f  e3 58 b3 23 76 18 1f af   ya·XP···  ·X·#v···
01c0  98 5a 4a 5e 13 1a 2c d0  e8 c5 a8 fd 00 c8 ec 2f   ·ZJ^··,·  ·······/
01d0  0f c5 8d a3 10 6c 18 68  94 91 d5 6d 6a 34 8c 41   ·····l·h  ···mj4·A
01e0  95 ce 46 e1 2e 49 77 9e  4f 5e 34 72 4a 4d 18 01   ··F·.Iw·  O^4rJM··
01f0  32 f8 b4 a4 07 ab 2b 6c  69 c3 c1 13 3b 47 43 20   2·····+l  i···;GC
0200  6c 58 14 85 c0 72 e7 52  40 34 ab c6 ae 5d 3d 02   lX···r·R  @4···]=·
0210  88 de f3 aa b3 3f 3e 7d  1c d8 c9 a9 06 0d 13 6b   ·····?>}  ·······k
0220  9f d0 01 83 31 a1 72 49  e0 50 59 b3 8a 24 24 05   ····1·rI  ·PY··$$·
0230  ad 0c 9c f4 1c 4c 52 51  27 c8 29 47 b8 7f 8c 3c   ·····LRQ  '·)G···<
0240  c8 a9 3d 4d 73 22 38 08  a8 da 08 42 e8 ce d0 c4   ··=Ms"8·  ···B····
0250  ec 66 8b 08 32 7e ba f2  0a eb ef 34 6d 17 b9 16   ·f··2~··  ···4m···
0260  b6 c6 36 d0 7d 79 7b ac  6e 5b ee fe 6a d5 32 e9   ··6·}y{·  n[··j·2·
0270  e2 f4 92 a3 37 09 20 8e  15 22 36 9c f7 fe 96 48   ····7· ·  ·"6····H
0280  81 ef 98 f2 f3 2b 26 78  e8 41 41 44 3c b5 ba 2f   ·····+&x  ·AAD<··/
0290  68 42 98 4e e4 9a 60 d3  a1 f6 1a 22 b8 93 38 6a   hB·N··`·  ···"··8j
02a0  83 a3 f5 31 2f 7b cf e0  c0 0a 47 0f 02 16 80 7e   ···1/{··  ··G····~
02b0  13 ef d3 73 3c 49 c4 d2  40 b8 1e 0b 37 bd 30 50   ···s<I··  @···7·0P
```

```
v Extension: key_share (len=1124) Unknown (4588)
    Type: key_share (51)
    Length: 1124
  v Key Share extension
    v Key Share Entry: Group: Unknown (4588), Key Exchange length: 1120
        Group: Unknown (4588)
        Key Exchange Length: 1120
        Key Exchange [truncated]: c550f9b137e5985ee58434e876fc0b39553471281240f72
```

Appendices H.
Detailed assessment of the experimentation

The **primary objective**, which was to secure TLS access using PQC (PQC) for a PQC compatible web browser (ML_KEM), was achieved by implementing a reverse proxy in front of the application in BDF's infrastructure.

However, the client standard certificate could not be forwarded to BDF's IAM system, which forced authentication to be offloaded to the proxy. As a result, the reverse proxy had to hold an authentication standard certificate for the client. This limitation could likely have been avoided if the IAM trusted the certificate sent in the HTTP headers by the proxy.

The advantage of this solution is that the PQC KEM component is handled entirely by the proxy, which avoids any changes to network VIPs and does not increase the load on the network equipment for post-quantum cryptographic processing.

The **second objective** was to secure TLS communications using PQC for applications. To achieve this, a different architecture was adopted: since both the client application and the server needed to support PQC, a forward proxy was interconnected on the client side with a reverse proxy on the server side. This allowed to establish a hybrid PQC tunnel through which standard TLS traffic could flow. Additionally, It enabled server authentication using PQC-based server certificates by the client proxy. This approach being transparent to both the application and the server, significantly simplifies the migration process to a PQC-compatible environment. A detailed comparison of the main post-quantum communication integration methods and their respective technical implications is presented in Table 3 – Post-Quantum Communication Integration Methods technical pros and cons (see appendices)

The **third objective** was to enable client authentication using **ML-DSA client certificates**, validated by the reverse proxy. This added a layer of post-quantum authentication on the client side, reinforcing the end-to-end security posture. By leveraging the proxy for certificate validation, it ensured compatibility and flexibility without requiring the application or backend systems to natively support ML-DSA.

a. Post-Quantum Communication Integration Methods technical pros and cons

Method	Pros	Cons
Reverse proxy (HTTP)	• Transparent for both application and client • Can be set up in front of most applications (via client certificate forwarding) • Often already exists (e.g., VIP Load Balancer) • No client-side component required (except the browser) • Once client applications support PQC certificates, full protection (KEM + Authentication) is achieved • Reverse proxy is easier to scale in environments already using Load Balancers or service meshes	• Client must support Hybrid PQC KEX/KEM to ensure quantum-safe communication • Current browsers/applications do not support PQC certificate-based authentication • IAM must support PQC client certificates for authentication • Does not enforce PQC-based authentication
Reverse proxy (stream / double tunnel)	• Simple client configuration (just a proxy) • Enforces PQC-based authentication (client and server sides) • Provides KEM protection from forward Proxy (FP) to Reverse Proxy (RP) • Offers better crypto agility (e.g., Chrome lacks support for some KEMs like FrodoKEM)	• Requires client-side component • Must be set up between two entities (e.g., company to company) • TLS connection between the client and the front proxy may still use classical crypto, even if the inner tunnel is PQC-protected.
VPN	• Transparent for application and user • VPN may offer better performance for long sessions (due to long-lived connections) • Provides full external protection as the VPN tunnel protects the entire flow from end-to-end, including mutual authentication	• Requires client-side component • VPNs may require more monitoring, key lifecycle management, etc. • Specific ports need to be opened • Must be set up between two entities (e.g., company to company) • No PQC protection upstream/downstream of the VPN tunnel

b. Post-Quantum Communication Integration Methods Evaluation Dashboard

Criterion	Reverse Proxy (HTTP)	Reverse Proxy (Stream)	VPN
PQC Standard Maturity	Early adoption, browser/ app limitations	Early adoption, more control	Early adoption, infrastructure-dependent
Operational Complexity	Low (if RP already exists)	Medium-High	High
Performance Impact	Low	Medium (extra hops/tunnels)	Medium-Low (long-lived, but routing-heavy)
Scalability	High	Medium	Low-Medium
Client-Side Impact	None	Requires setup	Requires setup
Sniffing (SNDL)	Protected	Protected	Protected
Authentication impersonation	Not protected	Partially Protected	Partially Protected

Securing Central Bank Regulatory Files Transfers with PQC
Quantum-safe experiment report
October 2025

Authors

Nicolas Margaine
Cryptography Project Manager,
Banque de France

Pierre Pouliquen
Cryptography Project Manager,
Banque de France & Banque de France team

Olivier Salomon
Head of Quantum Technologies Hub & deeptech expert
Allianz SE CXTT

Sponsors

Karim Akhavan-Hezavei
Head of Group Technology (CIO|CTO) & Head of COO Transformation
Allianz SE

Ralf Schneider
Senior Fellow and Head of Cybersecurity and NextGenIT Think Tank
Allianz SE CXTT

Loic Tourlourat
Chief Information Office
Allianz FR

Olivier Lantran
Head of the Banque de France Innovation Centre
Banque de France

825446 BdF Dircom Studio Création - 10/2025

BANQUE DE FRANCE

EUROSYSTÈME