Canadian National Quantum-Readiness

BEST PRACTICES AND GUIDELINES

Version 04 - July 10, 2024

Authored by:

Quantum-Readiness Working Group (QRWG)
of the Canadian Forum for Digital Infrastructure Resilience (CFDIR)

TLP:CLEAR

The contents of this document are **TLP:CLEAR**

CONTENTS

ANNEXES:

APPENDICES:

FOREWORD

On behalf of the Canadian Forum for Digital Infrastructure Resilience (CFDIR), I am pleased to present the latest ***Canadian National Quantum-Readiness Best Practices and Guidelines***.

The CFDIR's Quantum-Readiness Working Group (QRWG) has significantly updated this report to reflect critical advancements in quantum-safe practices. It now includes new guidance on implementing quantum-resilient cryptography within IT systems - a crucial step for protecting sensitive information as we prepare for the transformative impacts of quantum computing.

Within the next 5 to 15 years, quantum computing is expected to evolve to a point where it may break today's widely-used encryption standards. This shift poses a substantial risk to the confidentiality and integrity of digital communications across all sectors, making quantum readiness a priority for Canada's financial system.

Achieving quantum resilience is not a journey that any institution can undertake alone. Recognizing the importance of a coordinated response, the Bank of Canada has actively participated in cross-sector forums and collaborative working groups. These engagements foster the collective knowledge and shared strategies essential for safeguarding Canada's financial infrastructure. The Bank remains committed to advancing a quantum-resilient infrastructure - not only within our systems but also by supporting the sector-wide efforts that will drive resilience across the industry.

Preparing for quantum resilience requires a long lead time. We urge institutions to prioritize quantum-readiness planning now, utilizing this document as a strategic guide at each stage of their journey. This guidance will be regularly updated to reflect the evolving landscape of best practices.

It is essential that Canadian and international organizations collaborate with industry, government, and other key stakeholders to navigate the challenges and opportunities that quantum technologies bring. The CFDIR serves as a model of the public-private collaboration needed to address the transformative impact of quantum in the years ahead.

In closing, I would like to express gratitude to the CFDIR and QRWG members, whose dedicated efforts provided this resource. Their work fosters shared resilience across Canada's digital infrastructure as we collectively adapt to the new age of quantum technology.

The Bank of Canada will continue to actively support and promote sector-wide initiatives that build cybersecurity resilience - and advance quantum readiness - through collaborative efforts in Canada and around the world.

Hisham El-Bihbety

CISO – Bank of Canada

ACKNOWLEDGEMENTS

The contents of this document were developed during the course of CFDIR Quantum-Readiness Working Group (QRWG) meetings and workshops between July 2020 and June 2024.

The information and recommendations contained herein were informed by the active participation and engagement of subject matter experts from the following organizations (listed alphabetically):

CFDIR Members:

Accenture, AWS, BlackBerry, Canarie, CCCS, CIRA, Cisco Systems, Google, IBM, ISED, Microsoft, Nokia, Quantum-Safe Canada, Thales Canada

Canadian Critical Infrastructure (CI) Stakeholders:

Bank of Canada, BMO, CIBC, CTFS, Financial Services Regulatory Authority of Ontario, Manulife Bank, National Bank of Canada, Payments Canada, Royal Bank of Canada, Scotiabank, Shared Services Canada, Sun Life, TD, 2Keys

Quantum-Safe Ecosystem Stakeholders:

Crypto4A, Entrust, evolutionQ, InfoSec Global, ISARA, Quantum Algorithms Institute

Additional Organizations:

Banco Santander

Financial Services Information Sharing and Analysis Center (FS-ISAC)

German Federal Office for Information Security (BSI)

U.S. National Cybersecurity Center of Excellence (NCCoE)

Toronto Metropolitan University - Cybersecurity Research Lab

University of Sherbrooke - Cryptographic Agility research team

University of Waterloo - Open Quantum Safe (OQS) project

A FEW WORDS ON CRYPTOGRAPHY

Throughout this document, the terms “cryptography” and “crypto” mean the practice of cryptography, which includes constructs such as encryption, digital signatures, hashing, and more. In particular, the term “crypto” does not refer to cryptocurrency, which is a form of unregulated digital currency that utilizes cryptography and often blockchain technologies.

REVISION HISTORY

The following table describes the dates of the major changes to this document.

Authors	Date / Version	Notes
CFDIR QRWG Participants (July 2020 – June 2021)	July 7, 2021 / v.01	Initial version of recommended Best Practices developed from the QRWG's pilot project with members of Canada's Finance CI sector.
CFDIR QRWG Participants (July 2021 – June 2022)	June 17, 2022 / v.02	Updated Best Practices reflecting information obtained during the second year of the QRWG's collaboration with members of Canada's finance critical infrastructure sector, including meetings with post-quantum ecosystem stakeholders and three mini-workshops focused on hybrid cryptography.
CFDIR QRWG Participants (July 2022 – June 2023)	June 12, 2023 / v.03	Refreshed and expanded Best Practices including updates to previously published content, plus new guidance on (1) cryptographic-agility and (2) PQC product/service roadmap questions for Information and Communications Technology (ICT) vendors.
CFDIR QRWG Participants (July 2023 – June 2024)	July 10, 2024 / v.04	Refreshed and expanded Best Practices including updates to previously published content, plus new Annex J containing fresh insights into "who has to do what" to migrate the cryptography used in an existing IT system to become Quantum-safe.

1. INTRODUCTION

Cryptographic technologies are used throughout government and industry to authenticate the source and protect the confidentiality and integrity of information that we communicate and store. Cryptographic technologies include a broad range of protocols, schemes, and infrastructures.

> ***Quantum computers will break currently deployed public-key cryptography, and significantly weaken symmetric key cryptography, which are pillars of modern-day cybersecurity. Thus, before large-scale quantum computers are built, we need to migrate our systems and practices to ones that cannot be broken by quantum computers. For systems that aim to provide long-term confidentiality, this migration should happen even sooner.***
>
> *Cybersecurity in an era with quantum computers: will we be ready?*
> *Michele Mosca, November 2015*

Canadians rely on cryptographic systems to secure their applications and websites, and to protect the confidentiality and integrity of their data from domestic and global cyber threat actors. Quantum computers, when used by malicious actors, will be able to break many of today's cryptographic systems. To counter this threat, digital systems that process, store, or transmit sensitive or confidential information will need to be upgraded to use new "quantum-safe" Post-Quantum Cryptography (PQC).

Unfortunately, quantum-resistant solutions are not yet available. The U.S. National Institute of Standards and Technology (NIST) began work on new standards for PQC in 2015, and is currently on-track to publish new PQC standards for at least two digital signature algorithms and one key-encapsulation mechanism by the end of 2024.

> ***If your organization stores or communicates sensitive information, the use of post-quantum cryptography will be an inevitability in the next few years. To make this transition as smooth as possible, there are practical steps you can and should be taking to ensure your sensitive information remains secure both now and in the future.***
>
> *Forbes magazine, January 8, 2021*

The good news is there should be enough time for Canadian businesses and other organizations, including Critical Infrastructure (CI) owners and operators, to plan an orderly and cost-effective

transition to quantum-safe cryptography over the next few years, using the recommended practices and guidelines in this document.

1.1 Objective

The goals of this document are to provide a set of recommended practices and guidelines:

- to help Canadian Critical Infrastructure (CI) sector stakeholders and others to take actions now, to plan and prepare for how they will transition their digital systems to use new quantum-resistant cryptographic technologies and solutions; and
- to shorten learning curves by offering tangible advice and examples that illustrate "how to" undertake the recommendations made herein, so as to reduce the need for organizations to "start from scratch".

This document has been updated annually since 2021, to reflect industry feedback from implementing the best practices presented herein, and to provide additional examples of "how to" operationalize more of the strategic recommendations described in Section 3. The QRWG anticipates continuing to publish annual updates to this document for the foreseeable future.

1.2 The Quantum Threat

Asymmetric cryptography, or public-key cryptography, provides confidentiality and integrity for sensitive information. It is used extensively by the Government of Canada and by private sector organizations to secure and protect communications networks, cryptographic keys during their distribution, data at rest, and more. Most organizations currently rely on asymmetric public-key cryptography to secure:

- **digital signatures**: used to provide source authentication and integrity authentication as well as support the non-repudiation of messages, documents, or stored data;
- **identity authentication processes**: to establish an authenticated communication session or authorization to perform a particular action;
- **key transport of symmetric keys** (e.g., key-wrapping, data encryption, and message authentication keys) and other keying material (e.g., initialization vectors); and
- **privilege authorization processes**.

Security implications of quantum computing:

Current encryption protocols, such as Secure Socket Layer (SSL) and Transport Layer Security (TLS), based on existing public-key algorithms, are capable of protecting network communications from attacks by classical computers.

A fault-tolerant quantum computer, however, could break the mathematical challenges that underlie these and other protocols in a matter of hours or even seconds.

Deloitte Insights, April 2021

Asymmetric cryptography is based on the premise that two or more parties exchange public keys to establish a shared secret key to encrypt data. Symmetric cryptography on the other hand is based on the premise that all parties have already shared the exact same key prior to communicating.

Once developed, quantum computers will be able to use quantum physics to efficiently process information and solve problems that are impractical to solve using current computing technologies. Quantum computers will be able to compromise the algorithms used in asymmetric cryptography. This means that all classified, sensitive, and/or confidential information and communications that were protected using public-key cryptography, especially those having a medium to long-term intelligence value or commensurate need for long-term confidentiality, will be vulnerable to decryption by adversaries or business competitors that have quantum computers. [1]

1.3 Why Start Preparing Now?

The argument for starting now, to address the threat that quantum computers will pose to existing security systems, is based on the following considerations:

a) cryptographic technologies are integrated into most of the digital products commonly used by organizations to run their daily operations; [2]

b) some of the applications and systems used within energy, transportation, finance and government infrastructures have product lifetimes of 15 - 30 years, and even longer requirements for data protection and privacy;

c) fault-tolerant quantum computers, capable of breaking existing asymmetric encryption algorithms and cryptographic systems (e.g., public-key infrastructures), are widely expected to be available within the above timeline (e.g., by 2035); [3]

d) the time needed to migrate installed cryptographic technologies (e.g., SHA1) to newer standards can take many years; [4]

e) the number of cryptographic systems that organizations will need to migrate to use new "quantum-safe" cryptography will be large; and

---

1 Addressing the quantum computing threat to cryptography (ITSE.00.017), Canadian Centre for Cyber Security, May 2020

2 Using Encryption to Keep Your Sensitive Data Secure (ITSAP.40.016), Canadian Centre for Cyber Security, May 2021

3 National Security Memorandum on Promoting U.S. Leadership in Quantum Computing While Mitigating Risks to Vulnerable Cryptographic Systems (White House), May 4, 2022

4 The SHA1 hash function is now completely unsafe | Computerworld, February 2017

f) most organizations have no clear view of the cryptographic technologies used by their existing Information Management (IM), Information Technology (IT) and Operational Technology (OT) systems; this will make it difficult to discover and then prioritize the systems to be upgraded to post-quantum cryptography. [5]

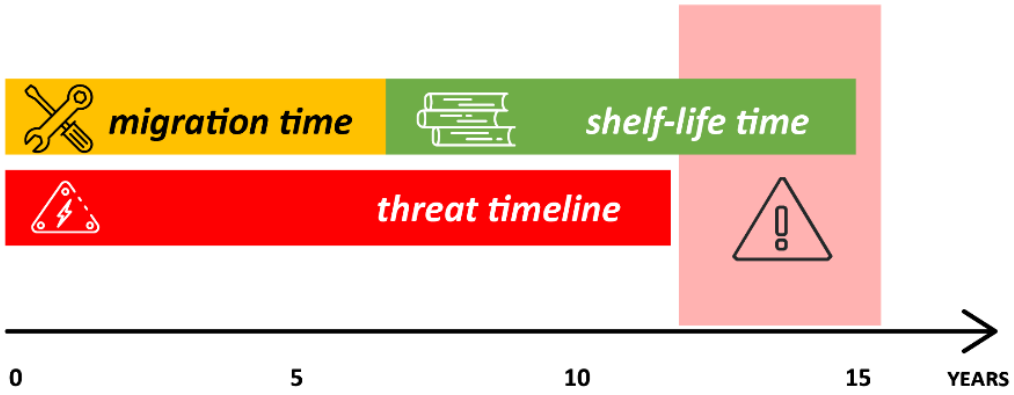
Migrating an organization's cryptographic systems to PQC will require significant effort. Organizations should begin planning now given that:

- the effort and time needed (e.g., to investigate, analyse, plan, procure, migrate, and validate new PQC) will not be small, and it will be different for every organization; and
- the amount of time remaining (until threat actors can access sufficiently powerful quantum computers to break existing cryptography) will decrease every day.

1.4 How much time is available?

The amount of time that an organization will have to transition its systems to use new quantum-safe cryptography (QSC) depends on three factors:

- the ***migration time***: the number of years the organization will need to migrate all of the systems that handle its important data to new quantum-safe cryptography;
- the ***shelf-life time***: the number of years that the organization's important, high-value information needs to be protected; and
- the ***threat timeline***: the number of years before relevant threat actors will be able to break the organization's existing, quantum-vulnerable, cryptography. [6]



[5] Post-Quantum Cryptography: Frequently Asked Questions, U.S. Department of Homeland Security (DHS), October 2021, 2 pages

[6] 2023 Quantum Threat Timeline Report, Global Risk Institute, 22 December 2023

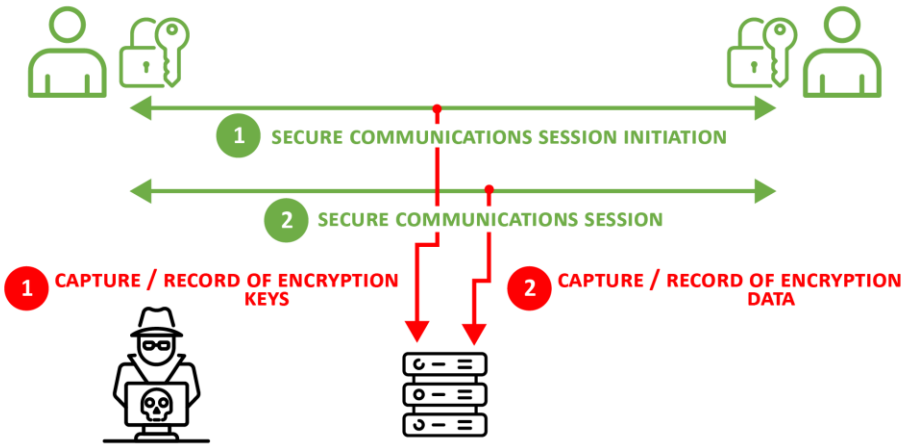
As illustrated on the previous page:

- organizations may need many years to migrate to Quantum-safe cryptography; and
- many organizations have important information (e.g., trade secrets, customer data, business plans) that they wish to keep confidential for a long time.

In the worst case, a threat actor will be able to use a quantum computer to break the encryption protecting important information before that data is protected by QSC.

Some threat actors (e.g., nation state level adversaries) are known to be harvesting copies of encrypted information today, and storing it for decryption in the future. Thus, any information that needs to be kept confidential for a long time (e.g., more than 10 years) may already be at risk of "harvest now, decrypt later" attacks. It must be noted that the shelf-life time for critical data and information such as trade secrets can be over 50 years.

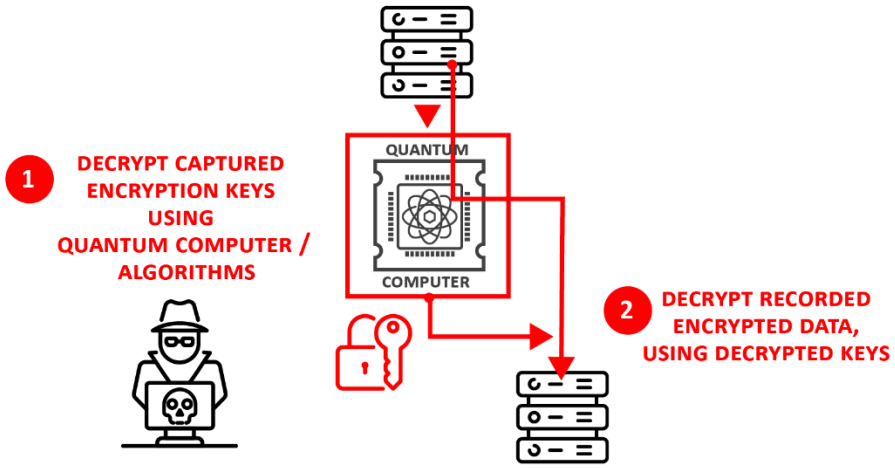
Today



	Users	Threat Actor
1	Secure communications session initiation includes crypto key exchanges	Capture / Record Encryption Keys
2	Encrypted session	Capture / Record encrypted data

Risk Mitigation Actions
Leverage information security and cyber security best practices and controls to help reduce current risks

In the best case, organizations that begin to assess their quantum-readiness now will have time to migrate their most important systems to use quantum-resistant cryptography before threat actors (and business competitors) obtain quantum computers. [7,8]

Future


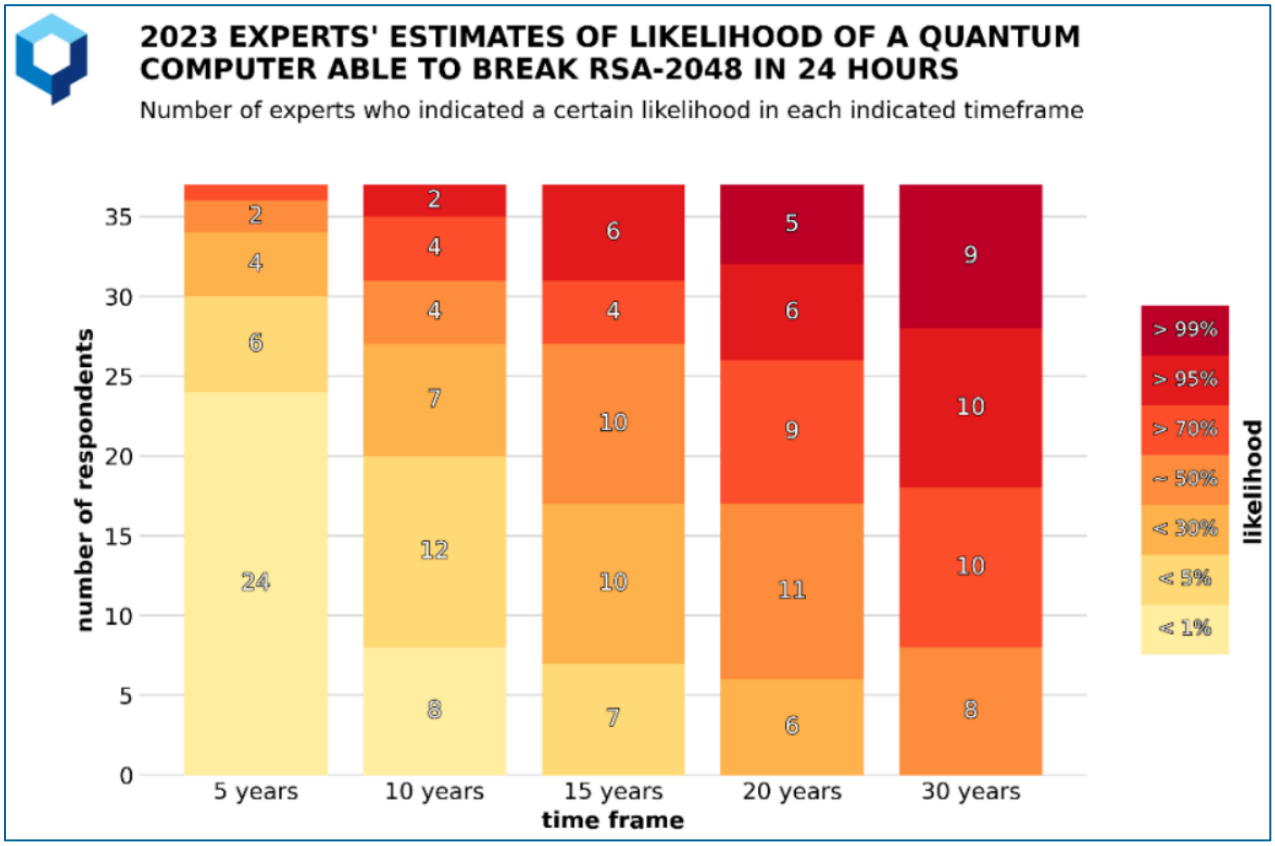
	Threat Actor
1	Captured encryption keys are decrypted using large scale quantum computers and quantum algorithms
2	Decrypted keys enable the decryption of captured encrypted data
Risk Mitigation Actions Immediate start of Quantum Readiness planning and actions to prepare for future quantum threats	

With respect to the threat timeline, the figure on the next page summarizes the latest opinions of 37 global quantum experts.

[7] The US is worried that hackers are stealing data today so quantum computers can crack it in a decade, MIT Technology Review, November 3, 2021, 5 pages

[8] The race to protect us from a quantum computer that can break any password (inews.co.uk), May 18, 2023, 9 pages

Every organization will need to review information such as this, and then decide on how much time they have, based on their own risk tolerance.



The opinions (of 37 experts from 15 countries) suggest that the quantum threat will become non-negligible relatively quickly and it could well become concrete sooner than many expect. For example, 20 out of 37 respondents felt the threat was more that 5% likely within a 10-year timeframe, while 17 respondents indicated a likelihood of 50% or more in the same timeframe.

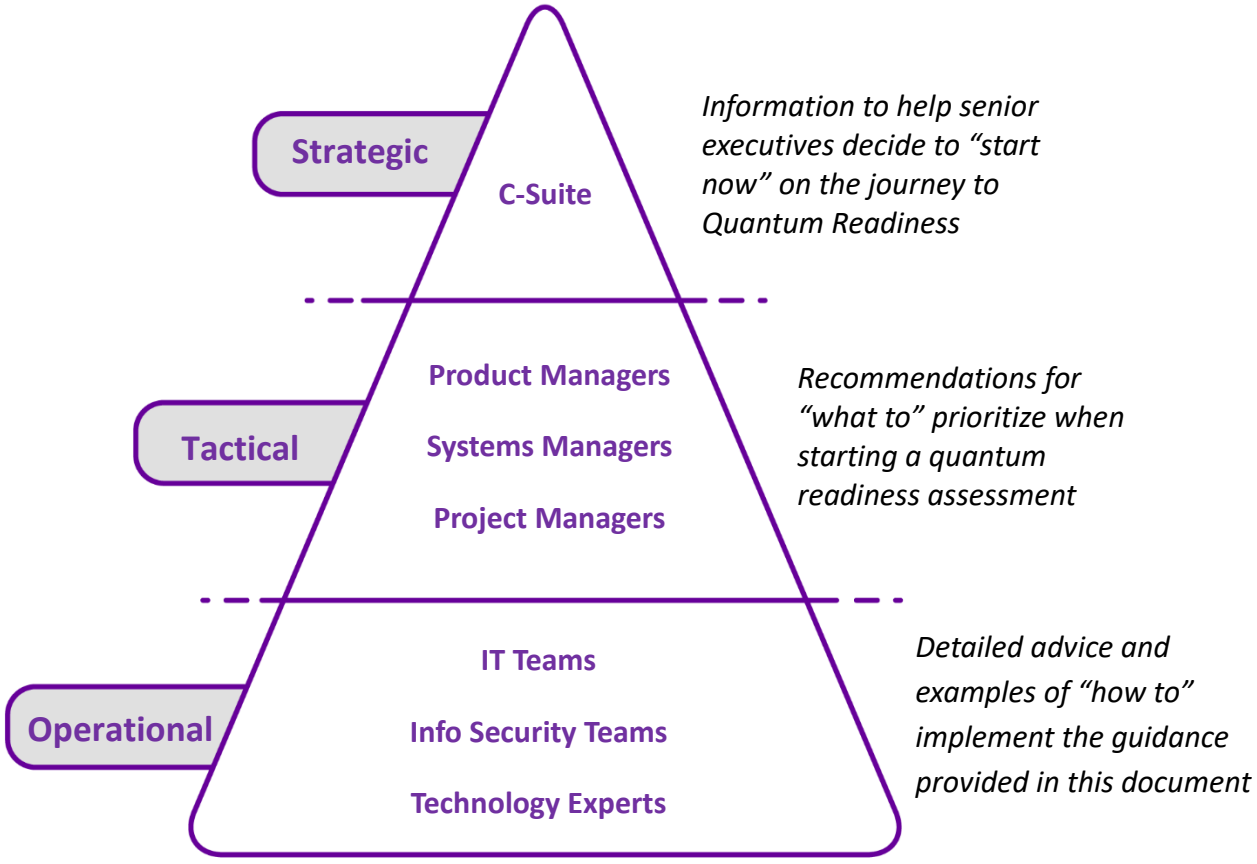
2023 Quantum Threat Timeline Report
Global Risk Institute, 22 December 2023

1.5 About this document

In July 2023, the Canadian Forum for Digital Infrastructure Resilience (CFDIR) rechartered its Quantum-Readiness Working Group (QRWG) to continue developing and updating its previously published best practices and guidelines for owners and operators of Critical Infrastructure (CI) systems. A series of discussions, discoveries and in-depth examinations were organized, in conjunction with stakeholders from Canada's Finance CI sector and the ICT vendor community, to refresh the guidance in this document on key considerations that C-suite executives, their direct reports, and their IM, IT, and OT staff will need to address to evolve their existing cryptographic systems to be "quantum ready" (i.e., quantum safe) in the coming years.

The information herein can be used and adapted by organizations as needed to inform decision makers on why and when to start their organization's journey to quantum readiness, and to provide guidance to operational staff on "how to" implement the recommended actions.

The contents of this document include strategic and tactical recommendations (in Sections 3 to 5), and operational advice (e.g., sample "how to" guides) in its Annexes and Appendices.



2. SOURCES OF INFORMATION

The sources of information used to formulate the practices and guidelines recommended in this document have been drawn from an extensive variety of sources in the public domain, and from discussions and deliberations within the CFDIR QRWG.

