Future Of Cryptography

Understanding Quantum-Safe Timelines and Deployments

A Technical Paper prepared for SCTE by

Massimiliano Pala
Director, PKI Architectures and Principal Architect
CableLabs Inc.
858 Coal Creek Cir, Louisville, CO 80027
+1 (303) 661-3331
m.pala@cablelabs.com

Table of Contents

Title **Page Number**

List of Figures

Title **Page Number**

List of Tables

Title **Page Number**

1. Introduction

The digital era, recently fueled by amazing advancements and synergies in computer science and information technology, has placed an unprecedented reliance on cryptography for securing data. However, the very cryptography that provides security and resilience across the world now stands on the cusp of a significant paradigm shift. The advent of quantum computers, a technological achievement in itself, is indeed poised to challenge the very bedrock of contemporary cryptographic systems, thus urging for fundamental shifts in our algorithms adoptions. In fact, quantum computers, with their fundamental differences in their computational capabilities from classical computers, have the potential to threaten current encryption and authentication systems. Traditional encryption schemes like RSA [Rsa16] or ECDSA [Ec05] rely heavily on mathematical problems such as factoring large integers and discrete logarithm problems, tasks that are computationally infeasible for classical computers but can be swiftly dealt with by a Cryptographic Relevant Quantum Computer (CRQC).

In the face of these imminent threats to digital security infrastructure, the security community must respond urgently with an adaptable and future-proof strategy: crypto-agility. This strategy, which emphasizes the ability to shift from one cryptographic system or algorithm to another, is quickly becoming a necessity rather than an option.

This paper aims to delve into the impending risks posed by quantum computing to present-day cryptography, scrutinize the challenges in crypto-analysis, and explore the need for the adoption of crypto-agility processes within digital security paradigms. In this work we focus on providing important considerations on practical timelines for quantum-safe migrations for the broadband community together with a look at *Future of Cryptography* project that aims at providing guidelines for how to build a safer and secure future in the quantum age.

1.1. Quantum and the Broadband Industry

Quantum computers pose a particular danger to broadband network operators and indeed all sectors relying on authentication and encryption for data security. The primary reason is that quantum computers are exceptionally good at solving the factoring problem — i.e., factoring a large number into its prime components. Traditionally, this fundamental problem has provided the backbone of many encryption techniques, including RSA (Rivest-Shamir-Adleman) encryption, which is one of the most widely authentication scheme used in the broadband industry.

However, a fault-tolerant quantum computer with enough qubits can solve these factoring problems far more quickly and efficiently than a classical computer, potentially breaking the security of device identities and trusted authorities (i.e., digital certificates) almost effortlessly. Such quantum systems are referred to as Cryptographic Relevant Quantum Computers or CRQCs.

If and when these computational devices become practical, they could disrupt the authentication methods currently used by broadband network operators to transfer authorization and encryption keys, making it essential for these organizations to start preparing alternative encryption methodologies. This implies that securely transmitted data today (or stored in the Company's databases) could be captured and stored for decryption later, especially if the process leverages quantum-vulnerable key exchange mechanisms.

1.2. Encryption and User-Level Privacy under Quantum

In the broadband industry, encryption is used both to protect corporate data and Access Network communications. For example, when considering user-level privacy, DOCSIS [SECv4.0] uses BPI+

[BPI+08] to provide privacy of communication between the CM and the CMTS. In this context, it is therefore important to understand the practical impact of Quantum Computing and Grover's Algorithm [Gro96] on symmetric encryption, particularly focusing on the Advanced Encryption Standard (AES).

In the quantum computing world, Grover's Algorithm, developed in 1996, theoretically allows for a very efficient search for unstructured data (black-box function) by leveraging the possibility to construct all possible output in a single operation via the implementation of an Oracle function and the use of quantum-mechanical properties such as superposition and entanglement. The algorithm, however, does not offer the same significant speed-up as Shor's quantum algorithm for factoring large integers.

This means that cracking a symmetric cipher like AES-128 with a quantum computer using Grover's Algorithm would still require significant computational resources, involving serial computing operations in a quantum setting that is currently thought to be a quite difficult task (i.e., more than 2^{64+} operations). Thus, switching from AES-128 to AES-256 due to quantum computing threats might need to happen a bit further in the future than originally expected, given today's understanding of quantum-computing and its limitations. Even institutions like NIST are still convinced that AES-128 will likely remain secure against quantum threats for several more decades because of this hardware dependency and lack of parallelizable quantum algorithms.

Nevertheless, many different environments are preparing to switch to higher entropy encryption systems (or enabling the possibility to do so), should the assumptions behind the implementation of the Grover's algorithms should not hold in the future [3GPP23].

1.2.1. Grover's Estimation and Entropy Requirements

Although the use of AES-128 can still be considered secure for a decade or more, a recent paper from ATIS [Deakin23] quantifies the effect of poor entropy on AES encryption. The paper shows how the run time of Grover's algorithm on a quantum computer can be significantly reduced unless the source of entropy is, in fact, truly random. Furthermore, as quantum computer gate fidelity improves over time, also the number of logical qubits required to run Grover's algorithm is further reduced – this combined effect with poor entropy could make symmetric key encryption less resistant to quantum attack than many expect. Using the equations 9 and 11 from ATIS' paper, we can provide an evaluation of the relative security of AES assuming keys are produced from low entropy sources.

Let's assume that the search space is n = 256, thus requiring the size of the key to be 256 bits.

According to [Deakin23], Grover's run time, namely how long the algorithm is expected to run given a physical key of n-bits produced from an h-bit entropy source, can be approximated by the time needed to run the Oracle operator followed by the execution of the diffusion operator for $\sqrt{N}$ (where N is equal to 2^{256}). Therefore, the runtime can be expressed as:

$$Runtime\ (n,h) = 2^{h/2}(n^2 + 4n + c_n) + 2n$$

Where ‘h’ represents the number of entropy bits where the value ranges between 32 to 256 bits.

Figure 1 from the Deakin’s paper shows how important the use of a high source of entropy is when evaluating the expected computational cycles of Grover as the exponential nature of the runtime curve manifests itself after the entropy exceeds that of approximately 220 bits.

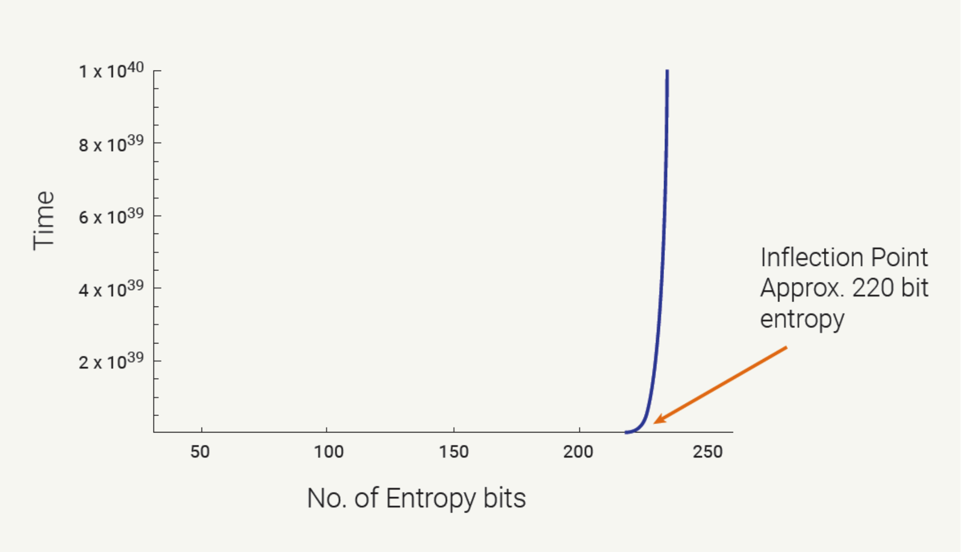

Figure 1 - Expected Running Time (Computational Cycles) for Grover's Algorithm.

It is important to highlight how this analysis not only shows how important it is to have larger keys to protect against quantum computers, but also how paramount it is to ensure that it is produced from a high entropy source. In other words, although we currently estimate the current generation of encryption algorithms (e.g., AES-128) to still be secure in the context of the broadband industry for a couple of decades, this work shows how important it will be to rely on true source of randomness to generate symmetric keys with the required number of entropy bits (e.g., how truly “random” the bits generated for the symmetric key are) and not just doubling the size of the key without relying on true random sources.

1.2.2. Quantum Technologies for Cryptography

Quantum technologies comprise many different fields, not only quantum computing. In particular, the better understanding of quantum-mechanics processes can be used to build new quantum-based devices (e.g., quantum-based sensing) that can be used, for example, to provide functions that are provable secure. Specifically, when it comes to security, the use of quantum technologies can help in solving one of the most fundamental and hard problems: the generation of random bits. Although conventional hardware random number generators continue to be robust and secure, Quantum Random Number Generators (QRNGs) are today available on the market that provide verifiable sources of randomness that can better address advanced entropy/randomness requirements.

In conclusion, it is important to underline that the source and quality of entropy will have a significant impact on the security of symmetric key cryptography to be quantum resistant. Specifically, as we have seen in the previous section, unless a source of entropy used for 256-bit symmetric key AES is truly random across at least 225 bits, it may have little effect in terms of being any more secure against the quantum-threat.

1.3. A Crypto-Agility Example for the Broadband Industry: The CARAF Framework

When it comes to the deployment of new cryptography, the fundamental step that is crucial for the success of the migration plan is the availability of a clear map of what is available today, a detailed prioritization plan, and a cost-effective action plan that both fits the plans from the ecosystem and the vendor's community.

In order to foster good processes across a single organization and/or a whole industry sectors, many companies will have to identify processes that can be used for a successful multi-year effort such as quantum-safe migration. To help with the internal organization and the coordination across industry-sectors and supply-chains, some frameworks that focus on crypto-agility have been developed. The CARAF framework [Car20] is one of these framework that outlines strategies to develop or maintain crypto-agility within an organization.

CARAF stands for *Cryptographic Agility Risk Assessment and Management Framework* and, in summary, is a tool to help managing the crypto assets of an organization by understanding where and how cryptography is used within the organization, conducting a risk assessment based on the criticality of operations and business functions, prioritizing risk mitigation activities, and ongoing management of cryptographic systems.

The core of CARAF, and other similar frameworks, is aimed at ensuring that organizations are prepared for any cryptographic changes via five main areas of interest/work:

1. **Creation of a Crypto Application Inventory.** Keeping track of all cryptographic algorithms and where they are used is the first step for any organization. This step of the process includes noting any hardware, software, and systems that relies on any element of cryptography.
2. **Categorization.** Categorize relevant assets (e.g., hardware, software, etc.) based on operational and business criticality, i.e., how important they are to ongoing operations and business functions.
3. **Assessment.** Assess the risks associated with each category. A higher risk might be assigned to systems utilizing outdated or weak cryptographic algorithms, those with no available alternate algorithms, or systems which are integral to important business functions.
4. **Prioritization.** Prioritize risk mitigation strategies based on the risk assessment findings and business needs. Although it is important to address high risk areas first, there might be other important considerations factors that might change the risk-based priorities.
5. **Management.** Implement changes, monitor effectiveness, and ensure to regularly repeat the crypto-agility process to keep your systems secure and up-to-date.

It is important that Companies start exploring the availability of such tools and start defining appropriate process for the specific organization, possibly in harmony with the larger ecosystem(s) the Company is a member of.

1.3.1. Different Types of Migration Strategies

As discussed in the previous section, the migration towards quantum-resistant cryptography is a critical step considering the potential threats posed by quantum computing. To effectively transition, a risk-informed strategy that involves a Quantum Risk Assessment (QRA) is necessary to identify and prioritize which assets will be affected. The options to migrate systems from quantum vulnerability to quantum resistance include a one-time migration, crypto agility by design, or a hybrid solution.

A one-time migration involves the singular migration of prioritized technologies to standardized post-quantum cryptography algorithms. It ranges from hardware replacement to a software patch, with constant monitoring of the quantum landscape to ensure the selected method remains quantum resistant.

Crypto Agility by design, instead, refers to the ability to quickly change cryptographic algorithms without requiring long downtimes or extensive code revisions. Products should be designed to swap in different cryptographic algorithms, including PQC, and they should be able to migrate quickly to NIST's PQC algorithms once they are available. When it comes to the practical aspects of updating credentials, especially device ones, there are many challenges that need to be considered. Specifically, although protocols exist that can provide some level of automation (provided that the software has been updated to handle the new algorithms), the possibility for infrastructure-deployed devices to directly reach certificate-renewal services might be limited (e.g., no direct access outside the operator's network) and additional considerations are warranted about the deployment of hybrid (multi-algorithm) identities.

The hybrid approach involves the simultaneous use of PQC and classical algorithms for comprehensive security. This approach acknowledges that vulnerabilities may still be discovered in quantum-resistant cryptography, hence the need to maintain the use of classical algorithms. A downside of the hybrid approach is that it does increase the overhead from cryptography and introduce new challenges related to certificate revocation. On the increased size of cryptography, it is important to notice how the addition of traditional keys and signatures to quantum-safe ones only increases the over-all size of the certificate to be less than 10%. Table 1 provides examples of one type of hybrid technology, i.e., Composite Crypto [Pala23], and the size of the average hybrid certificate when compared to the equivalent non-hybrid version.

Table 1 - Size of Digital Certificates that use Hybrid Technologies (Explicit Composite)

Algorithm	Sec. Level	PubKey Size	Sig Size	Cert Size
id-Dilithium3-RSA-SHA256	192 (112)	2275 (+320)	3564 (+270)	6103 (+600)
id-Dilithium5-ED448	192 (224)	2690 (+97)	3294 (+126)	7660 (+208)
id-Falcon512-ED25519	128 (128)	964 (+66)	734 (+79)	1932 (+136)
id-Falcon512-P256-SHA256	128 (128)	1101 (+203)	738 (+83)	1993 (+197)

All sizes are all expressed in number of bytes and the difference with the non-hybrid version of the certificate is reported in parenthesis. Other mechanism to provide hybrid approaches might focus on the use of multiple certificates (or identities) which, usually, require changes in protocols and/or application logic for their deployments.