Primary sources include:

- Canadian Centre for Cyber Security (CCCS) publications;
- U.S. National Institute of Standards and Technology (NIST) Computer Security Resource Center Publications on Post-Quantum Security;
- European Telecommunications Standards Institute (ETSI) Quantum-Safe Cryptography working group documents; and
- Internet Engineering Task Force (IETF) Request For Comments (RFC) documents.

Where appropriate in later sections of this document, links to specific publications from the above sources may be identified as “normative references”. Normative documents are publications that must be read to understand or to implement the guidance being provided.

In contrast, some of the other sources highlighted in this document are referred to as “informative references”. Informative documents help the reader to develop a better understanding of a particular subject area.

Informative sources cited in this document include:

- Open source magazine articles, peer-reviewed papers and conference proceedings;
- World Economic Forum (WEF) and Global Risk Institute papers;
- Publications from international PQC-focussed working groups (e.g., FS-ISAC’s PQC WG, GSMA-IBM-Vodafone’s PQTN TF);
- Archived webcasts of expert panel discussions and presentations from PQC conferences (e.g., PKI Consortium’s November 2023 PQC Conference)
- Open source white papers, case studies, and application notes from private sector CFDIR member companies and other suppliers of ICT products or services involved in the supply-chain for “Quantum-safe” solutions; and
- Lists of software applications, libraries and hardware that includes support for Post Quantum Cryptography (e.g., PKIC).

3. RECOMMENDED QUANTUM-READINESS BEST PRACTICES

Executive leaders are encouraged to direct their organizations to start work now: [9]

- to understand the risks that quantum computing advancements will pose to their IM, IT and OT systems and data; and
- to plan how to manage the risks to their quantum-vulnerable systems by transitioning those systems and important data assets to introduce support for standardized quantum-resistant cryptography as early as 2025-2026.

Recommended actions that can be started now include the following steps: [10]

1. Educating your peers and your teams on the emerging quantum threat and the new technologies for quantum-safety including **hybrid cryptography** and **cryptographic agility**. [11, 12]

2. Evaluating the sensitivity of your organization's information assets and determining their lifespans to identify information that may be at risk (e.g., as part of ongoing risk assessment processes).

3. Inventorying the IM, IT and OT systems in your organization that use cryptography, and then implementing new policies and procedures in your change management activities to maintain this inventory on an on-going basis.

4. Asking the vendors of your cryptographic products if they support cryptographic agility, as well as when and how they will implement standardized and validated quantum-safe cryptography. [13]

5. Talking to your business partners and other third party suppliers about their current PQC posture and timelines for quantum-safety. [14]

---

[9] Getting Quantum Safe in 5 Slides – Executive Presentation, Cloud Security Alliance Quantum-Safe Security working group, February 2022

[10] Preparing Your Organization for The Quantum Threat to Cryptography (ITSAP.00.017), Canadian Centre for Cyber Security, February 2021

[11] Overview of Hybrid Cryptography, CFDIR QRWG, Annex H of this document

[12] Guidance on Becoming Cryptographically Agile (ITSAP.40.018), May 2022

[13] PQC Roadmap Questions to Ask Vendors, CFDIR QRWG, Appendix G of this document

[14] Questions to Assess the PQC posture of a 3rd party, CFDIR QRWG, Appendix E of this document

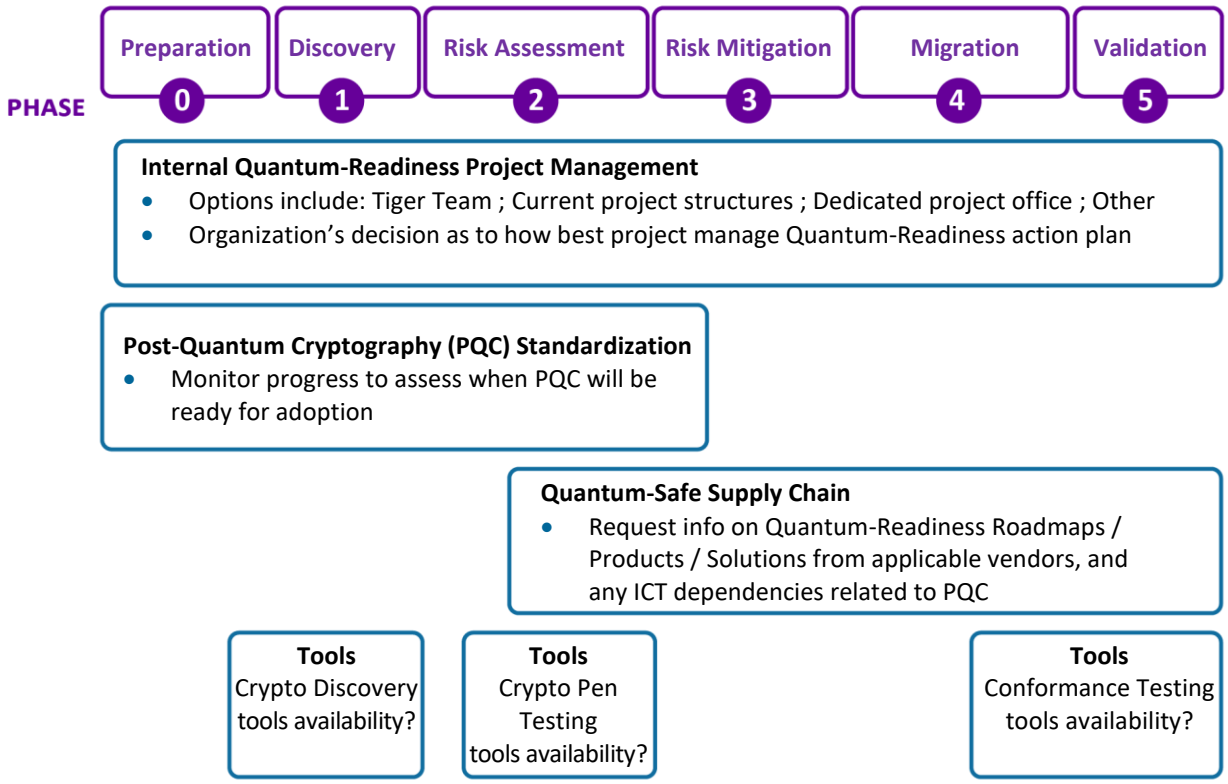
6. Budgeting for potentially significant software and hardware updates, as the timeframe for necessary replacement approaches.

7. Updating your IM, IT, and OT life-cycle management plans to explicitly describe how and when your organization will implement post-quantum cryptographic algorithms to protect your most important data and systems starting 2025 - 2026, or when validated cryptographic modules become available (e.g., one or two years later).

With respect to organizing these recommended actions into a Quantum-Readiness program, a multi-year and multi-phase timeline is recommended, as described below.

Quantum-Readiness Program Elements

Some Conceptual Building Blocks



While recognizing that every organization is unique and that no one size fits all, every Quantum-Readiness work plan should progress through the project ***Stages*** and ***Phases*** described on the next page.

- **Stage I: Initial Planning & Scoping**, managed as three distinct project phases that should be started before the first standards for new Post-Quantum Cryptography (PQC) are completed in 2024:

  Phase 0 - Preparation

  Phase 1 - Discovery

  Phase 2 - Quantum Risk Assessment

- **Stage II: Implementation**, starting in 2025, also consisting of three distinct phases:

  Phase 3 - Quantum Risk Mitigation

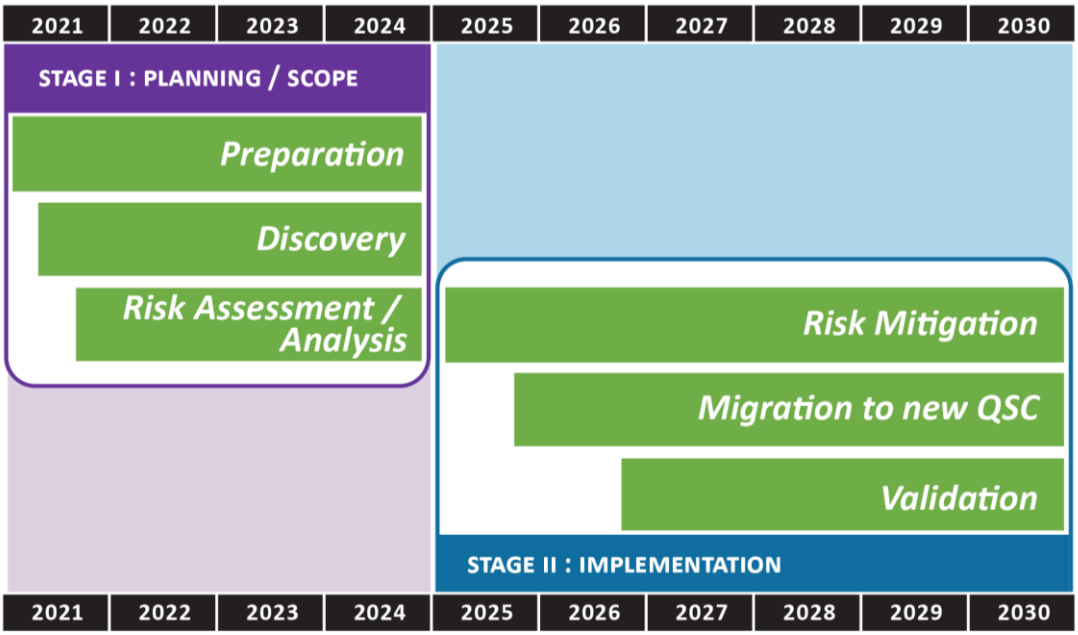
  Phase 4 - Migration to new QSC

  Phase 5 - Validation

The following timeline is recommended to set expectations with respect to the number of years that organizations may need to achieve full quantum-readiness using standardized PQC.

Quantum-Readiness Program Timeline

Recommendations as of July 2024



The anticipated duration (in years) for each Stage and Phase shown above reflects the current consensus of the CFDIR QRWG.

Sections 3.0 to 3.2 of this document recommend ***Planning and Scoping*** actions and best practices for the first three phases. They describe what an organization needs to start doing now

to prepare to their IM, IT, and OT systems for migration to new quantum-safe technologies post-2024.

Future versions of this document will offer additional guidance and recommended best practices for the post-2024 ***Implementation*** phases.

3.0 Phase 0 – Preparation

(RECOMMENDATIONS FOR C-SUITE EXECUTIVES)

1. Develop an understanding of the threats that quantum computing will pose for your ICT infrastructure in the coming years. Request a briefing within 6 months.

   Normative reference:

   - **NIST**: Cybersecurity White Paper - Getting Ready for PQC April 2021, 10 pages

   Informative references:

   - InfoSec Global Blog: The Time for Post-Quantum Readiness is Now, January 28, 2022
   - Cloud Security Alliance: Getting Quantum Safe in 5 Slides – Executive Presentation, February 2022

2. Ask one (or more) of your staff to form a team to investigate the scope of the effort that will be needed for your organization to start using standardized and new "quantum-resistant" cryptography in the coming years, and to identify which of your IM, IT and/or OT systems may need be remediated first.

   Normative reference:

   - **CCCS**: ITSAP.00.017 – Preparing Your Organization for the Quantum Threat to Cryptography, February 2021, 2 pages

   Informative references:

   - World Economic Forum: Transitioning to a Quantum-Secure Economy, September 2022, 35 pages
   - PKI Consortium Post-Quantum Cryptography Conference: Comparing Strategies for Quantum-Safe Cryptography Adoption in Organisations, November 7, 2023, 33 minute video replay
   - GSMA and IBM: The quantum clock is ticking – How quantum safe is your organization?, May 2024, 36 pages

3. Request periodic reporting on the progress of #2 (e.g., quarterly) and decide when to advance to Phase 1 (Discovery), as described in Section 3.1 of this document.

   Informative reference:

   - World Economic Forum: Is your business quantum-safe? 6 questions you should be asking, May 10, 2023, 7 pages

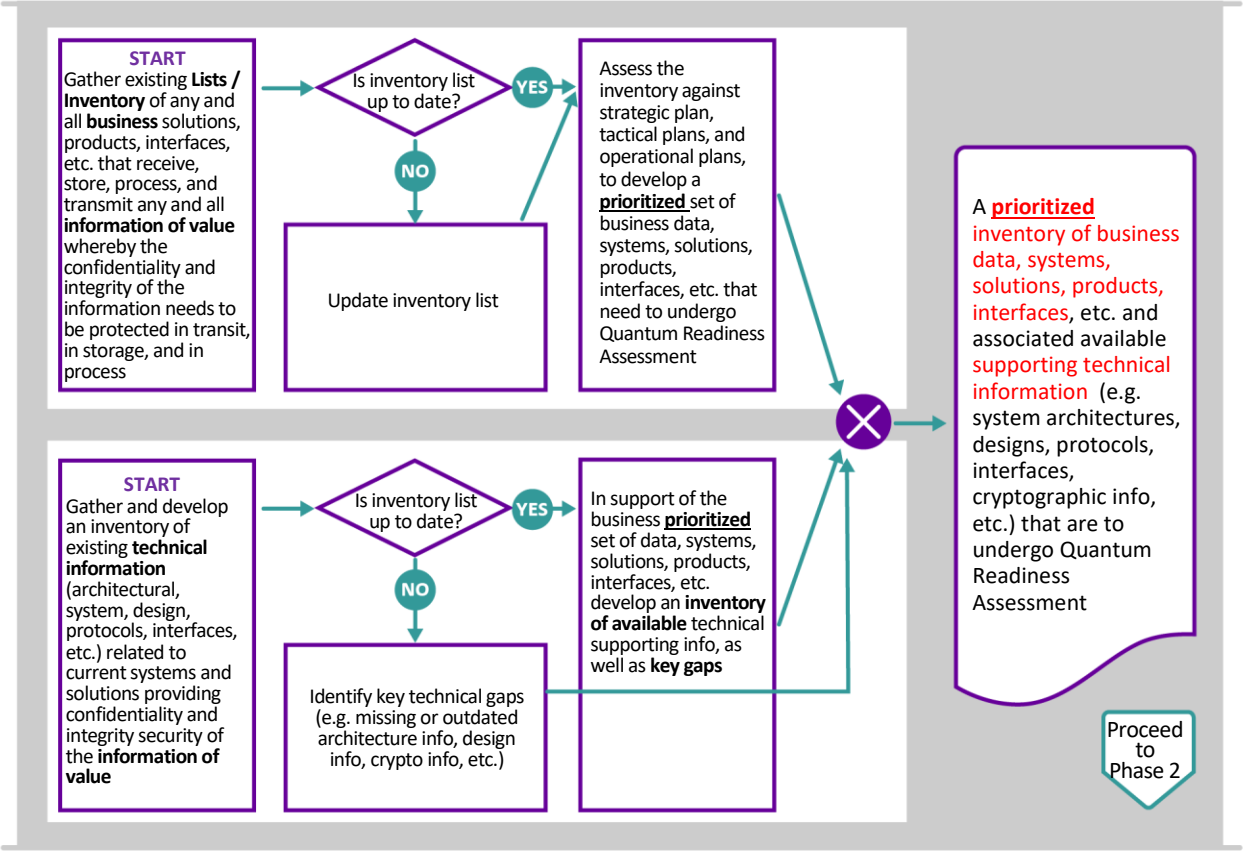
4. Email the CFDIR Secretariat with any questions on the above.

3.1 Phase 1 – Discovery

(RECOMMENDATIONS FOR C-SUITE EXECUTIVES AND THEIR DIRECT REPORTS)

5. Review the information to be collected during this phase, as illustrated below.
   - The goal is discover where and how cryptographic products, algorithms and protocols are used by your organization to protect the confidentiality and integrity of your organization's important data and digital systems.
   - The information collected during this phase will be needed to assess your organization's quantum risks in the next phase.

Phase 1 : Flow Chart : DISCOVERY



6. Appoint and empower someone to plan and execute a detailed discovery of where and how public-key cryptography is used by your organization.

   Normative reference:

   - **NIST**: Special Publication 1800-38B: Migration to Post Quantum Cryptography, Quantum Readiness: Cryptographic Discovery, December 2023, Preliminary Draft, 56 pages

Informative references:

- IBM Redbook: Chapter 2 - The journey to quantum protection, 19 July 2022, pages 15-26
- Thales Blog: Future-Proof Your Crypto Strategy for the Post-Quantum-Age: Insights from CNSA 2.0 and FIPS 140-3, June 27, 2024

7. Investigate whether using automated tools would facilitate your crypto discovery. Organizations should balance their security needs with their needs for usability and availability when considering such automated tools.

Informative reference:

- NIST: Migration to Post-Quantum Cryptography: Preparation for Considering the Implementation and Adoption of Quantum Safe Cryptography (Preliminary Draft); Volume A: Executive Summary, SP 1800-38A, April 2023 (updated May 2, 2023), 5 pages

8. Use the results from tasks #6 and #7 to build an inventory of where and how your organization uses public-key cryptography to protect its most important data and IM, IT and OT systems. Also identify any legacy cryptographic systems being used.

Normative reference:

- **CISA, NSA** and **NIST**: Quantum-Readiness: Migration to Post-Quantum Cryptography, 2023, Page 2

Informative references:

- FS-ISAC: Infrastructure Inventory Technical Paper, March 2023, 19 pages

9. Identify the important factors by which public-key cryptography affects the operation and security of your systems and applications (e.g., key sizes, latency and throughput limits, current key establishment protocols, how each cryptographic process is invoked, dependencies).

Normative references:

- **CFDIR QRWG**: Content Needed to Describe an Organization's Uses of Cryptography, Annex C of this document
- **CFDIR QRWG**: Matrix of Cryptography Use Cases, Annex G of this document

Informative reference:

- NIST: Getting Ready for Post-Quantum Cryptography, Cybersecurity White Paper, April 28, 2021, Page 5

10. Analyze the findings from #8 and #9 to develop a prioritized list of your organization's most important quantum-vulnerable systems that must be protected.

Informative reference:

- CCCS: ITSAP.00.017 – Preparing Your Organization for the Quantum Threat to Cryptography February 2021, 2 pages
- Engineering at Meta: Post-quantum readiness for TLS at Meta, May 22, 2024

3.2 Phase 2 – Quantum Risk Assessment

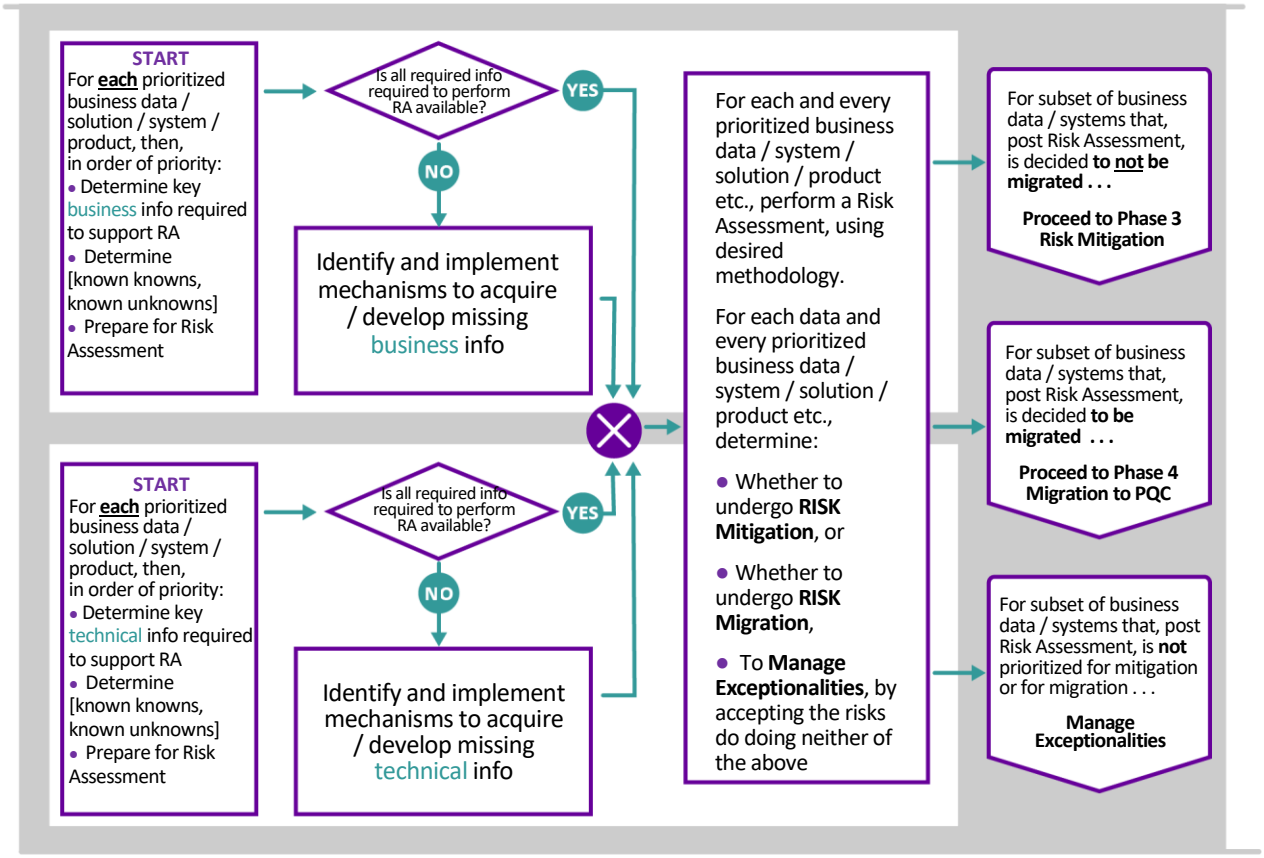
(RECOMMENDATIONS FOR IM, IT, OT MANAGERS AND THEIR DIRECT REPORTS)

11. Review the objectives of this Phase, as illustrated in the diagram below.

The objectives include:

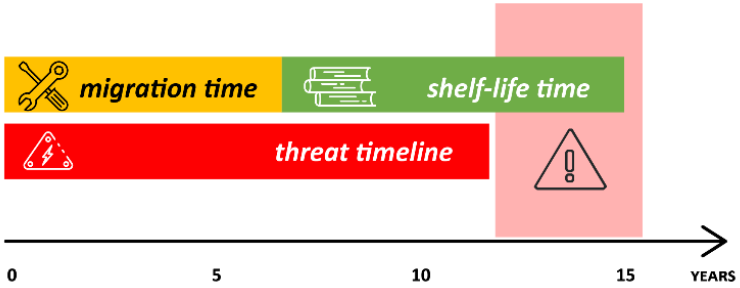
- Evaluating the sensitivity of your organization's information and determining its lifespan to identify the information that may be at risk (e.g. as part of ongoing risk assessment processes).
- Educating yourself and your teams on the threats that quantum computing will pose to your existing uses of cryptography.
- Asking your IM, IT and OT vendors and suppliers about their plans and timetables to implement quantum-resistant cryptography and crypto-agility, to understand any new hardware or software that will be needed.
- Reviewing your IT lifecycle management plans and budgeting for potentially significant software and hardware updates.

Phase 2 : Flow Chart : RISK ASSESSMENT (RA)



12. Start your Quantum Risk Assessment by reviewing the quantum risk equation introduced in Section 1.4, and the inventory of information discovered in Phase 1. That information is needed to determine the following variables for each of the digital systems that handle or store your organization's most sensitive information:

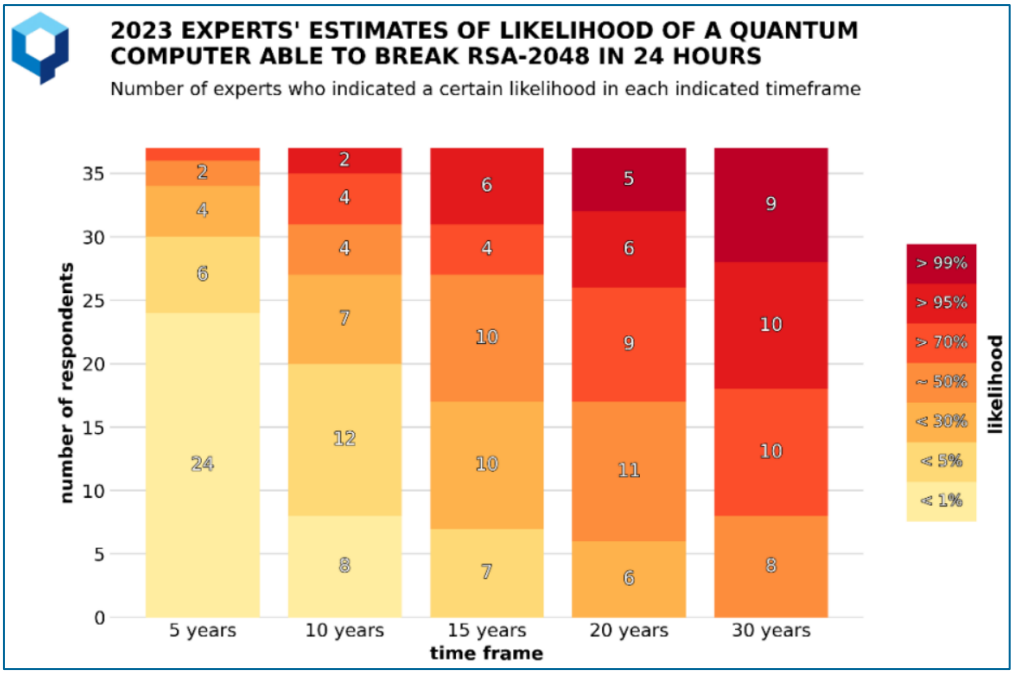
    - the ***shelf-life time*** (measured in years) that your most important data must be protected; and
    - the ***migration time*** *(also measured in years)* that your organization will need to upgrade the systems that handle your longest shelf-life data, to be quantum-safe.



Informative reference:

- Global Risk Institute: 2023 Quantum Threat Timeline Report, 22 December 2023, pages 8-9

13. Decide how the currently anticipated quantum ***threat timeline*** affects your organization's risk posture. To do this, review open source information such as the following and then determine your threat timeline based on your risk tolerance.