Despite these challenges, the migration towards quantum-resistant cryptography is a necessary step to ensure robust security and different industry sectors will need to decide which migration strategy better fits the specific ecosystem and individual companies.

2. The Future of Cryptography Project

To help with the definition of community-wide education, experimentation, and guidelines production, CableLabs and its Members have started a new project named Future of Cryptography. This initiative stems from the previous work done under the Extensible and Programmable Infrastructure for Cryptography and Access (EPIC Access) project, led by Steve Goeringer, which identified potential impacts of crypto failures on the industry and the need for a cost-effective plan for deploying crypto-agility processes.

The Future of Cryptography is detailed as a three-phased project involving education, experimentation, and standardization guidelines.

The first activities for the project are focused on collaboratively building an understanding of threats and solutions related to the deployment of quantum-safe cryptography by leveraging the input from experts across the community. During the second phase, the project aims at experiment with different methodologies for migration to post-quantum solutions, that will result in understanding the practical impact of the new cryptography. In phase three, the project's final efforts will be focused on summarizing the results of the experiments and provide the community with comprehensive guidelines that are meant to be leveraged for standardizing the use of the new cryptography within the broadband community and specifications.

Ultimately, the Future of Cryptography project is an example of a crucial activity that will provide the groundwork towards creating crypto-agile processes across the industry. However, to better understand the length of the timelines that lie in front of us, we provide, in the next few sections, considerations on the processes that need to happen *after* the new algorithm's standardization has been finalized and *before* the industry is ready to start the deployment of the new algorithms.

2.1. Quantum, Security, and Timelines

Today, we start building our Trust by embedding long-term keys in the form of Trust Anchors that are usually distributed via Self-Signed X.509 certificates (also called Root CAs) [X509]. Because the secure distribution of Trust Anchors (TA) is one of the most difficult problems to be solved in security, TAs are usually required to be somewhat long lived across multiple years or even decades. It is not uncommon to have TAs that are meant to provide the root of trust for specific ecosystems (i.e., the WebPKI, DOCSIS, Matter, etc.) for 30 or 50 years. Following similar considerations for the intermediates and edges of the Trust Infrastructure (i.e., Intermediate CAs and End-Entity certificates), it is easy to understand how updating or replacing such identities in the field requires large investments and, in many cases, hardware replacements.

When it comes to the distribution of the authorization key (or session key) in protocols, the situation is quite different. Specifically, at least today, the key exchange process is usually ephemeral, and the public keys used in the process are never persisted across sessions (e.g., ECDHE). This means that switching between different key exchange mechanisms (or key encapsulation ones) does not require global support for the algorithms (such as the public key used in the Root of a PKI), only local support between the entities participating in the communication – thus removing the need for algorithmic long-term commitment. Similarly, for Key Encapsulation Mechanisms or KEMs [Dent02], algorithms are expected to use ephemeral keys (i.e., negotiated at a local level).

In other words, while new algorithms for key exchange/key encapsulation mechanisms can be *deployed and negotiated locally*, planning for new algorithms for Trusted Identities is a *global issue* (in the sense of the entire ecosystem that relies on those identities).

2.1.1. Deployment Timelines and the Mosca Framework

Let's now look at some back-of-the-envelope estimations for when quantum-safe solutions (not just algorithms) can be reliably deployed across the broadband industry and, ultimately, across the world.

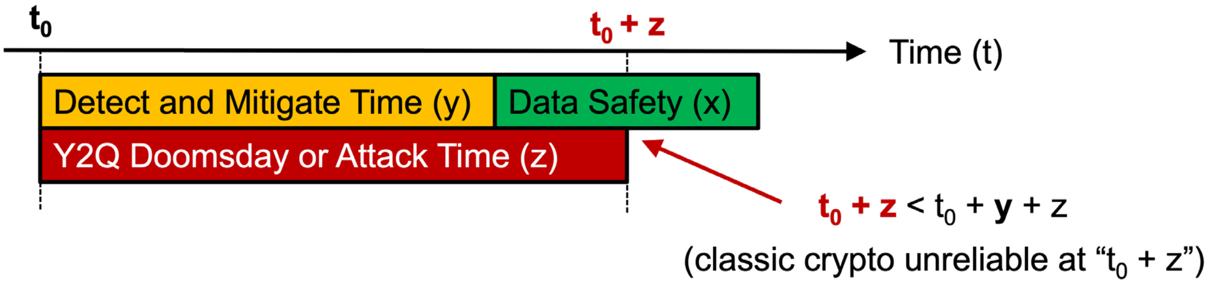
t_0
$t_0 + z$
Time (t)
Detect and Mitigate Time (y)
Data Safety (x)
Y2Q Doomsday or Attack Time (z)
$t_0 + z < t_0 + y + z$
(classic crypto unreliable at "$t_0 + z$")

Figure 2 - The Mosca Framework

In Figure 2, we use the Mosca Framework [Mosca15] where t_0 is the time when the deployment of quantum-safe solutions starts. In this framework, the time needed to complete the migration to quantum-safe solutions is indicated by the variable y, while the time until a CRQC is available is indicated with z.

Given the need for multi-year processes for migrating all systems to quantum-safety, we can estimate a base number between 5 and 10 years for the value of y. Additionally, if there are data safety requirements, an additional period, indicated with the variable x (e.g., another 3 to 5 years), to achieve full data safety (i.e., the green bar in Figure 1). When it comes to the estimation of z, we can use a large fork to encompass many different positions about the engineering feasibility and timelines for CRQCs. For our discussion, we will use a value of z between 7 and 20 years.

With these numbers in mind, if we already had all the algorithms standardized and integrated with existing protocols today, if all the TAs had already been upgraded and available for integration with products and network infrastructures, if all vendors had deploy-able and interoperable solutions that can be bought on the market…. That is when we could say that today is t0, however, it is clear that we are quite far from that today.

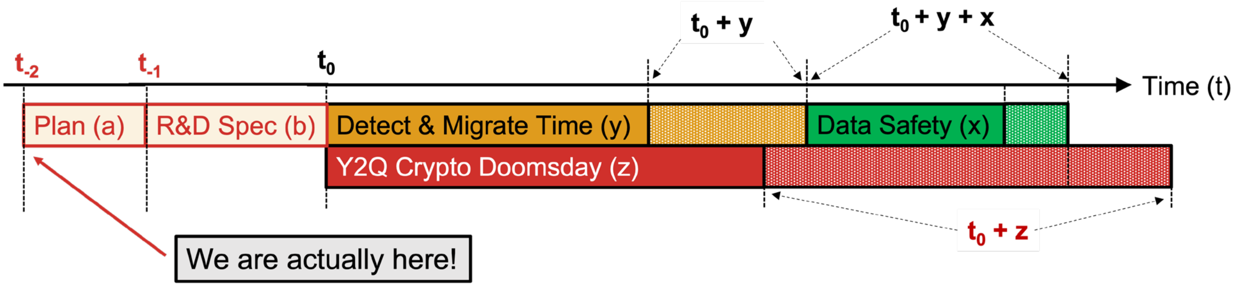

Figure 3 - The Mosca-Pala framework

In Figure 2 we propose an updated formulation of the Mosca framework, the Mosca-Pala framework, where two additional variables are introduced.