Normative reference:

- **Global Risk Institute**: 2023 Quantum Threat Timeline Report, 22 December 2023, page 19

14. Evaluate the sensitivity of your organization's information and determine its lifespan (i.e., the ***shelf-life time*** that your most important data must be protected) to identify information that may be at risk.

    Normative reference:

    - **CCCS**: ITSAP.00.017 – Preparing Your Organization for the Quantum Threat to Cryptography February 2021, Page 2

    Informative reference:

    - FS-ISAC : PQC Future State Technical Paper, March 2023, pages 13 and 14

15. Review your technology lifecycle management plans for each of the quantum-vulnerable systems identified in task #10 in Phase 1. Ask your IM, IT and OT vendors if their product development roadmaps include supporting crypto-agility and/or quantum-resistant cryptography in future updates. If yes, ask when those capabilities will be available.

    Normative references:

    - **CCCS**: ITSAP.40.018 - Guidance on Becoming Cryptographically Agile, May 2022, 2 pages
    - **CFDIR QRWG**: PQC Roadmap Questions to ask Vendors, Appendix G of this document
    - **CFDIR QRWG**: Questions to Assess the PQC Posture of a 3rd Party, Appendix E of this document

    Informative references:

    - CFDIR QRWG: Template to Catalog Technology Vendor/Supplier PQC Capabilities, Appendix F of this document
    - CFDIR QRWG: Crypto-Agility Exercise Notes, Annex I of this document
    - Accenture: The race to crypto-agility, 2021, 18 pages

16. Using the information from task #15, estimate the ***migration time*** *(measured in years)* that your organization will need to migrate each of the systems that handle your longest shelf-life data.

    Informative references:

    - CFDIR QRWG: Crypto-Agility Exercise Notes, Annex I of this document
    - CFDIR QRWG: Mock Migration to PQC Exercise Notes, Annex J of this document
    - NIST: Migration to Post-Quantum Cryptography - Project Description, August 2021, Pages 4-6

17. Prioritize the systems that will need the most urgent attention, by listing all of the systems that handle important data for which:

Migration Time + Shelf-life Time > Threat Timeline

Normative reference:

- **CFDIR QRWG**: Matrix of Cryptography Use Cases, Annex G of this document

Informative references:

- CFDIR QRWG: Mock Migration to PQC Exercise Notes, Annex J of this document
- FS-ISAC: PQC Future State Technical Paper, March 2023, pages 26, and 29 to 33

18. For each dataset, product, system, or solution flagged in #17, determine:

a) whether to undergo risk mitigation (per Phase 3), or

b) whether to start migration to PQC (per Phase 4), or

c) to manage exceptionalities, by accepting the quantum risk and doing neither of the above.

Informative references:

- FS-ISAC: Preparing for a Post-Quantum World by Managing Cryptographic Risk, March 2023, 7 pages
- FS-ISAC: PQC Future State Technical Paper, March 2023, pages 8 to 11, and 30
- TNO, CWI and AIVD: The PQC Migration Handbook: Guidelines for Migrating to Post-Quantum Cryptography, December 2023, pages 24 to 29

19. Also determine if your staff will need new training or additional resources (e.g., tools) to migrate your systems to use quantum-safe, post-quantum cryptography. If yes, the time needed to obtain those tools and/or training should be factored into the per-system migration time estimates developed during task #16.

Informative references:

- FS-ISAC: PQC Future State Technical Paper, March 2023, Pages 16 to 22
- CSIAC: Staying Ahead of the Curve: Planning for the Migration to Post Quantum Cryptography (youtube.com), Webinar replay, June 2024, 55 minutes
- UMBC: Post-Quantum Cryptography (PQC) Overview course
- CFDIR QRWG: Mock Migration to PQC Exercise Notes, Annex J of this document

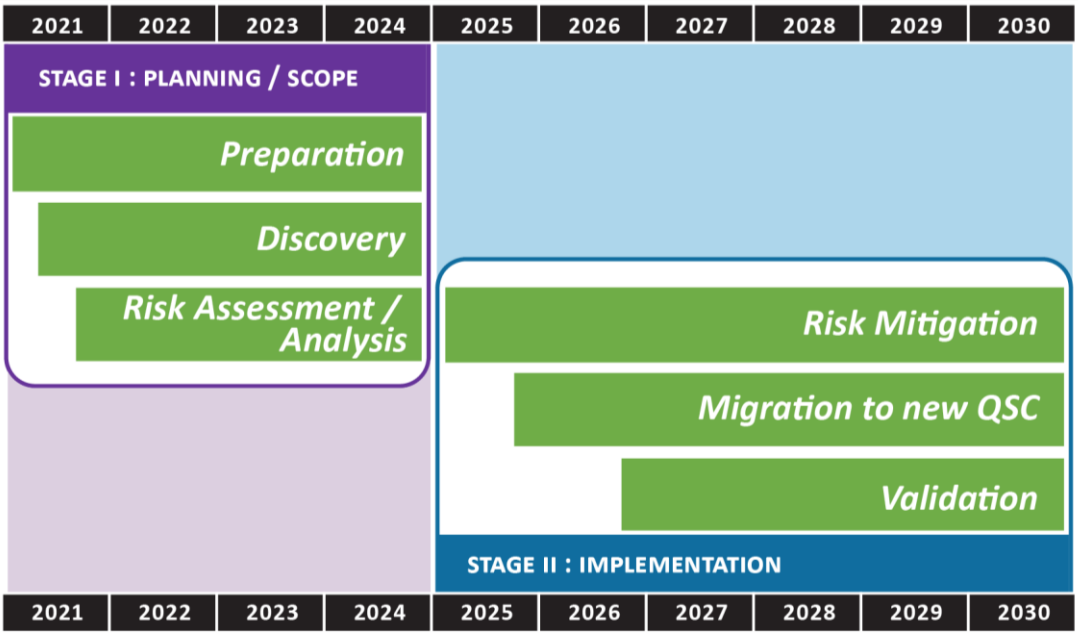
3.3 Stage II – Implementation (Phases 3, 4 and 5)

Future versions of this document will systematically offer an increasing amount of guidance and best practice recommendations for the three post-2024 ***Implementation*** phases, namely:

- Phase 3 - Quantum Risk Mitigation
- Phase 4 - Migration to new Quantum-Safe Cryptography (QSC)
- Phase 5 - Validation

Quantum-Readiness Program Timeline

Recommendations as of July 2024



To enable planners to begin preparing now, this document contains the following newly developed and/or updated Annexes and Appendices with guidance relevant to Stage II:

- Annex H contains a newly updated whitepaper on the topic Hybrid Cryptography. The QRWG anticipates that the use of standardized hybrid cryptography may help system owners to mitigate some of the risks of migrating to PQC;
- Annex I contains notes (developed during 2023) on a systematic approach to thinking about how and where to start migrating quantum-vulnerable IT systems to make use of quantum-safe cryptography by leveraging **Cryptographic-Agility**;
- Annex J contains brand new insights (developed during 2024) about the high-level activities and roles and responsibilities that technology providers, their customers, and

additional stakeholders will need to perform during the **migration to PQC**. The belief is that thinking through the exact steps of a migration using a systematic approach, such as the one described in Annex J, will enable system owners and operators to determine many different potential challenges that would have been previously unknown to them and inform their plans to migrate practical IT system use cases to use standardized PQC; and

- Appendix G of this document contains an initial set of **PQC Roadmap Questions**, developed in 2023, that system owners and operators can use today to ask when and how their technology providers will introduce PQC capabilities into their products and services.

4. AWARENESS AND SKILLS DEVELOPMENT

Creating an effective quantum risk awareness program will be important for every organization that uses cryptography, large or small, in the coming years.

The CFDIR QRWG developed a suite of slide decks to provide foundational building-block information and materials that can be used and adapted by organizations as needed to raise awareness and to inform decision makers and staff on why and how to begin their Quantum-Readiness journey. These decks may be obtained by emailing the CFDIR Secretariat.

	Contents & Focus	Pages	File Name	Version & Date
1	**Introduction & Context**	5	Quantum-Readiness-WG-Overview-v01	Version 01 July 7, 2021
2	**Master Chart Deck**	62	Quantum-Readiness-Best-Practices-Guidelines-v01	Version 01 July 7, 2021
3	Subset of Master Chart Deck Example #1 – **Executive Primer**	2	EX-01-Quantum-Readiness-Exec-Primer-v01	Version 01 July 7, 2021
4	Subset of Master Chart Deck Example #2 – **Executive Overview**	8	EX-02-Quantum-Readiness-Exec-Overview-v01	Version 01 July 7, 2021
5	Subset Example #3 – **Executive Overview with Backup slides**	34	EX-03-Quantum-Readiness-Exec-Overview-with-Backup-v01	Version 01 July 7, 2021
6	Subset Example #4 – **Detailed Overview for Managers**	32	EX-04-Quantum-Readiness-Mgmt-Overview-v01	Version 01 July 7, 2021
7	Subset Example #5 – **Detailed Overview for Managers with Backup slides**	60	EX-05-Quantum-Readiness-Mgmt-Overview-with-Backup-v01	Version 01 July 7, 2021

5. RECOMMENDATIONS FOR ENGAGING PQC VENDORS OR OTHER THIRD PARTIES

Solutions to transition to quantum-safe infrastructure are coming … IT and procurement teams must ask their current and prospective vendors for their quantum readiness plans to clarify who will handle which part of the transition and to ensure that investments that are being made today will position organizations towards quantum resilience.

A guide to a quantum-safe organization
U.S. Quantum Economic Development Consortium, December 2021

5.1 PQC Roadmap Questions to ask ICT Product or Service Vendors

It is recommended that system owners and operators start now to develop insights into the PQC roll-out plans of ICT vendors they depend on. [15]

Appendix G of this document contains eight "PQC Roadmap" questions that a system owner or operator could send to any technology product or service vendor today. These questions were developed by the QRWG during the spring of 2023 and then tested and verified to yield meaningful insights.

- The intent / focus is to provide system owners and operators with a way to start learning about the PQC product or service development plans of each of their technology vendors, in order to inform their own plans (and budgets) for migrating their systems to PQC.
- A secondary benefit (to all) may be that having more organizations asking their ICT vendors about their PQC Roadmaps will increase overall demand or "customer pull" for PQC solutions from vendors which may, in turn, accelerate the availability of PQC solutions.

[15] See task #15 in Section 3.2 of this document.

5.2 Recommended PQC Questions to ask Other 3rd Parties

Appendix E of this document contains a different series of questions to help system owners and operators to begin assessing the PQC maturity or 'posture' of any 3rd Party organizations they may do business with.

A 3rd Party in this context may be any supplier of products, goods or services (including ICT an non-ICT products/services), or any business partner or any customer.

- The intent/focus is to facilitate an evaluation of a 3rd Party's cryptography and PQC posture to assist the organization (that asks the questions in Appendix E) to determine their risk of doing business with the 3rd Party.
- This risk determination can and will vary in different organizations based on their risk tolerance associated to this topic.

5.3 PQC Procurement Clauses for RFI's and RFP's

Future versions of this document will offer guidance and best practices for this section.

6. CONCLUSION / KEY TAKEAWAYS

- Canadian businesses, organizations, and Critical Infrastructure owners and operators are advised to take action now, using the recommended practices and guidelines offered in this document, to begin planning an orderly and cost-effective transition to quantum-safe cryptography over the next few years to manage the risks that Quantum computers will pose to them.

Risks	
Cyber attack threat	• **Capture or 'Harvest' Now ; Replay and decrypt later ;** • **Data at Rest ; Data in Motion ;**
Key data at risk	• **Encryption keys ; PII ; Business "crown jewels" ;** • **Intellectual Property**
Risk scope	• **Organization ; Customers ; Supply Chain ; Ecosystems ; Dependencies/Interdependencies**
Perform Organizational Quantum-Readiness Risk Assessment to determine risk	

- Given that every organization is unique, there can be no "one-size-fits-all" approach.

- Quantum-Readiness planning should be started now because migrating an organization's quantum-vulnerable systems to use new quantum-safe PQC will be a multi-year process.

Cryptography	
Discovery	• **Key first step: Develop an accurate inventory of your organization's cryptographic usage across all of the products that depend on your digital systems**
Quantum-Readiness	• **Develop strategies, plans and budgets to upgrade or replace products and/or systems as needed for quantum-safety**
Crypto-Agility	• **One option to facilitate migrating existing cryptography to different or new crypto (e.g., standardized PQC)**
Organizations must prepare to upgrade / replace all cryptographic functions to standards-approved Post-Quantum Cryptography	

- Backward compatibility and interoperability between current and new cryptographic platforms, systems and solutions will be essential during the multi-year transition to Quantum-Safe Cryptography.
- Organizations should leverage all available information resources for the above, including but not limited to:
  - the recommendations presented in this document;
  - internal business and technical experts;
  - open source information; and
  - private sector Canadian and multi-national expertise and/or companies with experience and skills or products related to Quantum-Readiness.

Resources	
CFDIR Quantum-Readiness Working Group (QRWG)	• **Quantum-Readiness Best Practices and Guidelines**
Canadian Centre for Cyber Security	• **Open-source publications, including cryptographic guidance, alerts and advisories**
Canadian as well as global resources available to help guide organizations prepare for Quantum-Readiness	

ANNEX A: GLOSSARY

- CA - Certificate Authority
- CCCS - Canadian Centre for Cyber Security
- CFDIR - Canadian Forum for Digital Infrastructure Resilience
- CI - Critical Infrastructure
- DECT - Digital Enhanced Cordless Telecommunications
- ENISA - European Union agency for Cybersecurity
- FIPS - (U.S.) Federal Information Processing Standards
- HSM - Hardware Security Module
- IETF - Internet Engineering Task Force
- IKE - Internet Key Exchange
- IM - Information Management
- IPsec - Internet Protocol Security
- IoT - Internet of Things
- ISO - International Organization for Standardization
- IT - Information Technology
- Kerberos - Computer network authentication protocol to allow server communication over a non-secure network
- LDAPS - Lightweight Directory Access Protocol
- MFA - Multi-Factor Authentication
- mTLS - Mutual Transport Layer Security authentication
- NCCoE - (U.S.) National Cybersecurity Center of Excellence
- NIST - (U.S.) National Institute of Standards and Technology
- OAuth - Open standard for access delegation
- OT - Operational Technology
- PGP - Pretty Good Privacy
- PII - Personally Identifiable Information
- PKI - Public-Key Infrastructure
- PQC - Post-Quantum Cryptography
- QRWG - Quantum-Readiness Working Group
- QSC - Quantum-Safe Cryptography
- RA - Registration Authority

- S/MIME - Secure/Multipurpose Internet Mail Extensions
- SAML - Security Assertion Markup Language
- sFTP - SSH File Transfer Protocol
- SHA1 - Secure Hashing Algorithm version 1
- SSH - Secure Shell
- TLS - Transport Layer Security
- TLP - Traffic Light Protocol

ANNEX B: RECOMMENDED CRYPTOGRAPHY USE CASES TO BE DISCOVERED & DOCUMENTED

This Annex contains a list of technology protocols and broader IM / IT cryptography use-cases applicable to most public and private organizations and businesses across Canada.

Common Protocols:

1) TLS
2) mTLS
3) sFTP
4) FTPS
5) SSH
6) SAML
7) OAuth / OpenID Connect
8) IPsec
9) IKE
10) DMARC
11) DKIM
12) SPF
13) Kerberos
14) LDAPS
15) PGP
16) EAP-TLS
17) WPA (WiFi)
18) S/MIME
19) DECT
20) Mobile NEC
21) DNSSEC
22) DOT / DOH
23) MACsec

Broader Cryptography Use-case Considerations:

A. Code Signing
B. Multi-Factor Authentication (MFA)
C. Encryption of Data at Rest – may be vendor-specific
D. Cloud Native Encryption
E. Hardware Security Modules (HSMs)
F. Certificate Authorities (CAs)
G. Application Layer Payload Encryption

ANNEX C: CONTENT NEEDED TO DESCRIBE AN ORGANIZATION'S USES OF CRYPTOGRAPHY

This Annex provides an initial list of the information to be sought and then collated when an organization is ready to inventory the cryptography it relies on for any of the use cases listed in Annex B. This information is appropriate to develop during Phase 1 - Discovery.

The content to be inventoried per items 1 to 10 (below) will describe "how things currently are" in one or more of an organization's existing IM, IT and/or OT systems.

1. Use Case Description
2. Business Value
3. Potential Business Data in Scope / Volume of that Data / Lifespan of that Data
4. Use-case Class (e.g., Data in Transit, Data at Rest, Data in Processing, Digital Signature)
5. Technical and Threat Considerations
6. Types of Cryptography Currently in Use
7. Technical Components (e.g., end-points, networks, databases, file servers)
8. Locations where Cryptographic Information Exists (e.g., DLL, hardware)
9. Technical Dependencies (e.g., details on components within this Use Case that depend or rely on other systems for their own security)
10. Ability to Support (Pre and Post-Quantum) Cryptographic Algorithms Simultaneously

After the above information is collected, analyzing it will enable planning "What to do to reduce the quantum risk?" in later project phases (e.g., Quantum Risk Assessment, Quantum Risk Mitigation, Migration to Quantum-safe PQC), including:

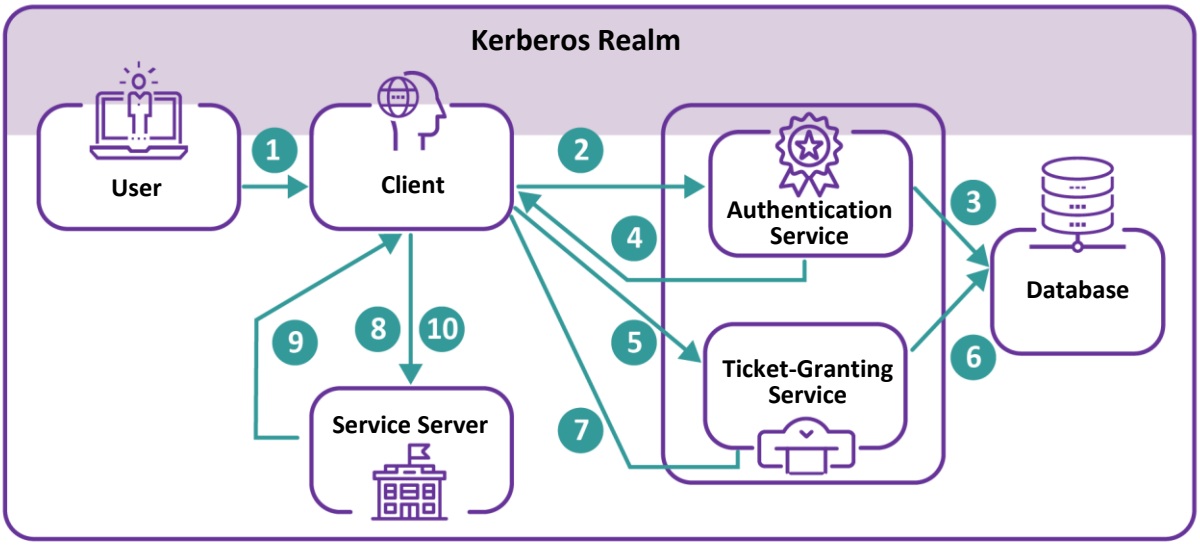
11. Best Choice of Algorithm to Use
12. Order or Sequence of what needs to be Upgraded
13. Path To Inline Quantum Remediation
14. Alternate Paths to Quantum Remediation (e.g., upgrade of entire system, change in paradigm)

ANNEX D: SAMPLE USE CASE #1 - USING KERBEROS FOR AUTHENTICATION

Section 1: Use Case Description

Kerberos is an authentication protocol on computer networks that allows clients to access services from providers. It does so by leveraging a Ticket-Granting Service (TGS) from a Key Distribution Centre (KDC) which will provide tickets to the service requestor to give to the service provider for access. It is often used as a main ingredient in Single-Sign-On (SSO) functionality.

A generic diagram of the network architecture in which Kerberos is used is given here.



1) User enters credentials (username + password).
2) Send KRB_AS_REQ.
3) Lookup user (and password) in database.
4) Send KRB_AS_RSP.
5) Send KRB_TGS_REQ.
6) Lookup service (and password) in database.
7) Send KRB_TGS_RSP.
8) Send KRB_AP_REQ.
9) Send KRB_SP_RSP.
10) Send service request to Service Server.

It should be mentioned that the initial contact and authorization of the client may occur over an insecure channel and, therefore, require some protection such as TLS. This channel is outside the scope of this use case.

Section 2: Business Value

Kerberos is mainly used to grant users and machines access to different services. It is often a critical ingredient in SSO implementations. Kerberos is also one of the basis elements of Microsoft Active Directory (AD).

Section 3: Potential Business Data in Scope/Volume/Lifespan

The data used by Kerberos is often limited to user and/or machine access data or data regarding the service being accessed. This would include userIDs and passwords, IP addresses, and potentially other limited-use and transitional information. Most of the information is of limited use and there is a limited time it would be available.

The data that is available to be accessed due to compromise of Kerberos would be unlimited as it theoretically can be used to access any service. However, this would be within the scope of the service being accessed and not directly tied to the Kerberos implementation.

Section 4: Use-Case Class

Identity Management and Access Control

Section 5a: Technical Considerations

The following are considerations for Kerberos with regard to implementing quantum-safe technology:

1) **Availability:** A system implementing Kerberos will often be accessed by many different users and services at the same time. There is always a Denial-of-Service (DOS) risk in any change.
2) **Compatibility:** Kerberos can be used by m any different services, each with its own coding. Any change would have to be one in a way which is compatible with the services that use it.
3) **Credential Management:** Kerberos does manage credential from users and services in order to properly authenticate them. Changes should not put these at risk.

Kerberos is often embedded into other products. Most organizations would be dependent on having their vendors make Kerberos be quantum-safe. However, individual organizations would need to track and test in order to ensure that any changes would not be disruptive.

Section 5b: Threat Considerations

Kerberos implementations often serve as the central access point for user interaction to services within an organization. Compromise of the Kerberos system can range from a limited one-time service access to complete, catastrophic access control failure.

It would be a target both for malicious insiders as well as external attackers.

There exist current classical attacks on Kerberos (e.g., pass-the-hash).

Section 6: Types of Cryptography

Kerberos is traditionally based on symmetric key cryptography and so is not especially vulnerable to quantum. However, there do exist extensions where asymmetric cryptography is used for initial authentication (e.g., IETF RFC 4556, RFC 8062 and RFC 8636).

There are two instances where asymmetric cryptography can be used in Kerberos:

1) **User Authentication:** Classical Kerberos will verify users through traditional access control methods such as a userID and password. However, the public key extension for Kerberos allows a user to send a client certificate which can be verified by a trusted CA.
2) **Session Key Agreement:** Classical Kerberos will use user information (e.g. password) to compute a session key between the client and Key Distribution Centre for encryption purposes. The public key extension allows asymmetric key agreement such as Diffie-Hellman.

Section 7: Technical Components

The main technical components of Kerberos are:

1) **Client (Service Requestor):** the user or machine that is requesting the service.
2) **Service Provider:** the service that is being accessed.
3) **Client Authenticator:** The entity responsible for authenticating the client. This is often embedded within the KDC.
4) **Ticket-Granting Service (TGS):** The service which will grant a ticket to the client which will allow it to access the service. This is often a part of the KDC.
5) **Certificate Authority (CA):** This optional for the extensions which rely upon a CA to verify client certificates.

Domain controllers are an example of a KDC as they often implement the Kerberos protocol.

The network over which communication will take place can also be considered to be a component. However, as Kerberos is not a network protocol, this is considered out of the scope of this use case.

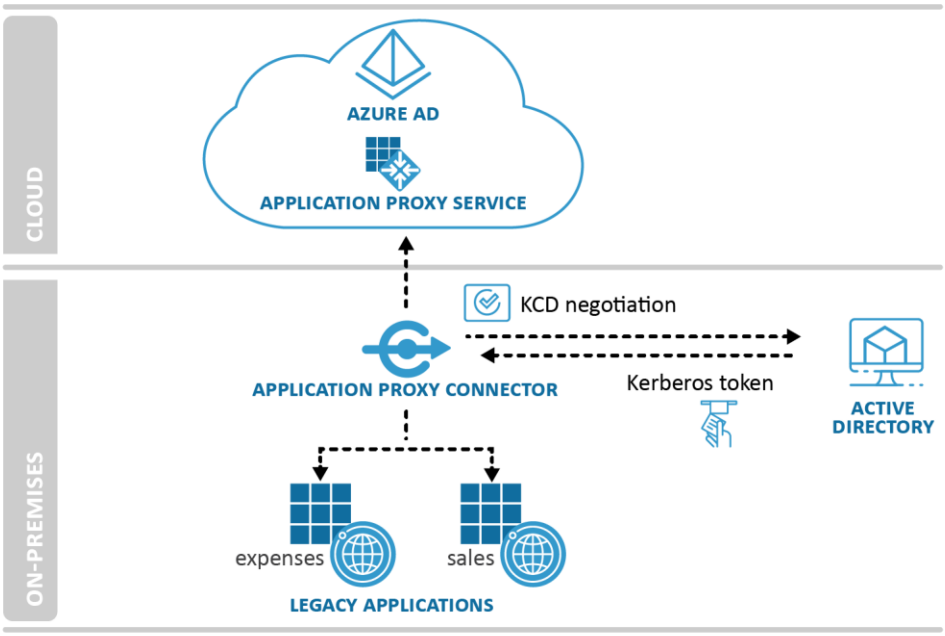
The Client Authenticator and TGS form the heart of the Kerberos system, often within the KDC. The client and Service Provider are separate systems which must be compatible with Kerberos KDC in order to function properly.

Section 8: Crypto Locations

A Kerberos implementation (i.e., the KDC) is usually a centralized system with its own cryptographic code and/or libraries. Its exact location would be product-specific. It must also be able to access a proper CA to verify a client certificate when used for initial authentication extension.