We use the variable 'b' to indicate the time needed for the different ecosystems (e.g., Broadband Industry, IoT, Internet Routing, DNS, etc.) to integrate the new algorithms in their protocols. We estimate this process to take between 3 and 5 years (we are very optimistic!). This time ‘b’ includes the development of new specifications and products for the market, where embedded or dedicated hardware needs to be upgraded (e.g., System on a Chip or SoC).

Even if we had the algorithms and implementations already standardized and available today, there is another variable that needs to be taken in considerations: the need for ecosystems to experiment and agree on which solutions to adopt (e.g., straightforward replacement? Hybrid deployments? etc.). We call this variable ‘a’ and we estimate its value to be between 2 and 5 years for most environments.

We can finally put everything together and try to estimate how much time will be needed to complete the migration to quantum-safe solutions in different industry sectors such as communications:

$$\text{Migration Time} = a + b + y + x$$

Which gives us a low and high boundary for our estimations to be the following Time To Quantum-Safety or TTQS:

$$TTQS_{MIN} = 2 + 3 + 5 + 3 = 13 \text{ (years)}$$

$$TTQS_{MAX} = 5 + 5 + 10 + 5 = 25 \text{ (years)}$$

Therefore, even with the most optimistic estimates, we are looking at a timeline of over a decade, at the minimum, to complete the migration to quantum-safe solutions, and, potentially, as long as 25 years given the complexities and challenges involved.

Action to prepare for quantum-safe solutions must be taken well in advance of the realization of practical quantum computing and given these estimations for migration timelines and the range of estimates for when a CRQC might become available, it is critical that organizations begin to evaluate and implement quantum-safe solutions now to ensure a smooth and successful transition in the future.

A typical migration timeline for the broadband industry is provided in Figure 4 where we highlight the two new variables of the Mosca-Pala framework: the Planning Time (‘a’) and the Specifications/Products Updating Time (‘b’). During this phase of the migration, we envision the need for ecosystems to evaluate and experiment with tools that provide practical information about risk changes and operational needs. For example, even while algorithms and their parameters are being standardized to address first deployments and risk-mitigating solution (horizontal or vertical algorithm-agility), the specific ecosystem(s) may start collecting, pending the availability of easy-to-use tools and implementations for prototypes and simulations, data on the use of different algorithms, their performances, and experienced bottlenecks in relation to the specific protocol where the new cryptography is being tested. Other factors should also be evaluated such as what lifetime for the device/software is envisioned and the availability of hybrid solutions.

2.1.2. Migration Timelines: An Example

In this section we propose imaginary timeline for migrating an ecosystem to use quantum-safe solutions. For this scenario, let’s imagine that the specific industry sector or ecosystem starts paying attention to the quantum-threat and how to address it at time T_{INIT} which can be easily expressed as:

$$T_{INIT} = t_0 - a - b.$$

Figure 4 - Example Timeline for QS migrations based on the Mosca-Pala framework.

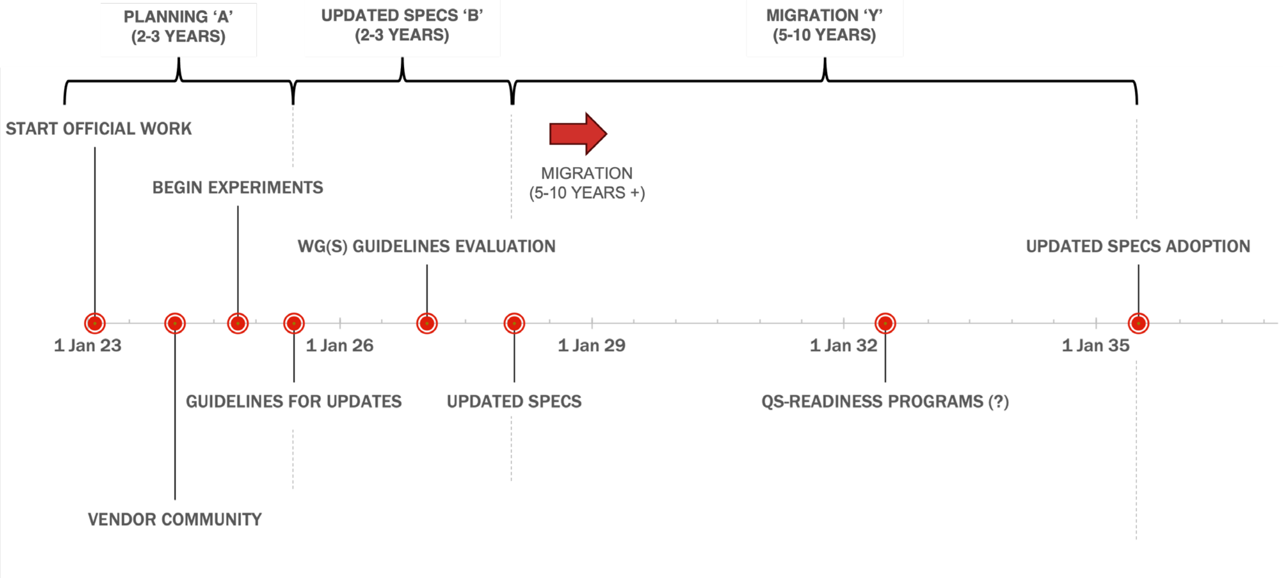

Figure 4 depicts the envisioned scenario where we can imagine that the ecosystem was ready to start its planning phase at the beginning of 2023.

During the initial phase of the migration effort, after involving the vendors' community, the initial set of experimental results should be shared across the whole community to evaluate risks, performances, and costs of each individual solution. On top of that, each entity in the ecosystem should list and prioritize assets updates across the board also in accordance with the envisioned timelines from the vendors' community. If everything goes well and specific requirements and solutions have been identified, we are now around 2026.

At this point, after agreeing on an interoperable approach that might involve the adoption of multiple algorithms (useful to guarantee the implementation of crypto-agility controls that go beyond the single algorithm replacement), the community needs to update the specifications of protocols and devices deployed across their networks. This second phase of the migration must make use of the results produced by the community in the first phase to inform how the specifications must change, when to adopt the new standards, and when devices and certification programs will be available to support such solutions. An important part of this second phase is also defining the characteristics of the Trust Infrastructures (or PKIs) and associated processes to provide a quantum-safe migration paths that different protocols, working-groups, and corporate environments will integrate. We envision this phase to take another three (3) to five (5) years to complete.

Last but not least is the deployment phase. This is the phase that individual operators have absolute control over. In fact, although the specifications and products might be ready for adoption at this stage, cost-related considerations and risk management considerations might drive the effective migration planning. During this phase, it is important to understand that coordination and planning must occur at all levels: not only across the supply-chain for corporate systems, but also explore the dependencies, requirements, and limitations of the access network. In our scenario, for example, even if the new specifications and some products are available for purchasing and deployment before, not all vendors might be ready. Delays in the actual migration might come from the lack of solutions (or upgrade path) from specific vendors.