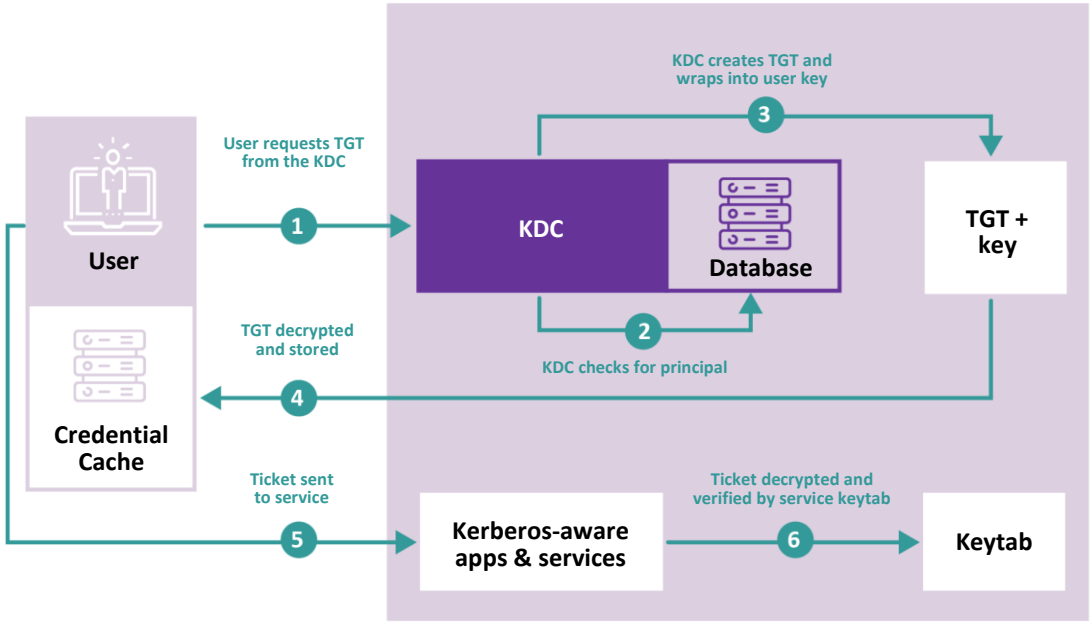
Note that asymmetric keys used within the KDC are ephemeral and so do not need to be stored for any length of time. Client certificates are used only for initial authentication and can then be discarded whereas the asymmetric keys used for key agreement can be discarded once the symmetric key is established.

The client and service provider would have their own method and location of cryptography. This, again, would be very implementation-dependent. The client would need to store the private key for its certificate. However, the asymmetric keys needed for key agreement would be ephemeral and would not need to be stored.



The most popular implementation of Kerberos is within Microsoft Active Directory (AD). An example in Azure AD is diagramed above.

Kerberos is also implemented by Red Hat. The following diagram shows its structure.



Section 9: Dependencies

The dependent use cases for Kerberos are:

- Data Storage – for client private keys (if certificates were used to establish authenticity of public keys)
- PKI/CA – (if certificates were used to establish authenticity of client public keys)
- TLS – to protect the initial client authentication.

Section 10: Ability to Support Algorithms Simultaneously

The main entity which would be required to support algorithms simultaneously would be the KDC. It would need to simultaneously authenticate quantum-safe and non-quantum-safe client public key authentication requests.

If the KDC can support both simultaneously, then it would make sense that it would be upgraded first. The client and service provider would need to support whichever version of the protocol the KDC has implemented. Hence, these can gradually be upgraded at their own pace after the KDC. These upgrades would be independent of each other.

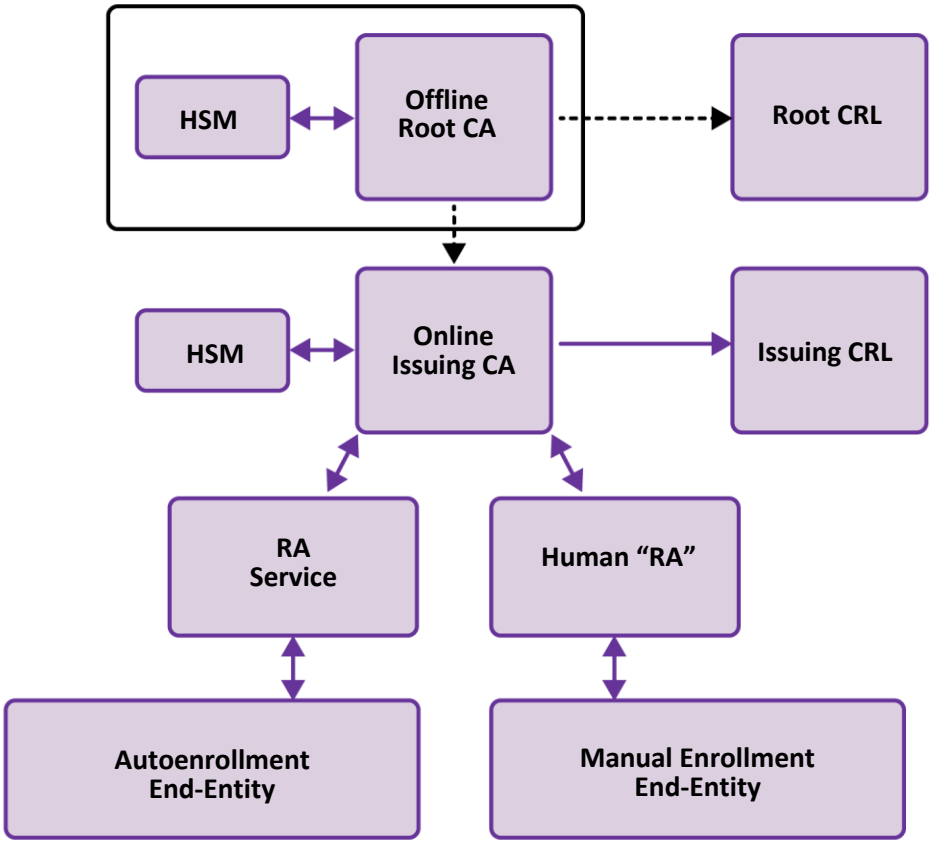
ANNEX E: SAMPLE USE CASE #2 - PKI/CAs

Section 1: Use Case Description

The purpose of a Public-Key Infrastructure (PKI) is to provide the technology and processes to leverage certificates for various other use cases such as TLS, sFTP, IPSec, and many others.

This is accomplished through the use of a Certificate Authority (CA) which has the ability to issue certificates which relying parties can use to authenticate individual entities. The certificates leverage public-key cryptography for authentication which makes it inherently susceptible to quantum computing.

A Certificate Authority will typically have a hierarchy such as shown in the diagram below:



CAs may have more or fewer levels, but they have the same basic structure.

In terms of scope, this use case will cover only the CA structure itself. It will not cover the use of certificates in such protocols as TLS and sFTP as those will be covered in their own separate use cases.

The CA/PKI use case will be separated into several sub-use-cases:

1. Public CAs – CAs which issue publicly or universally trusted certificates (e.g., Entrust, DigiCert)
2. On-Premises Internal CAs – CAs established within and managed by an internal organization (e.g., Microsoft PKI, KeyFactor)
3. Managed Internal CAs – CAs which are trusted only by an internal organization but managed by an external entity
4. Special Purpose CAs – CAs which are typically application specific within a well-defined domain (e.g., IoT CAs for mobile devices)
5. Inspection CAs – CAs which are used to intercept traffic in a Man-In-The-Middle scenario and inspect content (e.g., web content filtering and TLS inspection)

While similar each have their own characteristics which will be called out where different.

Section 2: Business Value

PKIs are typically classified as technology infrastructure. Its business value lies in its position as a key element in the operational security of critical operations. Thus, it would essentially inherit the business value of whatever application would depend upon it. As most applications which make use of a network require some level of security, PKIs are ubiquitous within most high- and low-value applications.

Section 3: Potential Business Data in Scope/Volume/Lifespan

While PKIs are involved in the protection of business data, they do not typically directly protect business data. This is often left to end-entity certificates within use cases such as TLS, sFTP, etc. This would be out of scope for this use case.

Furthermore, CA certificates are typically used for signing, not encryption or key agreement. Hence, there is no harvest-and-decrypt risk for CA certificates.

The only data present within a PKI would be infrastructure data such as Fully Qualified Domain Names (FQDNs) or routing information. With the advent of Certificate Transparency (CT), much of this information is now publicly available. Hence, it is most important to protect this information from an integrity and authenticity perspective.

Section 4: Use-Case Class

Entity Authentication for Critical Infrastructure

Section 5a: Technical Considerations

The following are considerations for PKI with regard to implementing quantum-safe technology:

1) **Certificate Size:** Applications may have limitations on size such as through-the-device or channel constraints or hard-coding of buffer sizes.
2) **Signing Performance:** Some applications require a high throughput CA for large volume or high-speed signing capabilities. The Inspection CAs are a good example as they must create new certificates on-the-fly with little to no noticeable impact to user browsing.
3) **Verification Performance:** Some applications such as IoT or high-volume servers may have restrictions on verification performance as devices may be constrained or deal with large amounts of verifications.

Note that technical considerations of CA chain verification for applications is not in scope as it would be covered in the use cases using the certificates.

Section 5b: Threat Considerations

The CA is often the central root of trust for a large number of systems. Compromise of a CA private key could lead to a large amount of fraudulent certificates and connections and, hence, unauthorized transactions. The potential fraud is directly attributable to the capabilities of the applications leveraging these certificates.

The following would be further considerations for each separate use case:

1) Public CAs are universally accepted, so compromise could be catastrophic and worldwide.
2) On-premises CAs would have affects typically only for the organization. As it is hosted internally, it would likely require access to the organization's internal network to determine the CA certificates and to conduct malicious activity.
3) Managed CAs would be similar to on-premises CAs in that access to the organization is required to conduct fraud. There is an additional threat vector in that compromise of an managed CA provider could compromise many different organizations.
4) Special purpose CAs would be specific to the application they are dedicated to. One of the threat considerations would be discovering these CAs. Quite often, these CAs are embedded within products and agnostic to users and administrators.
5) Inspection CAs would be similar to on-premises CAs except that compromise would likely be limited to browser-based applications accessed by internal users.

Section 6: Types of Cryptography

The cryptography is asymmetric mainly used in:

1) Signing of CA intermediate certificates.
2) Signing of end-entity certificates

3) Signing of Certificate Revocation Lists (CRLs)
4) Authentication of Registration Authority credentials

Note that root certificates are self-signed. However, the signing is often of little value as applications will accept a root if it simply exists within its root store.

The certificates also make use of a hash function within signing and for thumbprint purposes.

The PKI will also make use of random number generation in order to generate public/private key pairs and produce signatures.

Section 7: Technical Components

The technical components in implementing the CA depends on the type of CA being implemented. Several types of CAs are listed here:

A) Root CA
Root CAs are typically held offline and is only used for signing intermediate CAs and the corresponding root CRLs. The components typically consist of:
- Offline Hardware Security Module (HSM) and related peripherals
- Offline machine to facilitate signing (e.g. laptop, desktop, some sort of device)
- Software to facilitate CA functions
- Offline secure storage device to store private key information

B) Intermediate CAs (Networked)
The intermediate CAs are typically used for issuing certificates
- Online networked HSM and related peripherals
- Online server, virtual machine, or equivalent
- Software to facilitate issuing CA functions such as:
  - Certificate Signing Request (CSR) validation and signing
  - OCSP or equivalent compatibility
  - CRL generation and signing
  - RA credential verification
  - Public/private key pair generation (for some use cases where the CA generates and end entity's certificate)
- Online accessible file lookup for CRL
- Access control functionality
- Backup systems to store log and data

C) RAs (either manual or automated)
RAs would need the technical capability to accept certificate requests and perform verification of the request and validation of the entity. This would typically consist of:
- A machine (e.g. laptop, server) to run the RA software
- A portal or Access Control List (ACL) to provide information to validate
- RA credentials (usually an RA certificate)

D) Inspection CAs
Inspection CAs would usually be embedded within an appliance of some sort and have their own protection capabilities for the private key such as an onboard crypt card.

E) Special Purpose CAs
The components of a special purpose CA would be dependent upon the type of application it is used for. For example, such a CA to handle registration of surveillance cameras would have very different components than one for conferencing software. However, there would be at minimum:

- A machine to handle registration, signing, and issuance of the special purpose certificates.

F) End Entities
While end entities are generally out of the scope of this use case, we will include specifically the end entity function of generating a CSR and installing a certificate. In order to do so, the end entity components would be:

- The end entity itself
- The software used to generate the CSR and install the certificate.
- The storage location of the private key as well as any related protection mechanisms.

Section 8: Crypto Locations

1) Root and Intermediate CAs
The primary location of the cryptography in play would be within the HSMs. This would be heavily dependent upon the type of HSM and manufacturer. There may be some residual crypto functionality from the software which is meant to facilitate CA functionality or to perform OCSP signing and RA credential verification.

2) RAs
For RAs, this would likely be the software which facilitates RA login.

3) Inspection CAs
Inspection CAs would mostly rely on the crypto card that they use for certificate generation as well as the corresponding software. This is usually packaged together within an appliance.

4) Special Purpose CAs
This would be completely dependent upon the implementation and would be vendor-specific.

5) End Entities
This would be embedded within the CSR generation software on the entity such as OpenSSL.

For the majority of the use cases, there is typically no CA cryptography outside of the HSM. Crypto for the HSMs is handled in the HSM use case.

When a software-only implementation is used, the private keys are typically stored locally on the machine which is performing the signing. The code is embedded in the software product that is being used.

In terms of generating private keys and CSRs, one standard implementation is OpenSSL's req command-line utility.[16] The requisite code is within the OpenSSL binaries and the keys and CSRs are output to a file specified in the command line.

Section 9: Dependencies

The following use cases are dependencies for this one:

- Data Storage
- HSMs

In addition, certain considerations from other use cases may need to be taken into account from other use cases for which this use case is a dependency in order to ensure compatibility.

Section 10: Ability to Support Algorithms Simultaneously

Proposals exist for combining quantum-safe technology with existing methods to support both as a hybrid as listed here:

- RFC 9481 - Certificate Management Protocol (CMP) Algorithms, IETF, November 2023, 28 pages
- draft-ietf-lamps-pq-composite-sigs-02 - Composite ML-DSA for use in Internet PKI, IETF, July 2024, 50 pages

Thus the remaining work would be in getting the CA and end-entity components to implement them. The HSMs and all CA software must be able to support this. Applications would need to support as well.

[16] How to use the command 'openssl req' (with examples), November 5, 2023

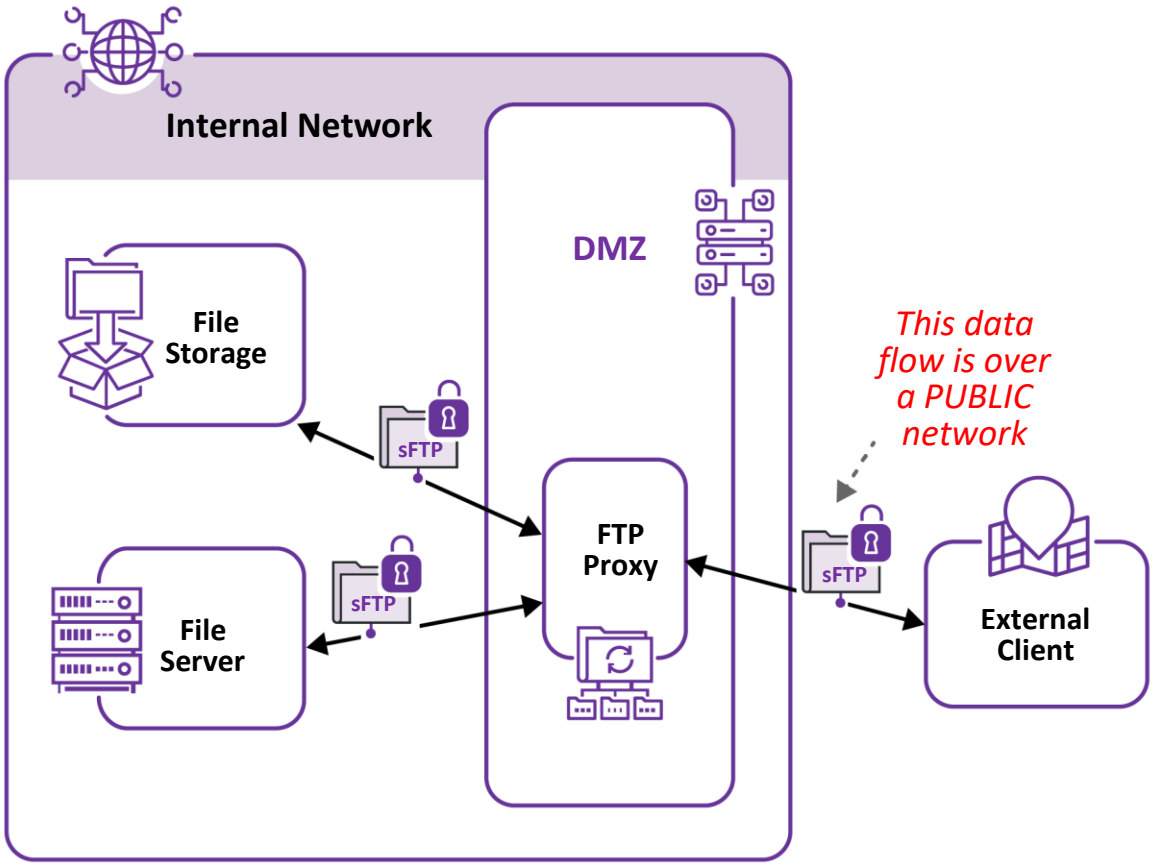
ANNEX F: SAMPLE USE CASE #3 - sFTP

Section 1: Use Case Description

sFTP, the Secure File Transfer Protocol (not to be confused with the Simple File Transfer Protocol) is a network protocol that leverages SSH authentication to securely transmit and manage files between two endpoints.

The SSH protocol is actually its own use case and so will not be considered in generality. However, for scoping purposes, as sFTP is widely used and has high business value, this use case will consider the use of SSH as bound to sFTP protocol and so will be considered one and the same. A separate SSH use case will be created for non-sFTP uses.

A generic diagram of the network architecture in which sFTP is used is given here.



Section 2: Business Value

Many organizations use sFTP servers to exchange files and other critical business documents with their trading partners. It is typically not used for low-latency transactional systems and is more apt for batch or bulk file transfers. Since these types of file transfers are ubiquitous in the

technical implementation of business systems, sFTP could have a place within any business system.

Section 3: Potential Business Data in Scope/Volume/Lifespan

sFTP can be used to transfer any type of data as long as it is in file format. Hence, there is essentially no limit to the value of the data which is transferred. The data itself will be largely dependent on the intended business use of the application leveraging sFTP.

Section 4: Use-Case Class

Data-In-Transit Protection – (files)

Section 5a: Technical Considerations

The following are considerations for PKI with regard to implementing quantum-safe technology:

1) **File Size:** sFTP can be used to transfer files of arbitrary size. The only limit could very well be the technical limit of the underlying hardware and software using sFTP.
2) **Throughput:** sFTP is not typically used for low-latency transactional applications, so real-time throughput is NOT typically a consideration. However, some business applications depend on sFTP to transmit large amounts of data within a restricted time window. Throughput becomes a consideration in this sense.
3) **Credential Management:** The underlying protocol enabling sFTP authentication (usually through SSH) requires credentials such as private keys to be properly and securely stored on the endpoints facilitating the sFTP connection.
4) **Support of Underlying Technology:** The endpoints facilitating the sFTP connection need to have the proper capabilities (e.g. OS, network connections, cryptographic software) to implement the sFTP connection.

Section 5b: Threat Considerations

sFTP servers have become a primary target for hackers, putting sFTP servers at risk of a costly data breach. (https://www.goanywhere.com/blog/2018/01/23/10-essential-tips-for-securing-ftp-and-sftp-servers).

The exact threats to sFTP depend upon the security environment in which it is used. For example, sFTP connections which are external over a public network are inherently more vulnerable to attack than those that are internal to an organization. Additional controls such as logging and monitoring can affect the overall threat level.

As sFTP uses asymmetric cryptography for authentication and key agreement, there is a both an inherent quantum threat to compromise the connection as well as a “Harvest-and-Decrypt” risk for the business data that is being transmitted.

Section 6: Types of Cryptography

sFTP mainly uses both symmetric and symmetric cryptography for protection of the file data which is being transmitted.

The asymmetric cryptography is used by the underlying SSH protocol to establish authentication and key agreement between the two endpoints. The files are then protected with symmetric cryptography during transmission.

Section 7: Technical Components

The main technical components are:

1) The Endpoints: the two endpoints engaged in the active session and their underlying technology.
2) The Network: the network over which the transmission occurs.

Note that the network may have several hops in between the connection endpoints. However, for the purpose of this use case, they are transparent passthroughs and so need not be given consideration.

The endpoints must:

3) Have the requisite capabilities to support the sFTP software including the requisite cryptographic functions.
4) Have the requisite capabilities to support the underlying authentication software (SSH).
5) Have the ability to store or otherwise send or receive the files being transmitted.
6) Have the ability to store and manage the credentials of the underlying authentication software (e.g. private keys).
7) Have access to the appropriate network over which the communication is to occur.

The network must be able to support the authentication protocol as well as the transfer of files.

Section 8: Crypto Locations

The sFTP protocol will either leverage its own cryptography as part of its own software when it was installed or will leverage the underlying cryptographic libraries of the machine on which it is used.

Any change to the cryptography used in the sFTP protocol amounts to a change in the cryptographic code in one of these locations. It is important to note that any such changes have some additional considerations:

1) The location of any cryptographic keys should be taken into consideration.

2) The surrounding protocols must be ensured to be compatible with any change in buffer size, throughput or protocol steps.

Please note that some sFTP implementations may either be bundled together with SSH or be modularly separated. In these situations, the cryptography and cryptographic locations of the two protocols may need to considered in tandem instead of separately. When considering changes to the cryptography of an implementation, whether or not the sFTP and SSH implementations are bound together or not should be taken into consideration.

Many popular sFTP products operate similarly in terms of cryptographic locations. The cryptographic code is embedded within the source code and binaries of the product. The private keys or certificates are typically stored locally and exist in .pem or .ppk files.

Section 9: Dependencies

The dependent use cases for sFTP are:

- Data Storage
- PKI/CA – (if certificates were used to establish authenticity of public keys)

Additionally, one would normally consider SSH as a dependent use case, but we have bound it together with sFTP for the purpose of this use case.

Section 10: Ability to Support Algorithms Simultaneously

By its nature, an sFTP endpoint would establish an individual sFTP connection with any number of other endpoints. Each connection would use fixed, established cryptographic algorithms for the lifespan of that connection. However, the connections between different endpoints would be theoretically independent of each other. Hence, any sFTP endpoint could theoretically implement different cryptographic algorithms for different connections. Thus, any migration to new algorithms can be done connection by connection when the other endpoint is ready.

The ability to support different algorithms simultaneously, therefore, depends on whether the particular sFTP product has been programmed to support this functionality. It would be beneficial to encourage sFTP providers to enable this functionality.

ANNEX G: MATRIX OF CRYPTOGRAPHY USE CASES

Annexes D, E and F of this document contain detailed multi-page write-ups of three different organizational uses of cryptography:

- Using Kerberos for Authentication
- Public Key Infrastructure & Certificate Authorities
- Secure File Transfer Protocols

This Annex is meant to give a more streamlined, referential view of the different factors which are of relevance to quantum-safe migration for the use cases listed in Annex B. Its main purpose is to display relevant use case considerations in a single pane of glass in order to serve as a starting point for planning a migration and/or transition to Quantum-Safe Cryptography (QSC). It has the potential to be used as both a point of reference for individual organizations in understanding the factors and potential consequences in planning a quantum migration as well as a basis for future workgroup activities in expanding guidance for the industry-at-large.

All of the information about the above use-cases can be summarized in a tabular format, in three of the twenty-nine rows of information within the matrix shown here (viz., rows 11, 7 and 8).

The other rows in this matrix are intended to provide planners with a starting point to discover how and where cryptography is used in the other use cases listed in Annex B.

A full-sized copy of this matrix may be obtained by emailing the [CFDIR Secretariat].