3. DOCSIS and Quantum-Safety

Several technical solutions have been proposed in the literature that investigate the possibility for delivering quantum-safe solutions not only for the newer versions of DOCSIS (4.0+), but also for earlier versions by leveraging a new registration process [Pala20, Pala21]. In this work, we took a step further and built a tool to provide practical measurements and initial comparison between the use of traditional vs. quantum-safe algorithms in BPI+.

In the following sections we share the results of the experiments we ran aimed at the impact of the new quantum-safe algorithms with BPI+ V2 [SECv4.0] authentication protocol. This is as an example of the activities needed, within the different eco-systems, to understand the impact of enabling the use of quantum-safe cryptography in networks and applications.

3.1. The BPI+ V2 Authentication Protocol

Before we provide a description of the simulation framework and the collected results, it is important to review the core design of the BPI+ V2 protocol and highlight the characteristics that could make it a candidate for drop-in replacement with quantum-safe cryptography.

BPI+ V2 design introduced several enhancements (when compared to BPI+ V1) that removed, on paper, the limitations for the deployment of quantum-safe cryptography, such as:

- The use of digital signatures with trusted X.509 certificates to authenticate BPKM messages
- The introduction of ECDHE key-exchange mechanisms for the derivation of authorization keys in alignment with TLSv1.3 [RFC8446]
- The use of fragmentation-enabled messages to extend the max supported message size (MMM V5)

The core of the BPI+ V2 messages (i.e., the Auth Request and Auth Reply ones) is depicted in Figure 5 where the Auth Info message is omitted for clarity.

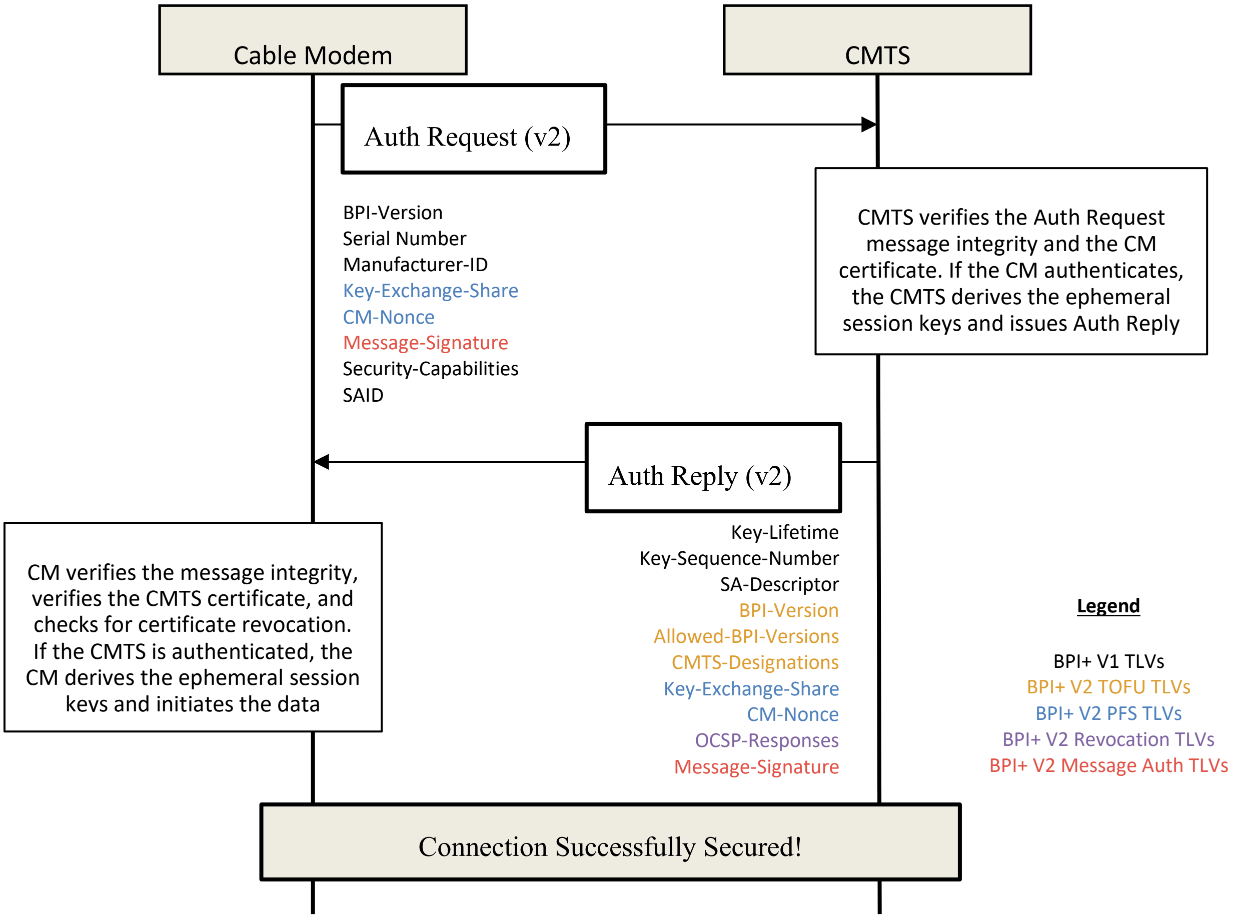
Cable Modem
CMTS
Auth Request (v2)
BPI-Version
Serial Number
Manufacturer-ID
Key-Exchange-Share
CM-Nonce
Message-Signature
Security-Capabilities
SAID
CMTS verifies the Auth Request message integrity and the CM certificate. If the CM authenticates, the CMTS derives the ephemeral session keys and issues Auth Reply
Auth Reply (v2)
Key-Lifetime
Key-Sequence-Number
SA-Descriptor
BPI-Version
Allowed-BPI-Versions
CMTS-Designations
Key-Exchange-Share
CM-Nonce
OCSP-Responses
Message-Signature
CM verifies the message integrity, verifies the CMTS certificate, and checks for certificate revocation. If the CMTS is authenticated, the CM derives the ephemeral session keys and initiates the data
Legend
BPI+ V1 TLVs
BPI+ V2 TOFU TLVs
BPI+ V2 PFS TLVs
BPI+ V2 Revocation TLVs
BPI+ V2 Message Auth TLVs
Connection Successfully Secured!

Figure 5 - BPI+ V2 authorization process (Auth Req and Auth Reply only)

3.2. The Simulation Framework

To better understand the impact of quantum-safe algorithms on core broadband algorithms, CableLabs built an experimental simulator for BPI+ V1 and V2, namely *LibBPLUS* framework.

Although the framework supports both versions of the BPI+ protocol, given the limitations of the type of MAC message used in the implementation of BPI+ V1 (i.e., MMM V1 that support maximum sizes smaller than 2000 bytes), BPI+ V1 is not suitable for drop-in replacement(s). Moreover, the explicit dependency on the RSA algorithm for delivering the authorization key to the CM (i.e., the lack of authenticated key exchange mechanisms) makes it impossible to use device certificates with quantum-safe public keys.