		Characteristics		Crypto Service					Data & Risk				Tech Details								Upgrade Considerations			
Use Case Family	Use Case/Protocol	Use Case Class	Industry Usage	Conf	Auth	Integrity	Dependencies	Downstream	Data	Harvest & Decrypt Risk	Classical Threats	Quantum Threats	Tech Consideration	Entities	Tech Components	Real-Time?	Algorithm Negotiation?	Persistent?	Typical OSI Level	Centralized or Decentralized	Changes to Standard Required for Hybrid or PQC?	Difficulty in Discovering Implementations?	Support for Simultaneous Algorithms	Upgrade Considerations
	CA/PKI	Authentication	Enables "Trust" in TLS and several application layer OSI protocols	Y	Y	N	None	Many	Limited	No	CA compromise Unathorized issuance of certificates Registration authority compromise	CA private key compromise RA private key compromise CRL compromise	Certificate size Signing performance Verification performance	Offline Root CA Online Issuing CA(s) Registration Authorities Inspection CAs Special Purpose CAs End Entities	Offline Root Environment Issuing Server and HSM CRL and OCSP capabilities RA devices CSR generation capabilities	No	No	Yes	TCP layer (level 4) and up	Centralized	Depends	Yes	Embedded in certificates	This is often a dependency for other use cases, so this would be consideration to upgrade first. It is unclear whether or not X.509 certificates are already capable of supporting PQC or hybrid. It would be imperative to push new CA chains to devices early in the process.
TLS	TLS/mTLS	Data Transfer	Enabling secure connection at OSI Application Layer	Y	Y	Y	CA/PKI	Some dependent protocols	Unlimited	Yes	MITM TLS Attacks (e.g. Heartbleed, POODLE, downgrade attacks)	Certificate private key compromise Passive session snooping	Certificate verification peformance Key agreement performance Cipher suite latency	Client Server CRL	Client or server machine Cryptographic libraries CRL capabilities	Yes	Yes	No	TCP layer (level 4) and up	Decentralized	No	No	Each entity can upgrade independently	While TLS is a standard protocol, it is so pervasive that there are many different implementations of it in many different real-world scenarios. Upgrade considerations may differ widely depending upon the implementation.
TLS	FTPS	Data Transfer	Enabling secure object/file transfer over TLS connection.	Y	Y	Y	CA/PKI	None	Unlimited	Yes	MITM TLS Attacks (e.g. Heartbleed, POODLE, downgrade attacks)	Certificate private key compromise Passive session snooping	Certificate verification peformance Key agreement performance	Client Server CRL	Client or server machine Cryptographic libraries CRL capabilities	No	Yes	No	TCP layer (level 4) and up	Decentralized	No	No	Each entity can upgrade independently	Much of the same considerations as with TLS. However, while connection may be real-time, file transfer has more of a batch feel to it, so latency is not as big of an issue
TLS	LDAPS	Client Entity Authentication	Enabling LDAP based client authentication over TLS connection	Y	Y	Y	CA/PKI	None	Limited	No	MITM Non-browser-based TLS attacks LDAP injection Log4j	Credential compromise Malicious access to resources	Certificate verification performance Key agreement performance	LDAP server LDAP client Resource	LDAP Client or Server machine LDAP Database	Yes	Yes	No	TCP layer (level 4) and up	Centralized Authority/Decentralized Use	No	No	Strategy between entities must be determined	LDAPS naturally uses the same TLS protocol as a standard TLS implementation. It will have much of the same considerations.
TLS	EAP-TLS	Wireless Authentication	Enable registration to a wireless access point	N	Y	Y	CA/PKI	None	Unlimited	Yes	MITM Non-browser-based TLS attacks	Credential compromise Session compromise Passive exposure of transmission	Certificate verificatio performance Key agreement performance	Client Server	Client or server machine	Yes	Yes	No	TCP layer (level 4) and up	Centralized	No	No	Each entity can upgrade independently	EAP-TLS naturally uses the same TLS protocol as a standard TLS implementation. It will have much of the same considerations.
TLS	DOT/DOH	Data Transfer	DNS over HTTPS (DOH) & DNS over TLS (DOT) provides confidentiality & integrity of DNS data and securing the communication channel.	Y	Y	Y	CA/PKI	None	Limited	No	DNS Spoofing and Integrity	Passive exposure of addressing information	Certificate verification peformance Key agreement performance Cipher suite latency	Client Server CRL	Client or server machine Cryptographic libraries CRL capabilities	Yes	Yes	No	TCP layer (level 4) and up	Decentralized	No	No	Each entity can upgrade independently	DOH/DOT naturally uses the same TLS protocol as a standard TLS implementation. Other than a few cursory checks to make sure nothing is missed, this will largely be taken care of naturally through the TLS upgrade.
SSH	SSH (proper)	Client Entity Authentication	Enabling Secure network channel for operating remote network services/resources	Y	Y	Y	CA/PKI	SFTP is a dependent protocol	Limited	No	Client Credential Theft and Spoofing Server (Root) Account Compromise Privilege Escalation	Client Credential Exposure	Key agreement performance Thumbprint storage	Client Server	Client or server machine Cryptographic libraries Thumbprint storage location	Yes	Yes	Yes	App layer (level 5)	Decentralized	Yes	Yes	Client upgrade depends upon server	Other than ensuring the server is upgraded before clients, the main consideration is maintaining persistent connection privileges with existing accepted clients for the upgrade.
SSH	SFTP	Object/File Data Transfer	Enabling secure object/file transfer over SSH connection.	Y	Y	Y	SSH	None	Unlimited	Yes	MITM Standard TLS attacks	Passive Exposure of Content	Key agreement performance Cipher suite latency	Client Server	Client or server machine Cryptographic libraries	No	Yes	Yes	App layer (level 5)	Decentralized	No	No	Client upgrade dependes upon server	It inherits the same considerations as SSH. This typically a batch process but may deal with high volumes of data, so overall latency may be a consideration.
	SAML	Authentication & Authorization	OpenID Connect and SAML are federated authentication industry standards. OAuth 2.0 provides authorization	N	Y	N	CA/PKI	None	Limited	No	Client Credential Theft/Impersonation/Privilege Escalation DoS Golden SAML Token Theft and Replay Attacks	Passive Golden SAML Attack	Signature verification peformance Key agreement performance Cipher suite latency	User Identity Provider Resource	User, IdP, or resource machines Cryptographic libraries	Yes	No	No	App layer (level 5)	Centralized Authority/Decentralized Use	Yes	No	Strategy between entities must be determined	The order of an upgrade is a key consideration. Upgrading the IdP could potentially lead to outages as resources and/or users may not be compatible with an upgraded IdP. It m ay be best to upgrade resources first, but it may not be possible to find and upgrade all resources in time.
	OAUTH / OpenID Connect	Authentication & Authorization	OAuth 2.0 provides authorization framework to protected resource	N	Y	N	CA/PKI	None	Limited	No	Client Credential Theft/Spoofing Privilage Esclation	IdP Credential Compromise	Signature verification peformance Key agreement performance Cipher suite latency	Client Authorization Server Resource Owner Resource Server	Entity machines Access tokens	Yes	No	No	App layer (level 5)	Centralized Authority/Decentralized Use	Unknown	No	Strategy between entities must be determined	Similar to the SAML situation, the order of an upgrade is a key consideration. Upgrading authorization servers first could lead to outages if resources are not compatible.
	Kerberos	Authentication & Authorization	Kerberos is Authentication device protocol over non-secure network	N	Y	N	None	None	Limited	No	Ticket spoofing (e.g. pass-the-hash) Compromise of ticket-granting service or authentication service	Compromise of User Authentication Extension Compromise of Session Key Agreement Extension	DNS TXT Records Integrity Certificate verification perfomance Key agreement performance	Client Service Provider Client Authenticator Ticket Granting Service (TGS)	Entity machines Key Distribution Centre (KDC) Tickets	Yes	No	No	App layer (level 5)	Centralized Authority/Decentralized Use	Yes?	No	Strategy between entities must be determined	Asymmetric cryptography is not used in the base Kerberos implementation. It is used in extensions for User Authentication and Session Key Agreement. This should simplify upgrade as compared to signature-based token-granting protocols.
IPSec	IPSec - AH	Authentication	IP Layer Authentication and Integrity of Transmission	N	Y	Y	IPSec - SA	None	Limited	No	Replay Attacks Option-Insertion Attacks	Packet Integrity (if IKE is compromised) Sender Authentication (if IKE is compromised)	Latency	Sender Receiver	Entity machines Packets	Yes	No	No	IP Layer (level 3)	Decentralized	No	No	Each entity can upgrade independently	IPSec-AH is based on symmetric crypto. The upgrade consideration typically occur in the IPSec SA use case.
IPSec	IPSec - ESP	Data Transfer	IP Layer Tunneling Protocol	Y	Y	N	IPSec - SA	None	Unlimited	No	Packet Substitution	Passive Data Snooping (if IKE is compromised)	Latency	Sender Receiver	Entity machines Packets	Yes	No	No	IP Layer (level 3)	Decentralized	No	No	Each entity can upgrade independently	IPSec-ESP is based on symmetric crypto. The upgrade consideration typically occur in the IPSec SA use case.
IPSec	IPSec - Security Association (e.g. IKE, ISAKMP)	Authentication & Authorization	Key Exchange Protocol for IPSec	N	Y	Y	CA/PKI	IPSec - AH, ESP	Limited	Yes	MITM	Key Compromise to affect dependent protocols	Certificate verification performance Key agreement performance	Sender Receiver	Entity machines Cryptographic libraries	Yes	Yes	No	IP Layer (level 3)	Decentralized	Yes	No	Each entity can upgrade independently	Much like TLS, entities can upgrade independently. The exception may be for a closed, dependent system where entities must upgrade together in a coordinated fashion at the same time.
	MACsec	Data Transfer	Link layer transmission protection	Y	Y	Y	None	None	Unlimited	Yes	MITM	Passive data snooping	Bandwidth Latency	2 network devices	Entity machines	Yes	Yes	No	Link layer (level 2)	Decentralized	Yes	No	Each entity can upgrade independently	MACsec can use an asymmetric extension to perform key exchanges to enable encryption. It is decentralized and both parties can upgrade independently.
	DNSSEC	Authentication & Authorization	Authentication of DNS resolution	N	Y	Y	CA/PKI	None	Limited	Yes	Key compromise Invalid DNS entries and rerouting	Passive DNS system compromise	Certificate verification Data integrity	DNS zone DNS resolver Client	DNS recursion Zone of trust	Yes	No	No	App layer (level 5)	Centralized	No	No	Client upgrade depends upon server	DNSSEC tends to be very centrally managed by zones. The impetus would be for the DNS zone to upgrade but be backward compatible and then allow clients and resolvers to upgrade.
	S/MIME	Data Transfer	Preserves the confidentiality of e-mail transmissions	Y	Y	Y	CA/PKI	None	Unlimited	Yes	MITM Message spoofing	Passive data snooping Forged emails	Certificate verification Signature verification	Sender Receiver	CA CRL Sender and receiver machines	No	No	No	App layer (level 5)	Decentralized	No	No	Each entity can upgrade independently	Even though a CA is often used, S/MIME is decentralized point-ot-point between sender and recipient. Thus, each can upgrade independently to be quantum safe as long as their parameters on the next message are compatible.
	PGP	Data/File Transfer	File transmission protocol (and product)	Y	Y	Y	None	None	Unlimited	Yes	Device compromise Data Leakage	Passive exposure of transmitted data	Cerrtificate verification performance Signature verification performance	Sender Recipient PGP server	PGP program PGP key storage location	No	No	No	App layer (level 5)	Decentralized	No	No	Each entity can upgrade independently	PGP is a decentralized point-to-point file encryption program. Once an individual program is upgraded to quantum safety, it is just a matter of ensuring that both parties individually upgrade. If a PGP server is used to centralize communication, it should be upgraded first.
	DMARC DKIM SPF	Email Authentication & Authorization	Domain protection from unauthorized use or spoofing	N	Y	N	CA/PKI	Variable	Limited	No	Email spoofing Fraudulent senders Spam/malicious e-mails	Compromise of e-mail authentication	Certificate verification Signature verification	ISPs Individual organizations	ISP mail server Organization mail server	No	No	No	App layer (level 5)	Decentralized	No	No	Dependent on ISP	As this protocol is primarily meant for ISPs, it is likely they who would initiate the upgrade. They would likey upgrade their systems first and then require individual organizations to the same.
	WPA	Data Transfer	Registers endpoints onto a wireless network	Y	Y	Y	None	None	Unlimited	Yes	Unauthorized registration Session compromise	Passive exposure of transmitted data	Key agreement performance Bandwidth	Client Wireless network	Client device Wireless router or base station	Yes	Yes	Yes	App layer (level 5)	Decentralized	Yes	No	Each entity can upgrade independently	In most cases, the wireless access point will list out the protocol options from which the client selects, so upgrade can be done in any order.
	DECT	Data (Voice) Transfer	Special protocol to protect voice transfer	Y	Y	N	None	None	Unlimited	Yes	Key compromise Session snooping	None	Bandwidth Latency	DECT handset DECT base station	Subscription PIN code OTA channel	Yes	No	Yes	App layer (level 5)	Decentralized	Yes	Yes	Each entity can upgrade independently	To this point, DECT does not contain any asymmetric cryptography and so there is nothing to upgrade.
	Mobile NFC	Mobile Enclave Operations	Enables wireless transmission of limited data next to another device (such as a POS device)	Y	Y	Y	None	None	Limited	No	Channel extension attacks Protocol attacks	Enclave compromise	Latency Bandwidth	Mobile device (enclave) Base station Backend processor	Chip hardware Special-purpose hardware	Yes	No	No	App layer (level 5)	Centralized Provisioning/Decentralized Use	Yes	No	Centrally managed by mobile provider or industry organization	This is a specialized area involving either telcos or card organizations for example. Much of the cryptography is embedded into hardware and so it will be a large effort to upgrade. The upgrade will likely have at least some centralized oversight.
	Code Signing	Code Integrity	Signing code to preserve its integriy and/or verify the origin	N	Y	Y	CA/PKI	None	Limited (to code) / Unlimited (code itself)	No	Signing service compromise Forged signatures	Signing key compromise	Signature verification Signature size	Signing service Code repository Code verifier	Signature on code Verification subroutines	Sometimes	No	No	App layer (level 5)	Centralized Signing/Decentralized Use	No	No	Strategy between entities must be determined	While it would be theoretically be advisable to upgrade code verifiers first, different vendors may make this challenging. It may be acceptable to upgrade each independently with responsibility on the verifier to ensure their system can handle quantum-safe signatures.
	MFA	Authentication & Authorization	A secondary authentication technique and channel	N	Y	N	Depends upon implementation	Likely none	Limited	No	Secondary channel compromise	Secondary channel compromise	Transmission time due to timeouts	MFA authenticator MFA client	MFA channel MFA data element	Yes	No	No	App layer (level 5)	Depends	N/A	No	Each entity can upgrade independently	MFA can refer to different methods of authentication such as a one-time-password or calculated token value. They are implementation-dependent. They are typically short-lived and may not always involve asymmetric cryptography. Those that do can usually be upgraded independently.
	HSMs	Data & Key Storage and Processing	Protection of highly sensitive data such as keys and the accompanying processing	Y	N	Y	None	Local dependencies only	Limited	Yes	DoS due to tampering	Master key compromise Appliance compromise	Throughput Latency	HSM Injection and supporting hardware HSM clients	HSM master keys Injection mechanisms	Yes	No	Yes	App layer (level 5)	Decentralized	N/A	No	Each entity can upgrade independently	HSMs sometimes leverage a set of asymmetric keys as their master keys. It would be up to the HSM manufacturer to upgrade their master key structure to be quantum safe. HSM master key use is typically localized and so can be done independently, although it may require a key encryption migration.
	Cloud Native Encryption	Protection At Rest	Protection of data at rest in a cloud environment	Y	N	Y	None	None	Unlimited	Yes	Key compromise Data exfiltration Unauthorized access to cloud services	Passive exposure of content Cloud service compromise	Encryption performance I/O performance Data integrity	Cloud native data storage Cloud key management system Consuming applications CASB	SaaS, PaaS, IaaS components	Yes	No	Yes	App layer (level 5)	Decentralized	N/A	No	Each entity can upgrade independently	Upgrade is dependent on the cloud services provider. The CSP would be responsible for the upgrade, but cloud consumers would need to understand the effects on their applications. In particular, as a data migration would likely be required, its effects would need to be socialized with the consumer.
	Data-At-Rest Protection	Protection At Rest	Protection of data at the storage level	Y	N	Y	None	None	Unlimited	Yes	Key compromise Data exfiltration	Passive exposure of content	Encryption performance I/O performance Data integrity	Key storage Data storage Consuming applications	Depends upon instance	Yes	No	Yes	App layer (level 5)	Decentralized	N/A	No	Each entity can upgrade independently	Data-At-Rest is a wide and varied field which is not standardized. Upgrade would depend upon implementation, but it is typically a localized service which is transparent to other entities. One commonality would be the necessity of a data migration which is typically a long and arduous task.

Term used in the Matrix	Definition of the Term
Use Case Family	The overarching family in which the use case belongs (where applicable). For example, TLS (1-way) and mTLS are both TLS protocols while FTPS and LDAPS both leverage the TLS protocol as part of a more extensive protocol.
Use Case Protocol	The actual use case.
Use Case Class	The main purpose of the protocol.
Industry Usage	A short description of what the protocol is typically used for.
Conf (Confidentiality)	"Yes/No" if the protocol provides confidentiality for the data involved. This is highlighted if it is the main purpose of the protocol.
Auth (Authentication)	"Yes/No" if the protocol provides authentication for one of the participants. This is highlighted if it is the main purpose of the protocol.
Integrity	"Yes/No" if the protocol provides integrity for the data involved. This is highlighted if it is the main purpose of the protocol.
Dependencies	Other use cases which would be an upstream dependency for this protocol.
Downstream	Other use cases downstream which would leverage this protocol.
Data	Usually "Limited/Unlimited". Limited if the data involved in the protocol itself is constrained. For example, SSH is Limited as it contains only simple identification and authorization information. However, SFTP is Unlimited as any type of file containing any type of data can be transmitted.
Harvest & Decrypt Risk	"Yes/No" if there is a Harvest & Decrypt risk.
Classical Threats	Lists the well-known classical threats associated with this protocol.
Quantum Threats	Lists the new quantum threats associated with this protocol.
Tech Consideration	Lists the main tech considerations or constraints needed to be taken into account in implementing this protocol. Examples include high latency, low bandwidth, memory restrictions, etc.
Entities	Lists the entities that are typically involved in the protocol.
Tech Components	List the main technical components of the protocol.

Term used in the Matrix	Definition of the Term
Real-Time?	"Yes/No" depending on whether or not the protocol is used in real-time systems. For example, SAML is “Yes” as it is used for Single Sign On (SSO) which happens in real time. SFTP is N since it is often used for batch processes.
Algorithm Negotiation?	"Yes/No" depending on whether or not the cryptographic algorithms are negotiated during the protocol itself.
Persistent?	"Yes/No" depending on whether or not a previous connection or instance retains knowledge of the previous one or starts anew.
OSI Layer	The layer in the OSI computing model in which this protocol typically operates.
Centralized or Decentralized	The details of the amount of centralization of the protocol. For example, CA/PKI are usually centrally managed. TLS is decentralized as any two devices can independently form a TLS connection. SAML has centralized authority (identity provider), but its users and resource owners work decentralized.
Changes to Standard Required for Hybrid or PQC?	"Yes/No" depending on whether changes to the standard need to occur in order to enable hybrid or PQ algorithms. For example, TLS is “No” since its new cipher suites can be added without changing the basic protocol. SAML is “Yes” since it is not clear how SAML will treat hybrid or PQ algorithms, particularly when some users and relying parties are quantum-ready while others may not be.

ANNEX H: OVERVIEW OF HYBRID CRYPTOGRAPHY

This Annex contains a whitepaper on the topic of **hybrid cryptography** to introduce this emerging area of standards and technology development in the context of Post-Quantum Cryptography (PQC) considerations. This Annex was initially published in 2022 and then updated to reflect new developments and discussions (e.g., in standards development organizations) as of June 2024.

Background / Overview

As the world prepares for the upcoming quantum era, work is underway globally to prepare for its potential impact on cryptography. The advent of powerful quantum computers able to run known quantum algorithms will threaten the cryptography in use today.

This preparation work is underway among international, regional and national standards bodies, as well as the global information and communications technology (ICT) industry and community. For example, the Post-Quantum Cryptography Standardization project by the U.S. National Institute of Standards and Technology (NIST) will select and standardize post-quantum cryptography (PQC) algorithms. Not only is there a need to standardize and implement these PQC algorithms, but also to provide guidance for the transition from the current cryptographic paradigms for the current ICT protocols, tools, and processes, to a future PQC paradigm for ICT protocols, tools and processes.

Two topics related to the upcoming transition to a PQC future are **cryptographic agility** and **hybrid cryptography**. These topics are receiving attention from stakeholders including academia, standards bodies, the ICT supply chain providing cryptographic products, services, and solutions, and enterprises and governments.

While at a high level the term '**hybrid cryptography'** has been used globally, there is not yet a consensus on the best-detailed approaches related to **hybrid cryptography**.

Alternative terminology is sometimes used, such as **dual signatures**, **composite cryptography** and **multiple encryption**. The term **hybrid cryptography** might not be ideal, but so far there is not yet a consensus on a better alternative.

This objective of this paper is to provide an overview of **hybrid cryptography** to increase the reader's understanding of this complex topic. This understanding will be essential to inform and facilitate appropriate decision-making during the upcoming transition to a quantum era.

It is anticipated that updated versions of the guidance outlined in this Annex will be released in the future.

What is Hybrid Cryptography?

Hybrid cryptography, in the context of this whitepaper, is defined as the usage of a post-quantum cryptographic system combined with another public-key cryptographic system (whether post-quantum or traditional) that contributes to the same cryptographic objective. The cryptographic objectives that rely upon public-key cryptography most commonly involve the use of digital signatures or key-establishment methods.

The goal of hybrid cryptography is for the cryptographic objective to achieve the security of the strongest of all cryptographic methods used in the combination. This goal may be achieved over time depending on how hybrid is employed. For example, a hybrid digital signature might enable backwards compatibility for verifiers that do not yet support PQC, but the ultimate goal will be that all verifiers will validate the stronger PQC signature at the end of the migration. Strictly speaking, during the migration, legacy verifiers may or may not support hybrid cryptography produced by the signer during the migration, but the system does.

In the context of this whitepaper, the following are **not** considered hybrid cryptography:

- In key establishment, obtaining key contributions out-of-band, such as previously established keys, passwords, or keys from quantum key distribution devices.
- Using different public-key cryptosystems at different network protocol layers (such as the lowest physical layer and the highest application layer)
- In public-key encryption, using public-key cryptography to establish a secret key, and using symmetric cryptography to encrypt the message with the secret key. This is occasionally called “hybrid public-key encryption”, as in RFC 9180.

Why is Hybrid Cryptography important to understand?

Some threat actors may already be storing encrypted information that they have intercepted and copied, with a view to decrypting it in the future using quantum computers. Any information that needs to be protected for a long time (e.g., corporate trade secrets, classified government documents, personal health information) may already be at risk if traditional cryptography, such as ECC and RSA, is used to safeguard that information today. Both ECC and RSA are known to be at risk from quantum computer attacks. Organizations should therefore transition to using post-quantum cryptography (PQC) to protect their information. However, the transition itself has its own costs and risks to consider.

Relevant considerations include:

- Migration:

  A total transition to using PQC may take **several years or even decades**. Business requirements need to be maintained throughout the duration of this transitional state.

- Resiliency:

  Post-quantum cryptography systems are relatively new. PQC uses mature designs and has been intensively evaluated over the past five years, but it has still not been subjected to as many years of cryptanalysis as the current public-key cryptography (ECC and RSA). So, there remains a risk that a particular PQC system—or even cryptographic family of PQC systems—could be broken by some unforeseen cryptanalytic attack. However, the risk to systems that do not transition to PQC is generally considered to be greater.

Hybrid cryptography has been proposed to address both considerations.

Advantages of hybrid cryptography may include:

- Facilitating migration:
  - Testing post-quantum cryptography in real world settings before the quantum threat materializes, and before we rely entirely on post-quantum cryptography.
  - Continuing to comply with existing cryptography requirements or certifications, while also defending against quantum attacks.
  - Providing backwards compatibility with legacy applications, in the context of digital signature cryptography.
- Improving resiliency:
  - Reducing the cryptographic risk of an unknown classical or quantum attack on a single cryptographic system (or family of cryptographic systems).
  - Support defence-in-depth by providing redundant cryptographic systems.
- Compatibility:
  - Allowing parties with differing policies on required cryptography to comply with both policies by applying both required kinds of cryptography.

Applicability of Hybrid Cryptography in cryptographic systems

The quantum threat to cryptographic systems predominantly targets public-key cryptography in its two most common use cases: digital signatures and key establishment. It is in these use cases that new post-quantum cryptography is being proposed and where system owners may wish to use hybrid cryptography.

Hybrid key establishment combines keys from two or more different key-establishment methods in such a way that a weakness in any individual method will not be sufficient to expose the resulting shared key. Typically, we would measure the security of the hybrid key establishment to be at least that of the strongest key-establishment method used in the combination. In particular, combining a traditional key-establishment method (e.g. Elliptic Curve Diffie Hellman (ECDH) or RSA key transport) with a post-quantum method would result in hybrid key establishment that maintains its security against the quantum threat only if the PQC method remains strong. Therefore, resiliency use cases may require hybrid to combine multiple PQC methods to ensure security against the quantum threat.

A hybrid digital signature combines two or more digital-signature methods in such a way that validation requires verification of some or all of the signatures, based on policy. If the verifier's policy requires all the included signatures to pass verification, the resulting security of the hybrid digital signature would be considered to be equal to the strongest signature. In the case where a policy requires only a subset to be verified, the policy could be specific to which signature(s) must be verified or only specify the size of the subset to be verified. The verifier’s policy might be configurable or imposed by the signer. Hybrid digital signatures that combine a traditional digital signature (e.g., Elliptic Curve DSA or RSA) and a PQC signature with a policy that one signature must be valid may allow for backwards compatibility to assist in system migrations. In such a use case, the policy must be configurable and should specify which signature must be valid in order to achieve the migration end-state where the post-quantum algorithms must be valid.

The security of hybrid digital signatures must be carefully assessed based upon the verifier policy and the strength of the underlying signatures. For example, if a policy allows any signature and does not specify which signatures must be valid, the security of that hybrid digital signature would be considered to be equal to the weakest of the signature methods; therefore, if a traditional digital signature is included, the hybrid cryptography would not be secure against the quantum threat under that policy. It is important for system administrators to understand the policy applied by the hybrid cryptography in use.

Implementation

Hybrid is a **very complex** topic, from cryptanalysis and implementation perspectives. Thus, **additional time and effort** will be required during some phases, such as risk analysis, migration and testing, so this should be factored into the overall plans and strategy for quantum readiness.

General considerations:

- Avoid in-house development; strongly prefer a standardized method when that becomes available.

- Prefer a solution that allows for cryptographic agility. **Cryptographic agility** describes a system, architecture or state where cryptography is **planned**, **built** and **operated** to ensure that replacing an algorithm does not significantly change the functioning of the application, protocol or system. The goal is to minimize the impact of changing cryptographic functions in terms of cost, time, resources, and information security risk. Cryptographic agility can assist in the transition to using hybrid cryptography, or from hybrid cryptography if a different end state is desired. Information on how an organization can employ cryptographic agility is available from the Canadian Centre for Cyber Security in ITSAP.40.018.