However, when it comes to BPI+ V2, all those limitations have been removed as described in the previous Section and the rest of the paper focuses on latest version of the protocol (i.e., BPI+ V2).

3.2.1. The Framework in a Nutshell

The LibBPLUS framework has been developed to be easily extensible and easy to integrate in existing and future systems. Its main purpose is to provide an implementation of the BPI+ authentication protocol (both V1 and V2) that can be used to test different cryptographic and communication parameters, even experimental ones (i.e., BPI+ V1 and BPI+ V2 messages, authentication certificates, post-quantum cryptography, etc.). It is important to highlight that this work, LibBPLUS, is not meant for production use – the code is not optimized and is meant to only provide early access to a reference implementation, even when devices that support new versions of the protocol are not available yet.

The codebase comes with a shared library, a message generation and parsing tool, and a TCP-based simulator that implements both client and server side. The project depends on few open-source packages to be installed (SDKs are needed for compilation) such as LibPKI [LibPKI], LibOQS (optional, only needed for QS-support) [LibOQS], and finally OpenSSL-OQS (1.1.1) [OSSL-OQS] or OpenSSL (1.1.1) [OSSL].

The LibBPLUS library has been architected with the developer and researcher in mind by providing different abstraction layers and easy integration options. The core piece of the LibBPLUS project is distributed as a shared library that can be easily linked in C/C++ applications and/or Python/Java/Node/etc. wrappers. The library is organized in a set of abstraction layers that implement different layers of functionalities: from a very high-level abstraction layer that provides easy-to-use message creation functions, to the lower-level BPKM and TLV libraries that allow for customization of the messages (even outside the standardized parameters).

The abstraction layers are organized as follows, from the higher level to the lowest:

- **BPLUS_CTX** - A complete set of support tools for loading and using X.509 certificates, Keys, Configurations, Server Configurations, etc. It also provides the support layer for the authorization and Traffic Encryption Key or TEK state machines.
- **BPLUS_API** – A high-level set of functionalities to generate and manage BPKM messages. The API provides message-specific functionalities such as generating the public key for the key exchange mechanism or deriving the authorization key from Auth Request and Auth Reply messages.
- **BPLUS_BPKM** - Provides support for BPKM message generation, encoding, and decoding. Specifically, the BPKM sub-library provides the definition of BPKM messages (grouped by BPI+ version) and associated management functions.
- **BPLUS_TLV** - Provides low-level layer for TLV generation, encoding, and decoding. The TLV sub-library provides the definition of TLVs (also grouped by BPI+ version) and associated management functions.

Additionally, the LibBPLUS project comes with a fully fledge network simulator and message generator that is provided via a separate Command Line Interface (CLI) tool called ‘bplus’:

- **BPLUS Tool** - A tool that leverages the BPLUS library to provide a threefold implementation: a message generator and parser, a BPI+ Auth state machine for server-side implementations (TCP server), and a BPI+ Auth state machine for client-side implementations (TCP client).

The package comes with extensive tests (src/tests) and tools (src/tools) that show the usage of the library in different scenarios. Additionally, extensive documentation for the supported APIs for the different abstraction layers is provided directly in the header files that modern IDEs such as VS Code made readily available to the developer[1].

3.1. Current Framework Limitations

As we hinted in the introduction to the section, the LibBPLUS framework is highly experimental and comes with some limitations that we are aware of and plan to remove in the next generation.

On the message generation and parsing front via the CLI application, the tool is still limited to be able to generate only few types of messages such as Auth Info, but it does not currently support the generation of all types of BPI+ messages.

On the simulation front, although the core of the simulator is designed to support multiple concurrent connections, the implementation is currently limited to handling a single individual authentication and encryption key delivery. After the Authorization State Machine (ASM) reaches the "Authorized" state and the TEK state machine reaches the "Operational" state, the simulation is terminated on both sides.

When it comes to implementation of quantum-safe testing, the dependency on the availability of implementations or, more precisely, the lack thereof, has proven to be a challenging problem to solve. In particular, the enhancements that we needed to provide for the crypto libraries can be summarized as follows:

- **Provide an implementation for hash-n-sign paradigm (Quantum Safe)**. This enhancement allows the crypto layer to be able to use the traditional model we are used to when it comes to signing. In particular, the provided enhancements allow to sign the CMS structure that is used for authenticating the messages (i.e., detached CMS signature) by pre-hashing the content and then signing the hash value (instead of signing the entire message directly). This was needed to get around OpenSSL's requirements to use the hash-n-sign paradigm for signing CMS structures.

- **Provide an implementation for Hybrid certificates (Composite Crypto).** This enhancement implements the Composite Crypto technology developed at CableLabs to support the testing of hybrid solutions. The availability of such technology is aimed at studying the impact of using multiple algorithms to lower the risk coming from the adoption of new cryptography without the need for changing the application logic to use, for example, multiple certificates.

On the front of Key Encapsulation Mechanisms (KEMs), there is still much work to do.

Since the OpenSSL-OQS wrapper does not support the generation and use of key pairs that use the Kyber algorithm (currently the only one selected for quantum-safe KEM standardization), the simulator does not currently have the option of replacing traditional key exchange mechanisms with quantum-resistant KEMs. This limitation is to be kept in mind during the discussion of the collected results that do not include the use of quantum-safe KEMs, even in the quantum-safe tests.

Finally, the simulator does not provide support for the encapsulation of BPKM messages (i.e, the BPI+ ones) inside MAC Management Messages. This means that the simulator does not create full MMM messages where BPKM ones are embedded as the payload of an MMM, but, instead, uses the TCP layer to transfer BPKM messages directly across the client and the server. Consequently, even for large BPI+ V2 BPKM messages the simulator does not have to fragment and then reconstruct messages that are

[1] The GITHUB repository is not currently publicly available. We are working to understand the implications (and required processes and improvements) for releasing the code to the community.

larger than a single DOCSIS frame (e.g., 2000 bytes) since it relies on the fragmentation support built into the TCP/IP layer.

3.2. The Methodology

To collect the experimental data, we ran several simulations where we used different types of certificates and provide a comparison among the collected results. All experiments have been run on a MacBook Pro laptop (2.3 GHz 8-core Intel Core i9) with 16Gb of RAM. In this first phase of experimentation, we focused our metrics on two dimensions only: size of the authentication trace and time to execute the authorization process.

First, we ran BPI+ V1 and BPI+ V2 simulations to get baseline values that are used to build the comparison matrix. After that, we configured the simulator to use certificates from a quantum-safe PKI where we used CRYSTALS-Dilithium keys at level 2 (the lowest security level provided by this algorithm) for the whole chain [Dil17].

During the simulation, every message received or sent is automatically captured by the framework and saved both in binary and textual format on local storage for later analysis – although these operations added extra execution time that we needed to be aware of, it did not have an impact on the overall set of measurements. The framework also saves the contents of the `Message-Signature` TLVs of authenticated messages in separate files. Additionally, extensive logs are also being recorded that provide additional information on the execution of the protocol such as the selected key-exchange field (e.g., P256, X448, etc.) or possible issues with the validation of signatures and certificates.