If the motivation to use hybrid is to improve resiliency by reducing cryptographic risk, then one should choose the component methods in the hybrid solution to satisfy cryptographic diversity. **Cryptographic diversity** is the availability of cryptographic methods from different families which are unlikely to be vulnerable to the same cryptanalytic attack. Hybrid cryptography employing cryptographic diversity will mitigate a broader cryptographic risk. Cryptographic diversity can also be of benefit to cryptographic agility, allowing a vulnerable method to be replaced with a different cryptographic family in a timely manner. With a plan to standardize a PQC portfolio that has cryptographic diversity, NIST has made initial PQC algorithm selections and is expected to select an alternate PQC key establishment algorithm from Round 4 candidates of the PQC Standardization Process,[17] as well as considering additional PQC digital signature candidates submitted under a separate call.[18]

It is important to consider the availability of a proposed hybrid solution, whether and/or how a third-party vendor provides the solution, and whether the solution has intellectual property restrictions.

---

[17] PQC Standardization Process: Announcing Four Candidates to be Standardized, Plus Fourth Round Candidates, July 5, 2022

[18] NIST Announces Additional Digital Signature Candidates for the PQC Standardization Process, July 17, 2023

It is also important to assess the suitability of a hybrid solution for the desired use case. In a migration use case, the hybrid solution will combine a traditional method with a post-quantum method. In a resiliency use case, the hybrid solution should combine more than one post-quantum method. Parameters to consider include processing time, memory requirements, bandwidth requirements, certification (regulations, standards), backwards compatibility, forwards compatibility, upgrade complexity, configuration complexity, management / operations. Specific protocols may require the use of hybrid cryptography, since PQC is often not a drop-in replacement for traditional cryptographic methods.

Resources, next steps, and references

Organizations requiring assistance are encouraged to contact the Canadian Centre for Cyber Security (contact@cyber.gc.ca or 1-833-CYBER-88) or the CFDIR Secretariat (cfdiroffice-bureaudufcrin@ised-isde.gc.ca).

Provided below is a list of informative references that offer more information on hybrid cryptography. However, be aware that hybrid is currently a fluid topic, and these documents may not reflect the final approach standards development organizations may take. The CFDIR Quantum-Readiness Working Group will continue to update this hybrid guidance paper and its other Best-Practices and Guidelines documents during the quantum-safe transition.

Products from Government Agencies and Standards Development Organizations:

- **Cloud Security Alliance**, “Mitigating the Quantum Threat with Hybrid Cryptography”, https://cloudsecurityalliance.org/artifacts/mitigating-the-quantum-threat-with-hybrid-cryptography/, 2019-06-17.
- **ENISA**, "Post-Quantum Cryptography: Current state and quantum mitigation", https://www.enisa.europa.eu/publications/post-quantum-cryptography-current-state-and-quantum-mitigation, May 2021.
- **ENISA**, “Post-Quantum Cryptography – Integration study”, https://www.enisa.europa.eu/publications/post-quantum-cryptography-integration-study, 2022-10-18.
- **ETSI** TS 103 744, "Quantum-safe Hybrid Key Exchanges", https://portal.etsi.org/webapp/WorkProgram/Report_WorkItem.asp?wki_id=56901, 2020-12-23.
- **IETF** RFC 9370, “Multiple Key Exchanges in the Internet Key Exchange Protocol Version 2 (IKEv2)”, https://www.rfc-editor.org/rfc/rfc9370.txt, May 2023.

- **ITU-T** X.509, "Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks", https://www.itu.int/rec/T-REC-X.509-201910-I/en, 2019-10-14.
- **NIST,** "Post-Quantum Cryptography FAQs", https://csrc.nist.gov/Projects/post-quantum-cryptography/faqs#xisl, 2020-01-28.

Draft Work from Standards Development Organizations:

- Mike Ounsworth, John Gray, Massimiliano Pala, Jan Klaußner, Scott Fluhrer, "Composite ML-KEM for Use in the Internet X.509 Public Key Infrastructure and CMS", draft-ietf-lamps-pq-composite-kem-03 <https://datatracker.ietf.org/doc/draft-ietf-lamps-pq-composite-kem/03/>, 2024-03-02.
- Mike Ounsworth, John Gray, Massimiliano Pala, Jan Klaußner, Scott Fluhrer, "Composite ML-DSA for use in Internet PKI", draft-ietf-lamps-pq-composite-sigs-01 <https://datatracker.ietf.org/doc/draft-ietf-lamps-pq-composite-sigs/01/>, 2024-06-06.
- Douglas Stebila, Scott Fluhrer, and Shay Gueron, "Hybrid key exchange in TLS 1.3", draft-ietf-tls-hybrid-design-10 <https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design/10/>, 2024-04-05.
- Alison Becker, Rebecca Guthrie and Michael J. Jenkins, "Related Certificates for Use in Multiple Authentications within a Protocol", draft-ietf-lamps-cert-binding-for-multi-auth-05 <https://datatracker.ietf.org/doc/draft-ietf-lamps-cert-binding-for-multi-auth/05/>, 2024-05-30.
- Stavros Kousidis, Johannes Roth, Falko Strenzke, Aron Wussler, "Post-Quantum Cryptography in OpenPGP", draft-ietf-openpgp-pqc-04 <https://datatracker.ietf.org/doc/draft-ietf-openpgp-pqc/04/>, 2024-07-08.
- Florence Driscoll, Michael Parsons, "Terminology for Post-Quantum Traditional Hybrid Schemes", draft-ietf-pquip-pqt-hybrid-terminology-03 <https://datatracker.ietf.org/doc/draft-ietf-pquip-pqt-hybrid-terminology/03/ >, 2024-05-09.
- Aritra Banerjee, Tirumaleswar Reddy, Dimitrios Schoinianakis, Tim Hollebeek, "Post-Quantum Cryptography for Engineers", draft-ietf-pquip-pqc-engineers-04 <https://datatracker.ietf.org/doc/draft-ietf-pquip-pqc-engineers/04/>, 2024-05-21.
- Nina Bindel, Britta Hale, Deirdre Connolly, Florence Driscoll, "Hybrid signature spectrums", draft-ietf-pquip-hybrid-signature-spectrums-00

<https://datatracker.ietf.org/doc/draft-ietf-pquip-hybrid-signature-spectrums/00/>, 2024-05-24.

Industry Specifications:

- Ehren Kret, Rolfe Schmidt, “The PQXDH Key Agreement Protocol”, Revision 3 <https://signal.org/docs/specifications/pqxdh/>, 2024-01-23.
- Apple Security Engineering and Architecture (SEAR), “iMessage with PQ3: The new state of the art in quantum-secure messaging at scale”, <https://security.apple.com/blog/imessage-pq3/>, 2024-02-21.

Academic Papers:

- Johannes Müller, Jan Oupický, “Post-quantum XML and SAML Single Sign-On”, Cryptology ePrint Archive, Report 2024/828 <https://ia.cr/2024/828>, 2024-05-27.
- Manuel Barbosa, Deirdre Connolly, João Diogo Duarte, Aaron Kaiser, Peter Schwabe, Karoline Varner, Bas Westerbaan, "X-Wing: The Hybrid KEM You’ve Been Looking For”, Cryptology ePrint Archive, Report 2024/039 <https://ia.cr/2024/039> 2024-04-15.
- Adam Petcher, Matthew Campagna, “Security of Hybrid Key Establishment using Concatenation”, Cryptology ePrint Archive, Report 2023/972 <https://ia.cr/2023/972>, 2023-06-21.
- Alexandre Augusto Giron, “Migrating Applications to Post-Quantum Cryptography: Beyond Algorithm Replacement”, Cryptology ePrint Archive, Report 2023/709 <https://ia.cr/2023/709>, 2023-05-17.
- Nina Bindel, Britta Hale, “A Note on Hybrid Signature Schemes”, Cryptology ePrint Archive, Report 2023/423 <https://ia.cr/2023/423>, 2023-03-24
- Alexandre Augusto Giron, João Pedro Adami do Nascimento, Ricardo Custódio, and Lucas Pandolfo Perin, "Post-Quantum Hybrid KEMTLS Performance in Simulated and Real Network Environments", Cryptology ePrint Archive, Report 2022/1639 <https://ia.cr/2022/1639>, 2022-11-25.
- Mila Anastasova, Panos Kampanakis and Jake Massimo, "PQ-HPKE: Post-Quantum Hybrid Public Key Encryption", Cryptology ePrint Archive, Report 2022/414 <https://ia.cr/2022/414>, 2022-11-05.

- Jiewen Yao, Krystian Matusiewicz, and Vincent Zimmer, "Post Quantum Design in SPDM for Device Authentication and Key Establishment", Cryptology ePrint Archive, Report 2022/1049 <https://ia.cr/2022/1049>, 2022 10 04.
- Diana Ghinea, Fabian Kaczmarczyck, Jennifer Pullman, Julien Cretin, Stefan Kölbl, Rafael Misoczki, Jean-Michel Picod, Luca Invernizzi and Elie Bursztein, "Hybrid Post-Quantum Signatures in Hardware Security Keys", Cryptology ePrint Archive, Report 2022/1225 <https://ia.cr/2022/1225>, 2022 09 15.
- Sara Stadler, Vitor Sakaguti, Harjot Kaur and Anna Lena Fehlhaber, "Hybrid Signal protocol for post-quantum email encryption", Cryptology ePrint Archive: Report 2021/875 <https://ia.cr/2021/875>, 2021-06-24.
- Reza Azarderakhsh, Rami El Khatib, Brian Koziel and Brandon Langenberg, "Hardware Deployment of Hybrid PQC", Cryptology ePrint Archive: Report 2021/541 <https://ia.cr/2021/541>, 2021-05-06.
- Matthew Campagna and Adam Petcher, "Security of Hybrid Key Encapsulation", Cryptology ePrint Archive: Report 2020/1364 <https://ia.cr/2020/1364>, 2021-01-14.
- Jia Xu, Yiwen Gao and Hoonwei Lim, "Practical Quantum-Safe Stateful Hybrid Key Exchange Protocol", Cryptology ePrint Archive: Report 2020/763 <https://ia.cr/2020/763>, 2020-06-21.
- Nina Bindel, Jacqueline Brendel, Marc Fischlin, Brian Goncalves and Douglas Stebila, "Hybrid Key Encapsulation Mechanisms and Authenticated Key Exchange", Cryptology ePrint Archive: Report 2018/903 <https://ia.cr/2018/903>, 2019-10-21.
- Panos Kampanakis, Peter Panburana, Ellie Daw and Daniel Van Geest, "The Viability of Post-quantum X.509 Certificates", Cryptology ePrint Archive: Report 2018/063 <https://ia.cr/2018/063>, 2018-01-27.
- Jacqueline Brendel, Marc Fischlin and Felix Günther, "Breakdown Resilience of Key Exchange Protocols: NewHope, TLS 1.3, and Hybrids", Cryptology ePrint Archive: Report 2017/1252 <https://ia.cr/2017/1252>, 2019-09-16.

ANNEX I: CRYPTOGRAPHIC-AGILITY EXERCISE NOTES

I.1 Introduction and Exercise Description

This Annex contains a detailed example of a systematic approach to think about "how and where to start planning" to migrate quantum-vulnerable cryptography, which may be currently used in an information technology system, to make use of standardized quantum-safe cryptography in the future. There is general consensus in the industry that making use of "cryptographic agility" may facilitate such a migration. This being said, there are many different perspectives on the precise meaning of crypto-agility, and a lack of clarity with respect to what crypto-agility means in practice for a system owner.

The approach documented in this Annex was developed during the course of fifteen meetings and greenlighting sessions by members of the CFDIR Quantum-Readiness Working Group (QRWG), spanning six months of elapsed time in 2023. The inputs and perspectives of security and cryptographic experts from sixteen different public and private sector organizations are reflected in this work.

I.1.1 Purpose

The purpose of the exercise described in this Annex is to provide an example of working through a cryptographic migration on a conceptual Information Technology (IT) system to identify and articulate the practical considerations for different use cases which the migration must consider. The belief is that thinking through a migration using a systematic approach will enable finding more of these considerations than if we were to attempt to list them.

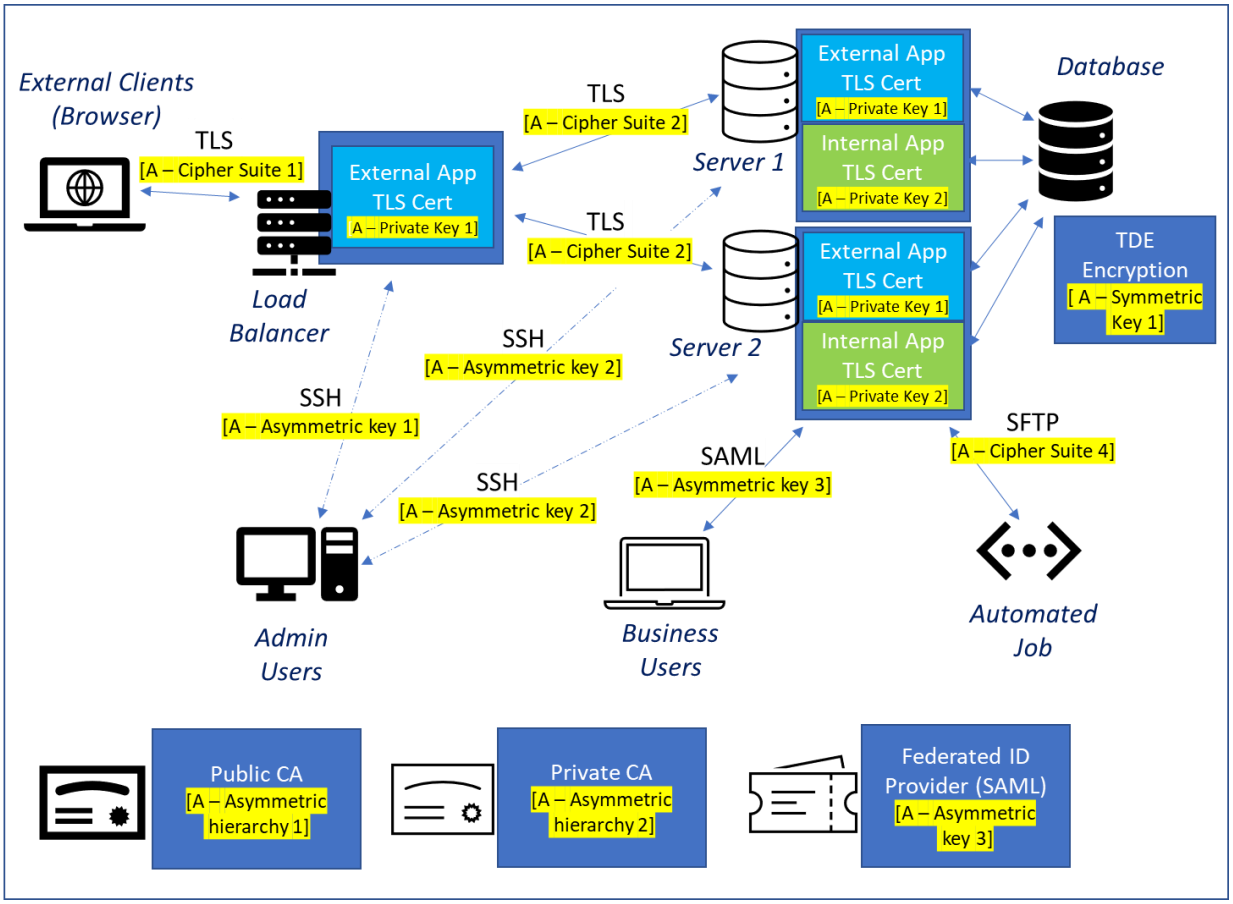
The general method is to start with the scenario:

> Your CEO comes to you and says: *"I want you to make our system quantum safe*."
>
> **What do you do?**

For this exercise, we diagramed an arbitrary IT system that any organization may be using today to interact electronically with its customers, suppliers, and/or other parties (internal and external to the organization). This system is illustrated in Figure I-1 on the next page.

We then worked through all of the considerations we could think of which could arise in the course of migrating the cryptography needed to protect the confidentiality and integrity of data handled by each component in the system. We also considered the cryptography necessary to authenticate users, a major use of cryptography that spans many of the use cases examined. While the quantum-safe angle was the specific focus of this exercise, the same process should be

applicable to any cryptographic algorithm migration. These considerations will be key in determining what would need to be done to add cryptographic agility to an IT system.



Notes:

1. The cryptography locations are highlighted in yellow. The goal is to change only these.
2. This is the "before" picture. An equivalent "after" picture is needed.
3. Public Certificate Authority (CA), Private CA, and Federated Identity (ID) Provider are enterprise services used by other systems.
4. The external application uses the public CA and the internal app uses the private CA.
5. The Federated ID Provider provides access for the business users to the internal app.
6. Administrative users can SSH into any box or 'appliance'.

Figure I-1. Arbitrary IT system, not currently "Crypto-agile", to be migrated to become "Quantum-Safe".

For the purpose of this exercise, we defined **crypto-agility** to be **the ability to achieve the desired cryptographic end state** (e.g., quantum-safety) **by changing ONLY the cryptographic algorithms used in a system**. With reference to the system illustrated in Figure I-1, pieces highlighted in yellow indicated a starting point for discussing elements that need to be migrated.

There are other components of the system that were identified through our systematic discussions that will need to be migrated as well. The details of the migration considerations for each identified component are captured in the text that follows.

A major goal of crypto agility is enable to quick reactions to resist new cryptographic attacks, ideally through system configuration updates, as outlined in the following references :

- ***Guidance on becoming cryptographically agile - ITSAP.40.018***, Canadian Centre for Cybersecurity, https://www.cyber.gc.ca/en/guidance/guidance-becoming-cryptographically-agile-itsap40018, May 2022
- ***Cryptographic agility***, Wikipedia contributors, https://en.wikipedia.org/w/index.php?title=Cryptographic_agility&oldid=1226050287, (last visited May 29, 2024)
- ***Cryptographic Agility Infographic****, U.S. Department of Homeland Security, https://www.dhs.gov/publication/cryptographic-agility-infographic,* May 12, 2022

We note there are other options that could contribute to achieving quantum-safety, such as:

- Changing the architecture of a system (i.e., ***architectural agility***);
- Changing the data flows of a system (i.e., ***data agility***);
- Changing the technology within a system (i.e., ***technological agility***);
- Changing the process involved (i.e., ***process agility***);
- Changing the business requirements involved (i.e., ***business agility***).

Although these types of agility are all worthy of their own studies, this Annex concentrates on the aspects of crypto agility as defined at the top of this page.

I.1.2 Structure of this Annex

The next section of this Annex describes thirteen different use cases. Each use case is explored in depth and based on the example system diagrammed in Figure I-1. The use cases were discussed among industry, academic, and governmental experts. Please note these use cases are by no means an exhaustive list. It is envisioned that new use cases may explored and added to future revisions of this Annex.

The notes for each of the use cases described in Section I.2 contain the following subsections:

1. **Description:** A general description of the use case with details material to this analysis.
2. **Discovery/Inventory:** Analysis on how to discover instances of this use case and/or a recommendation as to what data elements should appear in a related inventory.
3. **Migration Considerations:** The key factors to be aware of when planning a migration of cryptographic algorithms, and as preparatory elements which should a priori be in place in order to be cryptographically agile.
4. **Cutover Strategy:** Direction and analysis of what is involved in actually implementing the migration.
5. **Governance:** Elements that should be in place to assist with the overall governance of the migration, preparatory work to be cryptographically agile, and post-migration monitoring.

I.1.3 Scope of this Annex

There are certain considerations that are ever-present when dealing with crypto-agility or the considerations of a migration. These include:

- Budgeting and resourcing;
- Project management; and
- Executive and staff communication.

These considerations tend to be non-technical in nature and were not analyzed in this exercise, although it may be an interesting exercise (for future work) to determine what these considerations entail.

Note that some non-technical considerations did arise directly as part of this exercise (e.g., third-party governance) and they are explicitly mentioned where appropriate. Also, as there may be multiple transitions to different cryptographic technologies, it may be worthwhile to map out future transitions.

I.2 Crypto Agility Use Cases and Findings

Thirteen different use cases are described in the remainder of this Annex, in the following subsections:

I.2.1: Public Certificate Authority (CA) / Public Key Infrastructure (PKI);

I.2.2: Private Certificate Authority (CA) / Public Key Infrastructure (PKI);

I.2.3: End-Entity Certificate Requirements;

I.2.4: TLS Connections to General External Client Browsers;

I.2.5: Vendor Appliances Establishing TLS Connections;

I.2.6: Internally Developed Applications;

I.2.7: Code Signing;

I.2.8: Database Encryption;

I.2.9: Centralized File Encryption;

I.2.10: Tactical File Encryption;

I.2.11: Full Disk Encryption;

I.2.12: SSH Connections for Administration;

I.2.13: SAML or Other Federated Identity.

I.2.1 Public Certificate Authority (CA) / Public Key Infrastructure (PKI)

I.2.1.1 Description

This use case will cover the crypto-agility aspects of the signing algorithm for certificates issued by a public Certificate Authority (CA) from the perspective of a subscriber. In particular, the CA will be one which has been designated by the organization as being allowed to issue certificates on domain names belonging to that organization. The actions of an individual entity during the certificate lifecycle are handled in Section I.2.3 of this Annex.

For concreteness, we will assume that there will be a simple three-level hierarchy:

- root -> intermediate -> end-entity.

I.2.1.2 Discovery/Inventory

It is important to have a list of all CAs from which the organization can obtain certificates. For each such CA, the following should appear in an inventory:

- The different types of certificates available (e.g., Class 3 server, Class 2 client, Extended Validation);
- For each type, an inventory of the actual certificates which have been issued. For each certificate, this should include:
  - Fully Qualified Domain Names (FQDNs) and Subject Alternate Names (SANs), or other identifying information appropriate to the certificate use case;
  - Certificate Expiry;
  - Algorithm, fingerprint and/or public key;
  - Locations where the CA certificates exist.
- Certificate Revocation List (CRL) or Online Certificate Status Protocol (OCSP) server.

Most public CAs provide an inventory of the certificates it issues. Alternatively, there are scanning tools available that can find certificates in use on different systems within an environment. Organizations should balance their security needs with their needs for usability and availability when considering such automated tools.

I.2.1.3 Migration Considerations

Organizations are dependent on the public CA migrating their service to a new certificate signing algorithm. The following are dependencies that the public CA would be expected to address as part of the migration:

- If a new algorithm is required, establish or stand up a new root CA certificate with the new type of cryptography and make it publicly available;
- If a new algorithm is required, stand up a new intermediate CA certificate with the new cryptography; note that the intermediate CA may migrate at a different time than the root CA;
- Deploy a new Certificate Policy (CP) or Certificate Practice Statement (CPS) that describes the specifics with respect to how the cryptography works (e.g., hybrid, placing it in extensions); for example, in X.509 certificates it is common practice for new data elements to go into an X.509 extension of the certificate although this is not a requirement;
- Detail the extent to which the end-entity certificates are backward compatible (i.e., verifiable by entities expecting the older format);
- Specify any cross-signing hierarchy (e.g., the typical cross-signed hierarchy depicted in Figure I-2 on the next page);
- Update the CP/CPS to specify how the CRL or OCSP responses will be signed and provided for both root and intermediate CAs;
- Be responsible for the legitimacy of the CA via audits (e.g., Web Trust audits);
- Detail any specification the CA is doing from a requirement's perspective with respect to the subscriber Registration Authorities (RA).

With these items having been addressed, an organization preparing for a migration should do the following:

- Verification of new CA hierarchy and guidelines:
    - Understand what the new guidelines mean for the organization;
    - Understand how the cryptography (e.g., hybrid) is implemented and how it will affect general applications and clients that use those applications;
    - Communicate the implications to authorized subscribers to Certificate Authorities.
- Registration Authority (RA):

- Ensure that what the Registration Authority (RA) is doing, is considered in making yourself compatible with the new procedure or algorithm;
- The RA may need to use new asymmetric key pairs according to specifications and compatibility (e.g., TLS certificates to use new algorithms). This will apply to RAs which are used for both manual and automated renewal.

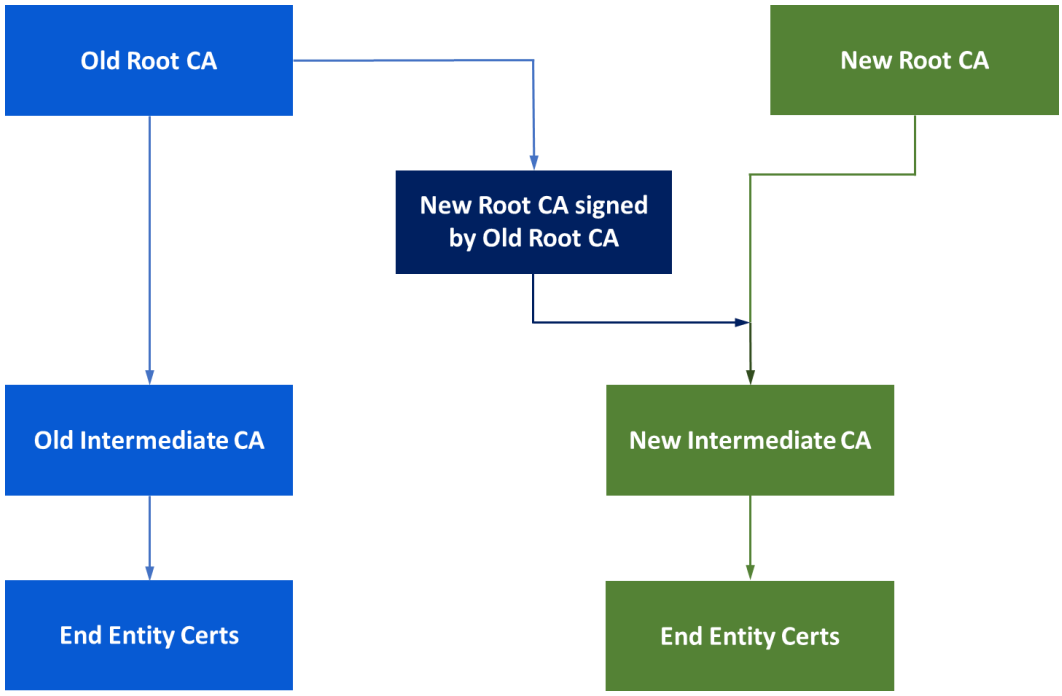


Figure I-2. Typical Cross-Signed Hierarchy

I.2.1.4 Cutover Strategy

Cutting over CAs to new cryptography has been done successfully in the recent past. Many CAs transitioned from 1024-bit RSA keys to 2048-bit RSA keys or to ECC, and then later signatures were transitioned from using SHA1 to SHA2. We suggest the following cutover strategy for organizations, modelled on those successful transitions.

- Once the new cryptographic algorithms and/or certificate profile are known, any systems that will interact with the new certificates as signers, verifiers, or servers must be updated to handle the changes; this could potentially occur through a software update, configuration change or other method;
- Once the new public CA has been stood up, a transition period and cut-off point should be established; the timelines may depend on the CA, as they may have a requirement for transitions to be completed by a certain time; these timelines may

be different for the root and intermediate CAs and may be subject to CA policy or CA/B forum baseline requirements;

- During the transition period, any new certificates and certificate renewals should be done under the new CA, provided the relevant systems have been upgraded;
- Continually monitor the progress of system upgrades, to ensure that all systems will be able to transition before the cut-off:
    - Develop a plan for any systems having difficulty cutting over.
- As the transition cut-off approaches, any certificates still using the old cryptography should be renewed outside of the regular schedule; ensure that any valid certificates using the old cryptography are revoked.

I.2.1.5 Governance

The overall governance for the migration would incorporate existing certificate governance of renewal upon expiry as standard process. It would additionally include the following:

- Monitoring of systems and their certificates to keep track of which ones have migrated and which ones have not;
- Removal of old CAs from active Trust Stores when the migration has completed;
- Proper audit mechanisms to ensure compliance.

I.2.2 Private Certificate Authority (CA) / Public Key Infrastructure (PKI)

I.2.2.1 Description

This use case will cover crypto-agility aspects related to the setup and use of a private CA managed internally by an organization. A private CA is the authority for a public key infrastructure (PKI) for a specific organization or a closed network of peers, and typically issues certificates that are intended for use only by the organization or its peers to which it is assigned. The actions of an individual entity during the certificate lifecycle are described in Section I.2.3. For simplicity, we will assume that there will be a simple three-level hierarchy:

- root -> intermediate -> end-entity.

In many ways, the crypto agility of the signing algorithm used by a private CA is similar to that of the public type. The main difference is that the organization now controls much, if not all, of the considerations involved. This is reflected in the considerations below.

There are various types of CAs which fall into the category of private, including:

- **Inspection CAs:** These are CAs which are specifically in place to intercept incoming and outgoing content for the purpose of inspection of traffic. This would include performing content filtering and malware detection. This CA is typically an intermediate CA off of an enterprise-accepted CA.
- **Special Use CAs**: Some applications require their own CA in order to function. These are often limited in scope to the devices involved with the application. Examples include special purpose hardware such as Encryption PIN Pads (EPPs) on an ATM or POS device, routers from a specific vendor, or IoT implementations such as cameras or display monitors.

I.2.2.2 Discovery/Inventory

It is important to have an inventory of the different private CAs that exist within an organization. Enterprise-wide CAs are generally well-known, but special-use CAs can sometimes be embedded and hidden from normal business operations. Discovering special use CAs may require either consulting application vendors or performing a network scan for certificates (e.g., scan ports 443, 1443, 8443 for HTTPS, other ports for TLS).

For each private CA, the inventory outlined in Section I.2.1.2 for public CAs should be followed. If the CA is internally hosted, then it would be important to also have information on the CA's operating infrastructure such as:

- Servers;
- Hardware Security Modules (HSMs);
- Network location;
- Location of CA private key including online or offline backups;
- CRL location.

For a special-use CA, it would also be important to have an inventory of devices that leverage the special-use CA.

I.2.2.3 Migration Considerations

Since this is an internal CA, it is assumed that the organization controls all aspects of its setup for crypto agility, including the following:

- Decide on the new type of cryptographic algorithms that will be used by the CA for signing certificates and can support the organization's needs with respect to designated factors (e.g., latency, throughput, storage space, etc.);

- Decide how the new cryptographic algorithms will be realized within the CA and its certificates (e.g., certificate extension fields);
- Decide on the cryptographic algorithms to be used for the CRL signing;
- Decide on how the CA is structured (e.g., cross-signed with old root);
- Establish the infrastructure for a Root CA (e.g., HSM, offline device, cryptography card) which is compatible with the new crypto;
- Create the new root CA certificate and export for backup purposes as appropriate;
- Establish the infrastructure for an online issuing intermediate CA (e.g., server, VM, HSM, networking capabilities, etc.) which are compatible with the new cryptographic algorithms;
- Create the new intermediate CA certificate and perform any additional operations such as cross-signing;
- Make the CA certificates available to the organization and/or push them out to the requisite systems;
- Establish or modify the Registration Authority (RA) setup to leverage the new cryptographic algorithms;
- Ensure provisioning protocols such as SCEP or PKCS #10 are able to leverage the new certificates;
- For a special-use CA that may be specific to a particular service or hardware, ensure that the devices leveraging this CA are compatible with the new hierarchy; this may require an upgrade to a different generation of device.

I.2.2.4 Cutover Strategy

Once established, the new CA would need a cutover strategy similar to that of the Public CA.

- Develop and manage a cutover strategy for moving from the old CA to the new CA:
  - Establish a cut-off point for a defined transition period;
  - Establish and communicate organizational guidelines for cutover;
  - Establish oversight for removal of old CAs.

An inspection CA would be similar to that of a standard private CA, with the exception that it is likely dependent upon the private CA from which it was signed. It can be treated as another intermediate issuing CA of the private CA.

Special-use CAs are vendor dependent. It would be up to the vendor to determine or recommend a cutover strategy. It could either be gradual or all at once, depending on the options provided by the vendor and the characteristics of the business use case.

I.2.2.5 Governance

Governance for this use case would be similar to that of the public CA/PKI. For special-use CAs, there would need to be an additional level of governance to track the different implementations and assessing each one's ability to migrate. Note that audit requirements here would be internal unless a specific use requires external oversight.

I.2.3 End-Entity Certificate Requirements

I.2.3.1 Description

This use case will cover crypto agility in the use of certificates throughout their lifecycle from an end entity perspective. This would include Certificate Signing Request (CSR) generation and certificate loading as well as revocation, and distribution, but not be tied to a protocol such as TLS.

Many organizations leverage a Content Delivery Network (CDN) (e.g., Akamai, AWS CloudFront) to filter the content that enters their systems. These often leverage certificates to assist with facilitation of services. From the perspective of this document, they will simply be considered as an end-entity requiring a certificate.

I.2.3.2 Discovery/Inventory

Having an inventory of end-entity certificates should be a requirement. Any inventory should include:

- Certificate details (e.g., common and subject alternate names, expiry date, etc.);
- CA it was obtained from;
- Locations where this certificate is used (i.e., where private key exists);
- Owner or accountable officer of certificate or appliance(s) on which it exists;
- Whether or not this certificate is associated to a CDN.

I.2.3.3 Migration Considerations

- CSR Generation:
    - The appliance on which the certificate will reside must have access to a tool which will create the Certificate Signing Request (CSR) that is compatible with the new CA requirements:

        - If multiple hierarchies exist with different cryptographic algorithms or specifications, the CSR generation tool(s) will need to have these capabilities/flexibility.
    - In cases where the CSR is generated on a different appliance:
        - The device that generates the CSR must have a tool compatible with the new formats and algorithms;
        - The appliance on which the certificate resides must be able to import the response to the CSR including the private key in the new format.
    - The tool used to generate the CSR must support new protection mechanisms for the file formats of the old and new CSRs and responses (e.g. .pem, .pfx):
        - It must be able to import private keys from certificates in the old format.
    - The asymmetric key pair must be generated using the new cryptography;
    - The key store for the new private key must be compatible with new format:
        - Hardware key stores must ensure the HSM vendor can support this:
        - Software key stores must leverage any new cryptographic algorithms for key store protection and be able to import private keys and certificates for the new cryptographic algorithms.

- Certificate Distribution and Loading:
    - If the CSR was created on a different device, the distribution method used to move private keys must be compatible with any new private-key format;
    - The mechanism used for distribution of the certificate must be compatible with the new certificate:
        - This may be a manual process or it may be an automated process.
    - The receiving device must be able to load the new hierarchy into its trusted store. Depending on the type of device this may mean a cross-signed hierarchy;
    - The receiving device must be able to properly load the certificate;
    - The receiving device must be able to verify the appropriate Certificate Revocation List (CRL) with its new signature;
    - At an appropriate time, the device must be able to remove the old hierarchy and/or switch away from cross-signed hierarchy.

I.2.3.4 Cutover Strategy

Most of the work in a cutover is on the part of the CA. The end-entity can cutover whenever it sees fit if it is in the window given by the CA.

The key points an end-entity must take into account when cutting over are:

- Ensuring that it can support the new cryptography;
- Ensuring that the entities that consume its certificate can support the new cryptography.

I.2.3.5 Governance

Overall tracking of end-entity readiness to migrate is usually handled by the operational entities of the organization. Each end-entity needs to take it upon itself to ensure that cutover will be successful.

I.2.4 TLS Connections to General External Client Browsers

I.2.4.1 Description

This use case will cover the use of certificates in a Transport Layer Security (TLS) connection to general external client browsers. This use case is discussed from the point of view of connections to the browsers. The properties of the server establishing the TLS connection are discussed in Section I.2.5.

I.2.4.2 Discovery/Inventory

An inventory of the different domains and subdomains accepting TLS connections is important.

In order to support crypto-agility and ease of migration, the main inventory item for each domain/subdomain is a list of the different browsers that are supported. This would include:

- Browser name;
- Browser version;
- Approximate number of connections for each name and version (e.g., daily average);
- Any interesting factors to note about the browser itself.

This data can be discovered through traffic monitoring or log analysis.

I.2.4.3 Migration Considerations

The browser community and/or public CAs (likely through the CA/Browser Forum) would be expected to address the following as part of the migration:

- Provide direction to external browsers to leverage the new TLS protocol and/or its new cryptography; (note: any changes to the TLS protocol itself would be managed by the Internet Engineering Task Force or equivalent standards development organization);
- Provide direction to external browsers to leverage the new certificate capabilities and CA hierarchies;

- Manage the upgrade path of most browsers in use by the public.

I.2.4.4 Cutover Strategy

The organization responsible for the server would then need to perform the following tasks:

- Extent of backward compatibility needed:
  - Understanding of how the new certificates and protocol will affect older browsers and technology;
  - Amount of external client on old browsers which will be degraded or unusable;
  - Plan to deal with handling those using old technology;
  - Determine if both old and new cryptographic algorithms can be used simultaneously.
- Anticipate and plan around any downtime;
- Establish a cut-off point for a defined transition period.

I.2.4.5 Governance

Governance requirements are very basic. Keep track of the different sites for which this applies and how well they are cutting over.

I.2.5 Vendor Appliances Establishing TLS Connections

I.2.5.1 Description

This use case will cover the end-entities that implement a TLS connection and that have been supplied by a third-party vendor. Note that whether the TLS connections are one-way or mutual is immaterial.

I.2.5.2 Discovery/Inventory

An inventory of vendor appliances implementing TLS connections should be a requirement. Any inventory should include:

- Inventory of appliances/servers/load balancers or equivalent used in this use case;
- Vendor for each of these appliances, etc.;
- Hardware, software, firmware versions.

I.2.5.3 Migration Considerations

The appliance in place must be able to perform the required functions, namely:

- Upgrade to the proper version to leverage the new cryptographic algorithms and support both old and new protocols as appropriate;
- Perform the requisite certificate provisioning and loading functionality as described in the End-Entity use case in Section I.2.3;
- Perform the TLS connection using the new cipher suites as determined in the TLS protocol (Note: any changes to the TLS protocol itself would be managed by the Internet Engineering Task Force (IETF) or equivalent standards development organization);
- Conduct proper testing of functionality.

I.2.5.4 Cutover Strategy

TLS connections are typically non-persistent, so new connections can start fresh. The following must be considered in any cutover strategy:

- Extent of backward compatibility needed (in addition to the considerations outlined in Section I.2.4.4):
    - The appliance must be able to support old and new cryptographic algorithms and old and new CA hierarchies simultaneously.
    - In the cases where the appliance can only support one root CA, it is preferable to use a cross-signed CA until all connections have been migrated.
    - Assessment of the impact of connections which will be degraded or unusable.
    - Plan to deal with connections which use old technology
        - If industry wide (i.e., outside the purview of the organization), follow plan as for external browsers
        - Otherwise, keep updated on the migration status of connections
        - Have a plan to deal with those who cannot/will not migrate
- Anticipate and plan around any downtime
- Establish a cut-off point for a defined transition period
- Remove old roots after transition
    - Move away from cross-signed hierarchy if applicable.

For non-persistent TLS connections, it is important to consider how to manage session resumption and data persistence. A proper cutover strategy would need to be formed.

I.2.5.5 Governance

There are some additional governance requirements:

- The appliance itself will often be supplied by a vendor. Processes must be in place to ensure:
  - The vendor is aware of the vulnerability/security risk
  - The vendor understands the associated risks as it applies to their product
  - The vendor has a roadmap to make their product secure against the vulnerability
  - The vendor takes all considerations from Sections I.2.5.1 to I.2.5.4 into account.

I.2.6 Internally Developed Applications

I.2.6.1 Description

This use case will cover aspects in dealing with internally developed applications. The general assumption is that whatever pipeline used to produce these applications (e.g., SDLC, DevOps, CI/CD) will not structurally change. It is the artifacts within these pipelines which will change. Examples of artifacts include: a crypto library (openSSL) in your application, code base, application code, secrets such as an SSH key.

I.2.6.2 Discovery/Inventory

For existing applications, the following are important to list in an inventory:

- Application name and version
- Framework or platform on which it is built
- Programming language(s)
- Software Bill of Materials (SBoM) in order to work with submodules [19]
- Cryptographic Bill of Materials (CBoM) or at least a list of cryptographic libraries used [20]
- Dependencies and constraints (e.g., throughput, hardware, latency)
- Appropriate software documentation
- Any instances of hard-coded cryptographic assets
- List of authentication mechanisms in place.
- List of clients for the application (sanitized as appropriate)

[19] Software Bill of Materials (SBOM) | CISA

[20] GitHub - IBM/CBOM: Cryptography Bill of Materials

Application inventories can be obtained through different activities:

- Manual list
- A general material scan
- A cryptography-specific scan of source code or binaries (e.g., using a code scanning tool such as BlackDuck, Vericode that search for vulnerabilities).

I.2.6.3 Migration Considerations

In order to prepare software development for crypto agility, it is important to make sure the development pipeline can handle it. The following are required:

- The pipeline has access to appropriate cryptography-compatible libraries for development
- Cryptography related vaulting and data-retrieval mechanisms are compatible with the new cryptography, including:
    - Passwords, secrets, or other authentication-credential retrieval methods
    - Certificates and private keys
- Vaulting and retrieval mechanisms support new cryptographic algorithms
- There is compatibility with other tools involving cryptographic assets, such as automated certificate management
- Pipeline pieces are each separately compatible with new cryptographic algorithms where cryptography is applied

As a general rule, a consistent development or application stack to make migration (as well as many other things) much easier.

The pipeline should maintain its normal lifecycle with these changes, and should now be set up to develop new applications using new cryptography in a crypto-agile way.

As for migrating existing applications, the following would need to be done:

- Triaging and prioritization:
    - As there will be many applications to migrate, there will need to be some sort of prioritization. This prioritization will be based on the internal requirements of the organizations, such as:
        - Risk of breach;
        - Availability risk;
        - Public accessibility;
        - Attack surface, ease of access;
        - Difficulty of upgrade;

        - Time to migrate;
        - Refresh cycle;
        - Application lifecycle;
        - Client support.

- Planning:
    - This stage is where the plan to change the actual application is developed. During planning, the following factors should be considered:
        - Understanding the requirements of the application (e.g., low-bandwidth, large amount of data processing, etc.);
        - Dependency chain of migration – migrating an application may depend on migrating the modules on which it is based first (some of which are made by vendors); this includes crypto libraries;
        - Understanding how to integrate new code into the existing code base;
        - Understanding the extent to which the application can be made crypto-agile;
        - Understanding how the application will communicate with other applications or infrastructure;
        - Understanding how these changes will affect the system as a whole;
        - Whether or not developers have been appropriately trained to work in a crypto-agile fashion.

- Implementation:
    - Send the application back through the pipeline to get a migrated application.

- Testing:
    - Normal testing procedures should apply, including first testing in lower environments and performing regression testing.

- Deployment, Operations, and Monitoring:
    - Follow a standard deployment, operations, and monitoring cycle.

Finally, we would note that developers would need to be trained on crypto agility. At minimum, the following should be included: [21]

- Hard-coding elements reduce agility, including:
    - Cipher suites;

[21] https://learn.microsoft.com/en-us/archive/msdn-magazine/2009/august/cryptographic-agility

  - Buffer size;
  - Paths;
  - Hostnames;
  - Passwords;
  - Secrets;
  - Configuration items;
  - Cryptography provider libraries;
  - Certificate fields;
  - So as to avoid hard-coding of cryptography, there needs to be a layer of cryptographic abstraction;
  - Use forward-leaning libraries.
- Document the developer('s) code:
  - Where the certificates are;
  - How certificates are used;
  - Which libraries are used;
  - Instances of cryptographic algorithms;
  - General standard developer documentation.
- Hardening requirements with respect to crypto agility:
  - Get rid of old ciphers / old libraries you do not want to be used;
  - Reduce side channels;
  - Harden cryptographic implementations (i.e., have the implementations do exactly what we want them to do, such as to provide confidentiality, or integrity, and to not do anything extra).

Even with the best of training and education, elements which will prevent crypto agility will inevitably occur in code. One of the main ways to account for these instances is to implement code scanning. In terms of crypto agility, it is again assumed that the structure of scanning will not change. Only the content of scanning will change. This will now include scanning for:

- Cryptography implementations;
- Interoperability and backward compatibility;
- Downgrading cryptographies;
- Hard-coding of parameters or data.

In terms of what is scanned, the following should be considered:

- Source code;
- Binaries;
- Input/Output;
- Containers;
- Infrastructure;
- Code repositories;
- Pipelines.

Output from code scanning should be handled with usual process including filtering out false positives, ranking the severity of a finding, and working to remediate.

I.2.6.4 Cutover Strategy

The cutover strategy will follow the normal SDLC, CI/CD, DevSecOps processes of the organization. An organization may also need to consider the upgrade path for clients.

I.2.6.5 Governance

The standard governance mechanisms relevant to internal development of application would still apply. However, there are additional steps which would be useful in ensuring crypto agility:

- Incorporating crypto agility in risk management processes to determine risk related to internally developed applications;
- Leveraging new avenues through which to find issues with regards to crypto-agility such as crypto-agility-related bug bounty or red teaming;
- Developing a scale which measures the extent to which an application is made to be crypto-agile.

I.2.7 Code Signing

I.2.7.1 Description

This use case deals with the structure of code signing within an organization. The main entities involved are the Code-Signing Requestor (such as a developer), the Code-Signing Service, and the relying parties or Code-Signing Verifiers (such as the operating system of the end user). The Code-Signing Service may have a Timestamp Service as part of its service. There may be a separate CA service which provides the signing certificate. This use case is regarded on its own and separate from the development cycle as it does not relate to the development of applications, but instead its own service involving cryptography.

Code signing uses the basic model shown in Figure I-3, below.

The cryptography spans from the Code-Signing Service to the Code-Signing verifier as it is usually a digital signature of some kind.

I.2.7.2 Discovery/Inventory

The inventory should list each signing service. For each code signing service, the inventory should include:

- Signing service metadata (e.g. name, vendor (if applicable), and version);
- The signing certificate and developer private key;
- The valid requestors;
- If possible, the code signing verifiers and their capability to support new cryptography and be backward compatible.

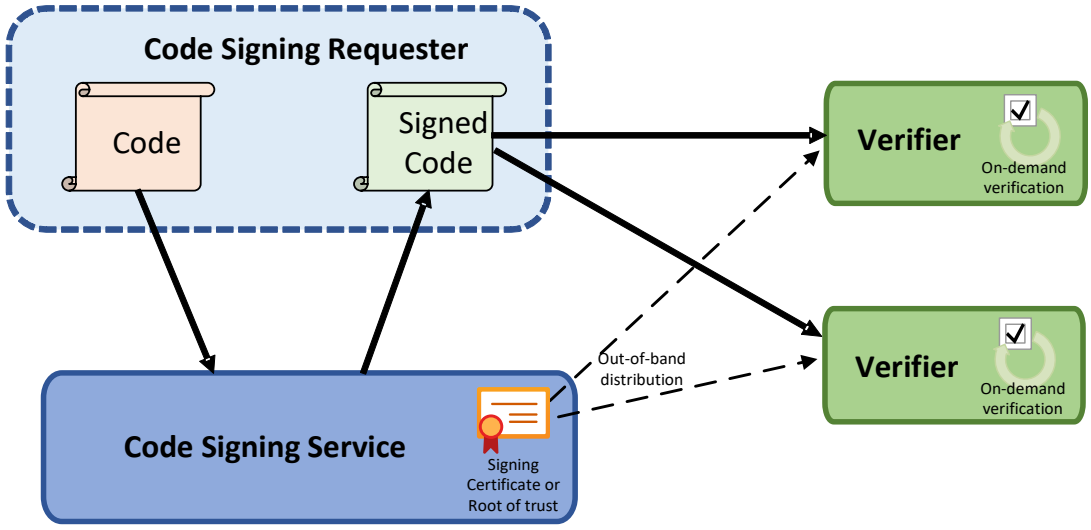


Figure I-3. Basic Code-Signing Model

I.2.7.3 Migration Considerations

Migrating to new cryptography will involve:

- Ensure that the Code Signing Service is migrated to be crypto-agile. If the service is in-house, this would include the following:
  - Choosing the new signing algorithm;
  - Defining the signature structure taking into account backward compatibility;

- Being able to handle the new Certificate Lifecycle as described in the End-Entity Use Case (Section I.2.3).

- Migrating each separate Verifier:
  - Being able to accept the new certificate hierarchy;
  - Being able to store the new signature and parameters;
  - Being able to implement the new verification requirements.
- Assessing the verifier's ability to be backward compatible:
  - If a hybrid signature is used, it is important to note whether or not the verifier will be able to accept the new signature before it is migrated.

I.2.7.4 Cutover Strategy

If the Code Signing Service is in-house, have a cutover strategy similar to the TLS Connections with External Browser Use Case described in Section I.2.4.

The difference between these use cases is that code signing is:

- Persistent – the old code must be remembered for backup purposes during migration;
- Not real-time – the code verification occurs when new code is deployed, which could come at a later point than when it is signed.

I.2.7.5 Governance

Governance requirements are basic. Keep track of verifiers and their current state towards migration.

I.2.8 Database Encryption

I.2.8.1 Description

This use case deals with databases or other structured data environments which have been encrypted. Typically, the method of cryptography is Transparent Data Encryption (TDE) which will encrypt the entire database in totality or encrypt the columns of the database tables individually. In the latter case, the data within a column of a table may be encrypted and use different encryption from that used for the data in other columns of the same table or may not be encrypted at all.

The encryption is referred to as *transparent* since the data is encrypted as it is committed to the database and decrypted as it is accessed. The encryption is thus transparent to any application which is accessing the data. The encryption is typically symmetric, and the key is held in or near the database itself.

One important assumption which we are making is that the encryption schema of the database (e.g., encryption policy) will not change. Columns that are encrypted now will continue to be encrypted. Columns that were not encrypted will continue not to be.

I.2.8.2 Discovery/Inventory

In terms of an individual database, it would be important to have the following information available:

- The database purpose;
- The database product, version, platform;
- The database schema;
- Column metadata (i.e., what each column contains);
- The type of encryption applied to each column.

I.2.8.3 Migration Considerations

The main considerations when migrating the data are as follows:

- Migrating the database product itself:
  - The database vendor would need to upgrade its product to allow the new encryption technology;
  - The vendor would need to support the new algorithms;
  - The vendor would need to state changes to considerations which its clients would need to take into account (e.g., disk space expansion, latency);
  - The vendor would need to state any effect that this would have on applications which access them; (note: as the encryption is transparent, there should be no impact theoretically, but any deviation from this should be communicated).
- For many databases, the amount of data is massive, and therefore the migration will be a monumental task; it should be noted that this is somewhat an existing problem today in terms of key rotation;
- How the change in cryptography may affect data format or size limits;
- Migration across database replication systems;
- A proper plan would need to be developed with appropriate contingencies.

I.2.8.4 Cutover Strategy

When cutting over to the new encryption, the foremost decision to be made is whether or not to adopt an all-at-once approach or take a forward-looking approach.

In order to do an all-at-once approach, the following must occur:

- Set up an alternate database instance;
- Decide whether or not to hold off on data commits for the migration period of time or simply keep track of changes after the migration begins;
- Perform the translation of the data, taking into account any commit changes after migration starts;
- Switch to the new database at a designated point in time;
- Monitor application use and access and be ready to rollback if necessary.

The forward-looking approach will encrypt data using the new encryption as it is committed to the database. Note that this will work in cases where encryption occurs at a row level. It may not be feasible depending on the product or if complete encryption is used. The following must occur:

- Assess whether this approach is possible given the product, schema, and encryption pattern;
- Understand the performance impact, if any, in taking this approach;
- Develop a plan for existing rows which are not re-committed, including whether to do a bulk translation when their percentage drops below a certain threshold.

In all cases, it will be important to manage multiple copies of a particular database and manage how these are copied, put into service, and deleted. One also needs to consider how long to maintain legacy encryption keys for database backups.

Another important consideration is to determine the extra cost in resources for potential extra storage space needed to accommodate the new algorithms. This would include the loss of de-duplication and compression capabilities

I.2.8.5 Governance

The governance requirements would be very basic. Overall tracking of readiness for each database implementation would be needed, but each database implementation would be responsible for its own migration and ability to be crypto-agile.