Table 2 - Message sizes for BPI+ V1, BPI+ V2, and BPI+ V2 QS.

Message Type	BPI+ Version	Certificate Type	Message Size	Signature TLV
Auth Info	BPI+ V1	RSA	1,067	n/a
Auth Request	BPI+ V1	RSA	1,400	n/a
Auth Reply	BPI+ V1	RSA	291	n/a
Auth Info	BPI+ V2	RSA	1,067	n/a
Auth Request	BPI+ V2	RSA	1,703	1,564*
Auth Reply	BPI+ V2	RSA	2,809	2,689**
Auth Info	BPI+ V2	Dilithium (level 2)	4,242	n/a
Auth Request	BPI+ V2	Dilithium (level 2)	7,053	6,914*
Auth Reply	BPI+ V2	Dilithium (level 2)	11,318	11,198**

3.3. The Results

The results are summarized in Table 2 where we provide the results from the BPI+ V1 runs with traditional cryptography (RSA PKI) and BPI+ V2 with traditional cryptography (RSA PKI) and quantum-safe cryptography (CRYSTALS-Dilithium Level 2) for certificates and signatures. As we detailed in Section 3.1, the key exchange mechanism has not been updated to quantum-safe variations because of the limitations of the OpenSSL-OQS library.

3.3.1. Considerations on Message Sizes

Since the size of crypto is one of the important factors for drop-in replacement of cryptography, let's first analyze the size of the authorization traces in Table 2 where different types of messages together with their sizes and signatures' sizes are reported. The signature size is the size of the encoded CMS data structure the implements the SigneData type (i.e., detached signature). In the table, we use the '*' (asterisk) symbol to remind the reader that the CMS structure also includes the signer's certificate in the `certs` field of the `SignedData` structure. Similarly, the '**' (double asterisk) symbol is used to remind the reader that the CMS structure that authenticates CMTS messages (e.g., the Auth Reply) not only includes the signer's certificate, but it also includes the CA one (i.e., the CMTS certificate's CA).

As expected, *BPI+ V1 messages with traditional cryptography* (i.e., RSA certificates) are the smallest ones. This is due to the fact that BPI+ V1 does not provide any direct authentication and, therefore, no signatures are generated and/or transferred across the wire.

For *BPI+ V2 authorization process with traditional cryptography* (i.e., RSA certificates), while the Auth Info message size is not changed from BPI+ V1 as it only carries the CA certificate, the Auth Request message size, instead, is increased roughly by the size of the message signature in the order of ~525 bytes. The size of the Auth Reply is also increased considerably. In particular, since BPI+ V2 provides mutual authentication, the Auth Reply message must include not only the CMTS' certificate and the CMTS' message signature, it also must include the CMTS certificate's issuing CA certificate. Therefore, while the size of Auth Request messages can stay below the 1976 limit for a non-fragmented MMM payload, the size of the Auth Reply goes beyond that limit having to include two certificates (i.e., the signer's and the CA's) in the CMS structure.

Ultimately, to test the possibility for drop-in replacement for BPI+ V2, we changed the configuration of the simulator to use X.509 certificates not from a traditional RSA infrastructure, but *from a quantum-safe PKI*. The new infrastructure has been setup as defined in the DOCSIS® PKI (the "new" PKI or 2nd Generation) [ClTi23] and DOCSIS 4.0 SEC specifications (i.e., the certificates' "profile"), except for the public-keys and signatures algorithm. In this case, we used the CRYSTALS-Dilithium algorithm at level 2 for the keys for all levels of the hierarchy (i.e., Root CA, Intermediate CA, and End-Entity). The collected results confirm the expected increase for all the messages in the Authorization process by an approximate factor of four (4). In fact, the Auth Info message is almost four times the size of the same message with traditional cryptography – the increase comes exclusively from the increased size of the CA certificate. The Auth Request message size, instead, is increased four fold because of the increased sizes of both the signature and the CM certificate. Similarly, the Auth Reply message is increased roughly four times because of the CA certificate's increased size together with signature size.

Putting everything together, we can consistently estimate that, in this configuration, quantum-safe BPI+ V2 authorization processes have a four-times increase (4x) in the size of transferred messages when compared to the same protocol using traditional cryptography.

3.3.1. Considerations on Execution Complexity

Although the simulator's is not optimized for speed, since it is in its first release, we can try to provide some simple comparison of the execution times for the BPI+ protocol when using the different configurations.

Before we provide any considerations on the results, though, it is important to remember the limitations of the simulator. Specifically, in the executed simulations we connected the two ends of the simulator via the TCP/IP interface on the localhost. This eliminates the need to account for the time needed to transfer the data over the wire and, therefore, should reduce the variability in reported measurements. Moreover, it is important to consider that both sides of the simulation (client and server) have been executed on the same host where measurements also have been captured via the simulator directly.

Let's first start our analysis by looking at the BPI+ execution with traditional cryptography. In this configuration, while the simulated BPI+ V1 (RSA PKI) takes approximately 50ms to transfer the Authorization Key from the CM to the CMTS, BPI+ V2 (RSA PKI) doubles that time. This is consistent with the use of signatures for both the CM and CMTS because of the increase in the number of cryptographic operations.

Among all the execution times for the different versions of BPI+ with different configurations, the results for the quantum-safe version of BPI+ V2 proved to be the most surprising. In fact, despite the increased size of cryptography, the run time for completing the delivery of the Authorization Key between the CM and CMTS (via ECDHE by using X448) is constantly shorter when compared to using traditional cryptography. Specifically, with the selected configuration, the overall authorization process execution time is reduced to almost half when compared with BPI+ V1. Similarly, when the quantum-safe BPI+ V2 execution time is compared with the traditional one, the quantum-safe one appears to be less than a fourth of the runtime needed to complete BPI+ V2 with RSA certificates (i.e., 20-25ms vs. 100-120ms).

Although interesting, these initial results need to be considered in the context of a simulation and its limitations. More work is envisioned to better understand these initial results.

4. Conclusions and Future Work

In this paper we looked at the *Future of Cryptography* for the broadband industry with the intent of providing important considerations on timelines and required processes for migrating the cryptographic algorithms we use today to quantum-safe solutions.

While in the first part of the paper we introduced the quantum-threat and looked at some practical considerations on the impact of quantum computers over encryption, in the second part of the paper we proposed the Mosca-Pala framework and use it to reason about migration timelines and required steps to execute them. The provided back-of-the-envelope estimations suggest that the industry might not have time to wait to be able to update and deliver quantum-safety across the corporate, access, and converged networks in the next decade (e.g., 10-15 years).

The third and last part of the paper focuses on providing considerations for quantum-safety and DOCSIS. For this work, we built a BPI+ simulator that we then used to simulate the authorization process and compare the results among traditional and quantum-safe implementations. While BPI+ V1 is inherently tied to traditional cryptography (RSA) and cannot be used with quantum-safe certificates (i.e., device identities), the removal of algorithmic dependency between the certificate and the key exchange mechanism makes BPI+ V2 a good candidate for drop-in replacement. In fact, the collected results on the simulation show how BPI+ V2 supports the use of quantum-safe cryptography, thus making BPI+ V2 and DOCSIS 4.0 suitable for algorithm-replacement. On top of that, our experiments show how the use of quantum-safe cryptography could be quite efficient when compared to the RSA algorithm, nevertheless the increased size of crypto-objects.

As described throughout our paper, the work is far from being completed though.

First, we need to better understand the efficiency of quantum-safe algorithms and update the simulator to be able to use quantum-safe KEMs (still not implemented). Once these updates are in place, we also need to investigate the use of algorithms other than CRYSTALS-Dilithium such as Falcon [Fa17], together with the impact of using hybrid technologies such as Composite Crypto or the use of multiple certificates. Last but not least, we need to improve the measurement tools to better understand how the crypto-load might change between traditional and quantum-safe solutions for the different types of protocols, especially on the server-side (i.e., the CMTS, AAA, etc.).

Abbreviations

AES	Advanced Encryption Standard
BPI	Baseline Privacy Interface
BPI+	Baseline Privacy Interface Plus
BPKM	Baseline Privacy Key Management
CA	Certificate Authority
CRL	certificate revocation list
CRQC	Cryptographic Relevant Quantum Computer
DER	Distinguished Encoding Rules
DN	X.500 Distinguished Name
DOCSIS	Data Over Cable Service Interface Specifications
EC	Elliptic-Curves
ECC	Elliptic-Curves Cryptography
ECDH	Elliptic-Curves Diffie-Hellman
ECDSA	Elliptic-Curves Digital Signing Algorithm
EE	end entity
DH	Diffie-Hellman
HSM	hardware security module
IETF	Internet Engineering Task Force Standards Organization
KEM	key encapsulation mechanism
KEX	key exchange (algorithm)

NIST	National Institute of Standards and Technologies
P256	NIST EC Curve at 128bit of security
PKC	public-key cryptography
PKI	public-key infrastructure
QC	quantum computing
QS	Quantum Safe
R-PHY	Remote RF Layer (PHY)
R-MACPHY	Remote Media Access Control and RF Layer (PHY)
RSA	Rivest-Shamir-Adleman (cryptosystem)
SHA-256	Secure Hash Algorithm 2 (256-bit)
SHA-3	Secure Hash Algorithm 3
SCTE	Society of Cable Telecommunications Engineers
SE	Secure Element
SSD	Secure Software Download
TEK	Traffic Encryption Key
TLS	Transport Layer Security
TTQS	Time To Quantum Safety
X448	EC Curve for ECDHE that provides 192 bits of classical security
X25519	EC Curve for ECDHE that provides 128 bits of classical security

Bibliography & References

[Ec05] American National Standards Institute, *Public Key Cryptography for the Financial Services Industry: the Elliptic Curve Digital Signature Algorithm (ECDSA)*, ANSI X9.62, November 2005.

[Rsa16] The Internet Engineering Task Force (IETF) – IETF RFC 8017. PKCS #1: RSA Cryptography Specifications Version 2.2, edited by K. Moriarty et al., November 2016. Also available at https://datatracker.ietf.org/doc/rfc8017/

[Gro96] Lov K. Grover. *A Fast Quantum Mechanical Algorithm for Database Search.* In Gary L. Miller, editor, Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22–24, 1996, pages 212–219. ACM, 1996.

[3GPP23] GSM Association, Post Quantum Telco Network Impact Assessment Whitepaper, Version 1.0, February 2023. Available at: https://www.gsma.com/newsroom/wp-content/uploads/PQ.1-Post-Quantum-Telco-Network-Impact-Assessment-Whitepaper-Version1.0.pdf

[Deakin23] I. Deakin, *Implications of Entropy on Symmetric Key Encryption*. February 2023. ATIS. Available online at: https://www.atis.org/?smd_process_download=1&download_id=1783427

[Car20] Chujiao Ma and others, *CARAF: Crypto Agility Risk Assessment Framework*, Journal of Cybersecurity, Volume 7, Issue 1, 2021. Available online at: https://doi.org/10.1093/cybsec/tyab013

[Pala23] The Internet Engineering Task Force (IETF) – I-D draft-ounsworth-pq-composite-sigs-09 - Composite Keys and Signatures For Use In Internet PKI, edited by M. Ounsworth and M. Pala, Apr 2023. Also available at https://datatracker.ietf.org/doc/draft-ounsworth-pq-composite-sigs/

[X509] ITU-T Recommendation X.509 (2005) | ISO/IEC 9594-8:2005, *Information Technology - Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks*.

[Dent02] Alexander W. Dent, *A Designer's Guide to KEMs*, Cryptology ePrint Archive, 2002. Available online at: https://eprint.iacr.org/2002/174

[Mosca15] Michele Mosca, Cybersecurity in an era with quantum computers: will we be ready?, Cryptology ePrint Archive, 2002. Available online at: https://eprint.iacr.org/2015/1075.pdf

[BPI+08] Data-Over-Cable Service Interface Specifications, Baseline Privacy Plus Interface Specification, CM-SP-BPI+-C01-081104, November 4, 2008.

[SECv4.0] DOCSIS 4.0 Security Specification, CM-SP-SECv4.0-I04-220328, March 28, 2022, Cable Television Laboratories, Inc.

[Pala20] Massimiliano Pala. DOCSIS® PKI: A Proposal for a Next-Generation Quantum-Resistant Infrastructure. SCTE Cable-Tech Expo, October 2020.

[Pala21] Massimiliano Pala. Enabling Encryption and Algorithm Revocation for Post-Quantum DOCSIS Certificates. SCTE Cable-Tech Expo, September 2021.

[RFC8446] The Internet Engineering Task Force (IETF) – IETF RFC 8446. *The Transport Layer Security (TLS) Protocol Version 1.3*, edited by E. Rescorla, Aug 2018. Available online at: https://datatracker.ietf.org/doc/html/rfc8446

[LibPKI] The LibPKI project. https://www.openca.org/projects/libpki

[OSSL] The OpenSSL project. https://www.openssl.org

[LibOQS] The Open Quantum Safe project. https://www.liboqs.org

[OSSL-OQS] The OpenSSL-OQS project. https://www.openssl.org

[OCAOD] The OpenCA OCSPD project. https://www.openca.org

[RFC5652] IETF RFC 5652, R. Housley, *Cryptographic Message Syntax (CMS)*, September 2009.

[Di17] Dilithium-Crystals, *Dilithium digital signature scheme.* Available online at: https://pq-crystals.org/dilithium/

[Fa17] Falcon, *Fast Fourier Lattice-based Compact Signatures over NTRU*. Available online at: https://falcon-sign.info

[ClTi23] CableLabs C-PKI-TI-V1.4, Trust Infrastructure Document (Certificate Templates), November 2022. Available online at: https://www-res.cablelabs.com/wp-content/uploads/2022/12/13105005/CableLabs-Trust-Infrastructure-C-PKI-TI-V1-4-2022-11-15-WEBSITE.pdf