I.2.9 Centralized File Encryption

I.2.9.1 Description

This use case deals with unstructured data such as file servers, Network-Attached Storage (NAS) shares, document repositories, etc., but where the encryption has been centralized. This use case does share some similarities with the structured Database Encryption use case. The encryption would be considered transparent. File encryption primarily uses symmetric encryption, but asymmetric encryption may be occasionally used.

I.2.9.2 Discovery/Inventory

An inventory in this case, would consist of:

- File or filesystem name;
- File type;
- File sensitivity label (i.e., level of confidentiality);
- Location;
- Encryption technique.

Quite often a data discovery tool can be implemented to find and determine these files (for example: IBM Guardium, Varonis, Microsoft's MIP).

In addition, where appropriate, the key hierarchy should be included in the inventory as well.

I.2.9.3 Migration Considerations

The considerations would closely parallel those of the Database Encryption use case. In particular, the decision of all at once vs a forward-looking approach would need to be decided. A few notes to highlight are:

- The product itself must be upgraded to the new technology. This would include the key store if it is internal. The same applies if it is an external key store, although the integration between the two would need to be included in the migration.
- Due to the unstructured nature of the file server, the forward-looking approach would likely be much more feasible.
- While it could be another monumental effort, access to centralized file storage is not typically high-availability or in real-time. Thus, the migration could happen gradually.
- There are cases where the administrator dictates what is encrypted globally. But there may also be cases where the user decides which files are to be encrypted.

I.2.9.4 Cutover Strategy

If a forward-looking approach is decided upon, the centralized authority could encrypt all new files created and encrypt existing files the next time they are saved. A cleanup activity to decrypt/re-encrypt the files which have not been accessed since the migration started could occur at a later time.

If an all at once approach is used, then there may be some downtime needed while files are decrypted and re-encrypted. Note that this activity could happen in batches.

I.2.9.5 Governance

This use case depends on having an accurate inventory of files and locations. This is not always feasible with 100% accuracy. However, a data discovery and retention program may be of great use for this use case.

One major issue with this use case is the idea of crypto-shredding (defined as the deletion/destruction of an encryption key for the purpose of making data inaccessible). From a governance perspective, it is often the case that crypto-shredding is seen as a valid method of data destruction. This is useful in cases such as cloud environments where there is essentially no control over data. However, the idea that cryptography can be broken brings this paradigm into question.

I.2.10 Tactical File Encryption

I.2.10.1 Description

This use case handles unstructured file encryption, but where the service is not centralized such as PGP, secure file zipping, or other file encryption tools. In this case, instead of a centralized key server, keys will be disparately located. It is possible that a key-escrow service is in place in this case.

I.2.10.2 Discovery/Inventory

As with the centralized case, a data discovery scan (using the same tools as described in Section I.2.9) would yield a similar inventory. The one item that is different is that the encryption key may not inherently be centralized, so there may be no determination as to where it would be, so it cannot be listed in the inventory.

If a key escrow service is used, this service would have an inventory of the keys used and the files to which they apply.

I.2.10.3 Migration Considerations

Since this use case is typically for ad hoc file encryption, the migration would be split into all of the individual instances of the encryptions. In particular, for each one,

- The appropriate encryption tool would have to be upgraded to a compatible version;
- The appropriate files would have to be migrated;
- The appropriate keys would have to be placed in an accessible location.

One of the biggest considerations is being able to find and perform all of these migrations. It is not always clear if this would be possible in a practical setting.

When key escrow is used, the following additional considerations should take place:

- The key-escrow service, including the key-recovery agent, should be upgraded;
- The keys stored by the service and their corresponding files should be migrated;
- The key lifecycle should continue to be monitored.

I.2.10.4 Cutover Strategy

Since files are individually encrypted, the migration can happen any time at the discretion of the file owner.

I.2.10.5 Governance

Governance options include:

- Shifting these instances to the Centralized File Encryption use case;
- Enforcing key escrow;
- Setting up a tracking program.

I.2.11 Full Disk Encryption

I.2.11.1 Description

This use case deals with the encryption of the storage space of a particular device. Note that this is at a lower level than that of a database or file encryption.

This could apply to individual user-level devices such as a laptop, mobile device, or Internet of Things (IoT) device. However, it could apply to larger appliances, such as servers which house databases, as it may be seen as an alternative to Database Encryption.

I.2.11.2 Discovery/Inventory

An inventory of devices is essential to this use case. It would include:

- Device name, Operating System (OS), other device information;
- If appropriate, person to whom it is assigned;
- Type of encryption technology being used;
- How the encryption key is protected or regenerated.

I.2.11.3 Migration Considerations

The following considerations are involved:

- The tool used to encrypt needs to be upgraded to support the new algorithm;
- Storage-space requirements and cost would need to be taken into account;

- The compression and de-duplication issue would still be present. De-duplication refers to the removal of duplicated data streams within large data sets. Both compression and de-duplication are used to reduce data size. However, encryption of large data sets typically render compression and de-duplication ineffective;
- Strength of the encryption key protection (e.g. password strength, biometric, multi-factor);
- The availability and use requirements would need to be taken into account as they may be different.

I.2.11.4 Cutover Strategy

The method for migration would be dependent upon the type of device which is being migrated.

- Laptop or IoT devices could be made to install updates by an administrator and translation could occur as part of updates. This would be dependent upon either inventory or access edict.
- Servers could be in situation similar to Database Encryption in terms of availability and real-time requirements. The 'all at once' approach would likely be the only option.
- Data backups would be offline by nature and so could be translated offline. The keys are often held by the client, in which case the key vault would need to be upgraded and there would need to be coordination.

I.2.11.5 Governance

Governance would again be very basic. A general tracking program is a good idea, but each instance would be responsible for its own migration.

I.2.12 SSH Connections for Administration

I.2.12.1 Description

The SSH protocol allows users to remotely connect to a resource, such as a server or appliance. Users may use public key cryptography, single sign-on methods, or passwords to authenticate to the resource. Public-key-based authentication requires generating an asymmetric key pair and storing the public key on the resource. Then, when the user attempts to connect to the resource, they are challenged to prove that they hold the private key as part of the authentication process. Server authentication may be implicit or explicit, but the server will have its own asymmetric host key regardless.

While SSH keys may sometimes be tied to a certificate, they are often not. Thus, connections are pairwise and can occur between client and server with no interaction or oversight from other entities. Clients will often cache the fingerprint of a server's public key to trust it for future

connections. This often creates a "wild west" situation where SSH connections can occur from anywhere and be active at any time. That includes keys still being valid after many years of inactivity.

Organizations often find themselves in one of four states, listed here in increasing level of maturity:

- ***The "Wild West":*** SSH user keys are completely decentralized with no or minimal centralized involvement;
- ***Centralized Tracking:*** SSH keys are created and exist as in the "Wild West" situation, but there is a centralized inventory to keep track of where the SSH keys exist;
- ***Centralized Management***: SSH connections still occur between client and server, but in addition to tracking, a centralized service will create, distribute, and renew SSH keys to client and server. There is likely some level of automation in this stage;
- ***Centralized Operations:*** SSH connections themselves are managed through a centralized platform so that not only are keys distributed to client and server, but the connection itself will run through the platform. There is a greater degree of automation in this stage. The centralized entity may be part of a Privileged Access Management (PAM) environment.

It should be noted that even in the most mature state, there is still the possibility that "wild west" SSH connections will occur.

The SSH protocol also uses key agreement and symmetric key cryptography to provide confidentiality. The confidentiality properties of SSH have not been considered in this version and will be addressed in a future revision.

SFTP is a file transfer protocol which usually leverages SSH to make an initial connection. SFTP as a protocol has no cryptography, rather it assumes it is run over a secure channel. Thus, the crypto-agility consideration for SFTP are the same as the considerations for the underlying protocol securing it, which is most often SSH.

I.2.12.2 Discovery/Inventory

In order to be crypto-agile, an organization must at least be able to identify where its SSH keys are. Hence, it is a requirement that the organization must at least be in the ***Centralized Tracking*** state. This would mean that they would have a centralized inventory of SSH keys with the ability to discover new ones which may pop up. This discovery can be performed with currently available scanning tools or by monitoring connections to centralized SSH servers.

In terms of inventory, clients and servers could theoretically be any machines, and keys could exist anywhere on those machines, so a complete discovery may be very difficult. As the

purpose of SSH is to establish access to the server, we will focus our attention on server-side SSH key discovery.

At minimum, an inventory must include the following details from server-side resources:

- Server metadata (e.g., server name, URL, IP address, network location, etc.);
- Server private-key metadata (e.g., algorithm, key length);
- For each user for which an SSH connection will be accepted:
    - Metadata of clients the user has used (e.g., client name, client version, operating system, IP address, network location, etc.);
    - User public keys or hash of public keys (including algorithm used);
    - User access level (e.g., root, user, permissions);
    - Latest time of access.
- In the case of SFTP, it may also be useful to list:
    - Landing zone location for files on the server.

This information can typically be obtained from current discovery tools.

Performing a scan and inventory on clients may be infeasible and so will not be listed as a requirement for being crypto-agile. However, even a partial list may be of value to an organization. If an organization does wish to have such an inventory, the following are recommended inclusions:

- Client metadata (e.g., machine name, IP address, network location, etc.);
- For each SSH key on this machine:
    - Directory location of SSH key;
    - Metadata of user associated with SSH key;
    - Public key;
    - Public-key metadata (e.g., algorithm, length).
- For SFTP, it may also be useful to list:
    - Landing zone location for files on the client.

It would additionally be useful to list the servers which will grant resource access to this client public key. However, this is usually not inherently available. It may be more feasible to cross-reference the client and server inventories to glean information such as this.

The traditional process of discovery involves scanning common locations in client and servers. It is theoretically possible to scan the network looking for SSH traffic, although this is more complicated and, for many organizations, infeasible.

I.2.12.3 Migration Considerations

When migrating to new algorithms, the main considerations are:

- It is assumed that the vendors or an appropriate industry working group will have made the appropriate standards modifications to the protocol, if necessary, to support the new algorithms.
- Clients and servers typically have a many-to-many relationship, where a client is often used to connect to multiple servers, and a server generally accepts connections from multiple clients. Moreover, a server in one SSH connection may be the client in another SSH connection.
- As a result of this interconnectedness, it is essential that, at least server-side, connections using old and new algorithms be allowed simultaneously so as not to disrupt business operations.
- The appropriate algorithm(s) which are to be supported must be chosen. The server may support multiple host key algorithms and provides a preferential order. The choice of which client authentication algorithms to accept is made server-side.
- Both the server and the client would need to be migrated to support the new types of cryptographic keys and algorithms. SSH is often performed through a vendor product and so the vendor would be responsible for the new product.
- Where the model used is the *Centralized Management* or *Centralized Operations* model, the central entity will need to be migrated for use in the new algorithms. In these situations, the onus on algorithm selection transfers from the servers to the central entity. The central entity will likewise need to be capable of creating the new SSH keys and, in the case of *Centralized Operations*, facilitating the new SSH connections.
- For SFTP, encryption of files is typically not persistent, so retention of old keys should not be an issue.

I.2.12.4 Cutover Strategy

The cutover strategy would depend on the maturity level, but in general, it should be server-focused. It is important to ensure that the servers are migrated first and can support both new and old connections. Upon upgrade of coding, this may simply involve re-instantiating the daemon which is accepting connections. A daemon in this context refers to code (i.e., software) that runs as a background process.

Once the servers have been migrated, the clients will be able to migrate at their own convenience. The organization would have to decide for how low long to allow backward compatibility with the old algorithm. There would presumably be a cutover date at which point the old algorithms would no longer be accepted by servers. It would additionally have to have a strategy to deal with clients whom cannot or will not migrate to the new algorithms.

When in the Centralized Management or Centralized Operations model, the central entity will obviously need to be upgrade and be simultaneously compatible with old and new algorithms. It can then be used to enforce the migration operations between clients and servers according to their readiness.

For SFTP, it would be important to track any active file transfers which are in effect at the time of migration. If they are not halted or paused, then extra care should be taken to make sure they are not interrupted.

I.2.12.5 Governance

Governance, again, can only occur when the organization at least is doing Centralized Tracking. The centralized entity could either manually or automatically track the migration status of servers at a minimum.

The governance process would need to track the migration status of tracked clients and servers. It would also need to discover and deal with any "wild west" connections which may pop up. Finally, it will need to deal with SSH clients (and possibly servers) whose connections have gone dormant for long periods of time.

I.2.13 SAML or Other Federated Identity

I.2.13.1 Description

Security Assertion Markup Language (SAML) is a standardized set of protocol messages based on XML syntax which enables authorization of a user to access a particular service. It inherently verifies identity, authenticates users, and authorizes services. Depending on the version of SAML, some inherent elements may be encrypted.

Cryptographic security in SAML is most often provided by running the service over TLS channels. Important considerations for TLS are covered in Sections I.2.4 and I.2.5.

SAML relies on three major parties:

- User – the entity requesting access to a service;
- Identity Provider (IdP) – the entity which verifies the identity of the user;
- Service Provider (SP) – the entity providing the service.

The main asset is a SAML token provided by the Identity Provider which is verified by the Service Provider to allow the User to access the SP's resource.

Please note that the method of verification to authenticate the User to the IdP is beyond the scope of this Annex.

An alternative SAML access flow can occur when the User performs an initial login to the IdP to get a SAML token and then is free to use the token with any SP. An example would be a User logging into their computer at the beginning of a workday and then accessing services throughout the day via Single Sign-On (SSO).

SAML also has several different methods of flow. They are:

- **Bearer** – the presence of any valid SAML token will grant access to the resource;
- **Holder of the Key** – similar to Bearer, but the SAML token is bound to the User and the User must verify to the SP that they are the entity identified in the SAML token;
- **Sender Vouches** – similar to Bearer, but there is an additional entity called an Intermediary which handles all processing on behalf of the User and additionally signs messages to the SP.

Other frameworks used to provide identity authentication, such as OpenID Connect, a commonly used extension of the OAuth 2.0 authorization standard, have certain differences but follow a similar theme. Therefore, many of the considerations for these frameworks would be similar.

I.2.13.2 Discovery/Inventory

Any inventory of SAML should start with a list of each different instance of a SAML network, usually with a unique IdP. For each IdP, the following information should be inventoried:

- SAML version used;
- IdP name and metadata (e.g. machines on which IdP resides);
- IdP vendor name and version or internal identifier if developed in-house;
- List of downstream SPs accepting tokens from this IdP;
- General list of users (actual list or generic information on types of system is too dynamic to keep track); this would include information as to what type of SAML is used (e.g. bearer, holder-of-the-key, sender-vouches).

I.2.13.3 Migration Considerations

While the SAML specification is maintained by OASIS, the cryptography available in SAML is inherited from XML, which is maintained by W3C. To migrate SAML for use with new algorithms, the expected path would be for an update to the XML encryption standard by W3C to include these algorithms.

From there, the following would be the major considerations:

- The vendor or in-house development team would have to update the coding of the IdP, SP, and user to accommodate the new SAML standard:
    - IdPs and SPs perform processing and would need coding changes in all cases;

  - Clients who perform holder-of-the-key would also need coding changes;
  - Clients who only do bearer or sender-vouches would need to ensure ancillary changes such as buffer sizes and storage are compatible with new data types and sizes.
- The appropriate algorithms would need to be selected.
- It would be vital for the product vendor or creator to give guidance on how their product changes would affect users, IdPs, and SPs whom have not migrated yet. This is critical for backwards compatibility.

I.2.13.4 Cutover Strategy

The cutover strategy is very dependent upon the considerations described in Section I.2.13.3. Migrated products for which backward compatibility is fully supported would require a different strategy from those which do not or have some issues with it. In any case, there is no way of knowing the effects of a migrated protocol at this point in time. Thus, we can state only some generic principles in terms of cutover strategy.

To cut over:

- There would first have to be an understanding of the new protocol, the way different products work, and the results of product testing.
- Each entity would have to generate new keys offline and have them distributed to the other entities, again offline
- When putting the new keys (and hence the migrated product) into service, there must be a strategy in place to turn on new capabilities in a certain order, observe behavior, and deal with entities which have been rendered inoperable or degraded.
- There must also be a strategy in place to deal with SPs which were previously untracked and unaccounted for and will be degraded or rendered inoperable.
- Beyond this, we cannot say much about a cutover strategy at this point.

I.2.13.5 Governance

More than any other use case, this one would appear to require the most inherent governance during cutover. It should consider not only all of the different entities which have been tracked, but also those that will be discovered during the migration. It will likely have to deal with entities becoming degraded or inoperable during cutover.

ANNEX J: MOCK MIGRATION TO PQC - EXERCISE NOTES

J.1 Introduction and Exercise Description

This Annex contains a detailed summary of insights from a "mock migration" to Post-Quantum Cryptography (PQC) exercise conducted with a Public Key Infrastructure (PKI) technology provider (viz., Entrust) that systematically developed new insights into "who will need to do what" to migrate quantum-vulnerable cryptography, in a number of use cases, to use standardized quantum-safe cryptography in the future.[22]

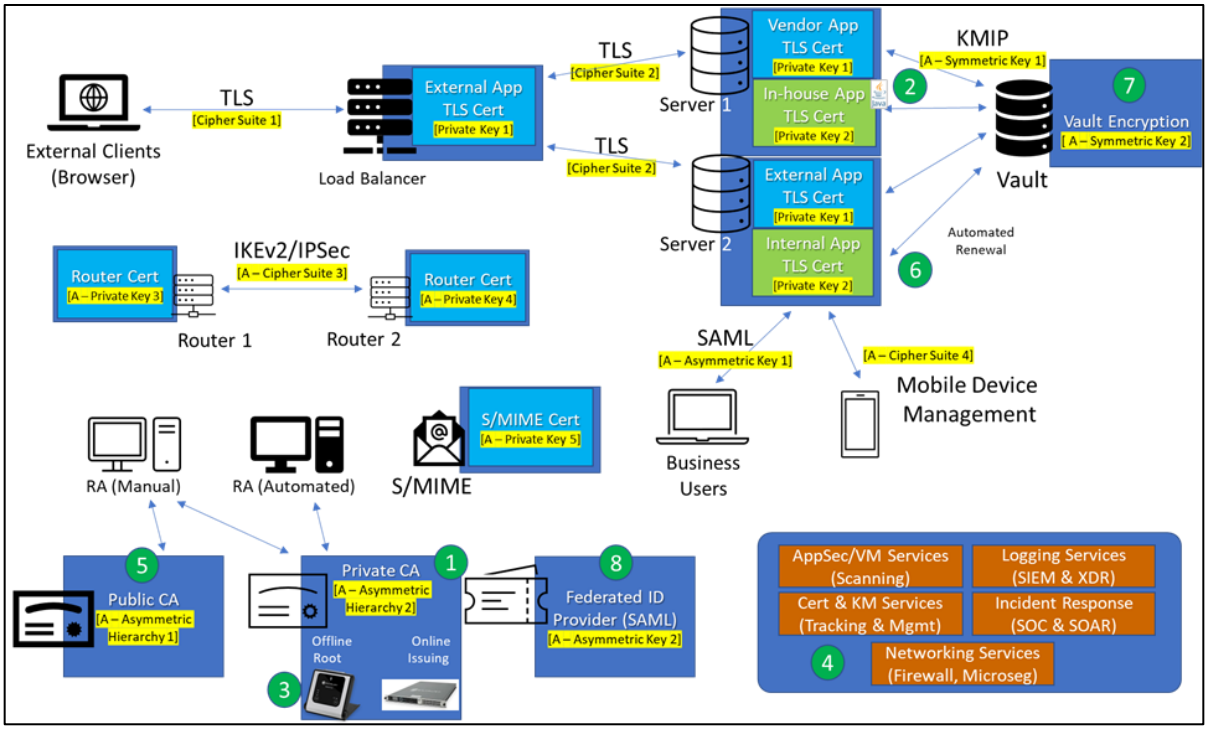
J.1.1 Purpose

This Annex was to developed to provide some insight into the inherent complexity and effort needed to work through a cryptographic migration (on paper) for a specific Information Technology (IT) system in which most of the technology is supplied by more than one vendor. Whereas Annex I: Cryptographic Agility Exercise Notes focuses on introducing the reader to the practical considerations of a (not necessarily quantum-related) migration, this Annex expands on high-level activities, roles, and responsibilities that technology providers, their customers, and additional contributors will need to perform during a migration to PQC. The belief is that thinking through the exact steps of a migration using a systematic approach, such as the one described in this Annex, will enable system owners and operators to determine many different potential challenges that would have been previously unknown to them and inform their planning for an actual migration of practical use cases to use standardized PQC.

For reference, the insights documented in this Annex emerged during the course of sixteen working sessions held over a period of six months in 2024. Each meeting focussed on identifying the steps needed to migrate one of the sixteen different use cases or sub-use cases described in Section J.2 of this Annex.

For the purpose of this exercise, we collaborated with Entrust, which currently has many technology offerings involving different realizations of cryptography, as well as a PQC roadmap for the products and solutions that it offers. We began by diagraming a representative example IT system that an organization might be using today leveraging Entrust's technology and products, though not to the exclusion of other technology providers, to interact electronically with its customers, suppliers, and/or other parties (internal and external to the organization). This system is illustrated in Figure J-1 on the next page.

[22] The approach described in this Annex was developed and refined during meetings between members of the CFDIR Quantum-Readiness Working Group (QRWG) and R&D stakeholders at Entrust from December 2023 to May 2024.



Legend: Current and Planned PQC Capabilities :

1. PKI-as-a-Service
2. PKI Toolkit for Java
3. PQC for HSMs
4. PQC Related Professional Services
5. PQ Certificate Services (Public Trust SSL)
6. PQ Certificate Agents
7. PQ Key Management Products
8. ID-as-a-Service / PQ SSO

Notes:

1. Cryptography locations are highlighted in yellow, and will need to migrate to PQC.
2. This is the "before" migration picture. An equivalent "after" should be developed for reference.
3. The Public Certification Authority (CA), Private CA, and Federated Identity (ID) Provider (SAML) are enterprise services used by other systems.
4. External Clients and applications use the Public CA, and internal applications use the Private CA.
5. The Federated ID Provider provides access for the business users to internal apps.
6. Security services that may be impacted are shown on the bottom right of Figure J-1.

Figure J-1. Arbitrary IT system, prior to being migrated to become "Quantum-Safe".

While the diagram highlights the most prevalent components of the system, there will also be inline network devices such as firewalls and proxy servers through which the transmitted data will pass. A change in cryptographic algorithm could potentially cause issues with these devices due to larger buffer sizes, differences in format, unrecognized newly-defined configurations, or other factors. While the considerations for these devices will not be dealt with in this Annex, it is worth noting that these components may play a role in the migration and would need to be adequately tested.

We then worked through the sequence of activities which would essentially have to occur in order to migrate quantum-vulnerable cryptography to PQC for every critical component in the system. This included the cryptography necessary to protect the confidentiality and integrity of data and communications as well as to authenticate users, a major use of cryptography that spans many of the use cases examined. As opposed to Annex I, quantum safety is the primary focus as we decided to drill down to a more granular level involving individual standards and protocols.

We note here that, for various technical or non-technical reasons, it may not always be possible to migrate a system to be quantum-safe. Handling such situations is a topic on its own and deserving of its own analysis. For this purpose of this Annex, such considerations will be considered out of scope.

J.1.2 Structure of this Annex

Section J.2 of this Annex describes thirteen different use cases that, as much as possible, correspond to those described in Annex I. This being said, there are a few differences in Figure J-1 (compared to Figure I-1 in Annex I) that led to the expansion of two use cases to include "sub-use case" variants. This Annex also introduces two additional use cases (viz., *FIDO2 and Other User Authentication Methods*; *Mobile Device Management / MDM*), and a new glossary in Section J.3.

Each of the use cases are supported by different components in the example IT system diagrammed in Figure J-1. Please note these use cases are by no means an exhaustive list. It is envisioned that new use cases may be explored and added to future revisions of this Annex.

Each use case documented in Section J.2 of this Annex contains six subsections as follow:

1. **Description**: A general description of the use case with details material to this analysis.
2. **Assumptions**: A set of assumptions with regards to the use case. This can include assumptions on existing systems or on future activities pertaining to this use case.
3. **Migration Activities**: The various high-level activities which various actors would perform in the course of the migration. These activities will occur in six different migration stages as described on the next page.

4. **Backward-Compatibility Considerations**: Considerations involved in the migration in terms of ensuring backward compatibility of older algorithms and technology.
5. **Potential Downgrade Attacks**: The potential for an attacker to downgrade the use case so that classical quantum-vulnerable cryptography could be in service rather than quantum-safe cryptography.
6. **Summary of Key Issues or Things You May Not Have Initially Thought Of**: This highlights the key points discovered from the exercise which should be top of mind when planning a practical migration of this use case.

One of the findings of this exercise is there may be at least ***six stages in a migration to PQC***:

1. **Standards and Direction**: Standards or authoritative bodies need to set direction in terms of algorithms and protocols.
2. **Planning and Decisioning:** Technology providers and their customers (called "consumers" throughout this Annex) will then need to decide which cryptographic algorithms they will support and conduct high-level planning.
3. **Technology Development:** Technology providers will then develop new versions of their products and/or services to incorporate PQC functionality as well as ancillary functions.
4. **Technology Installation**: Consumers will then install and test the new versions of technology products.
5. **Operational Migration**: Consumers will then perform the necessary operational activities to enable PQC in their systems using new technology and to update user or client devices as appropriate.
6. **Post-Migration Activities**: Consumers will then monitor and validate the new PQC implementations as well as work toward deprecation of older algorithms and technology.

These stages are anticipated to be sequential, although there may be overlap where migration actions in one stage may start before the previous one has finished. These stages have been incorporated into the description of **Migration Activities** in subsection J.2.**n**.3 of each use case in this Annex, where '**n**' is the use-case number.

In addition to the above, visual representations of the steps involved during the migration activities are provided for all of the use cases. They are based on the ***Quantum Migration Conceptual Model*** shown in Figure J-2 on the next page.

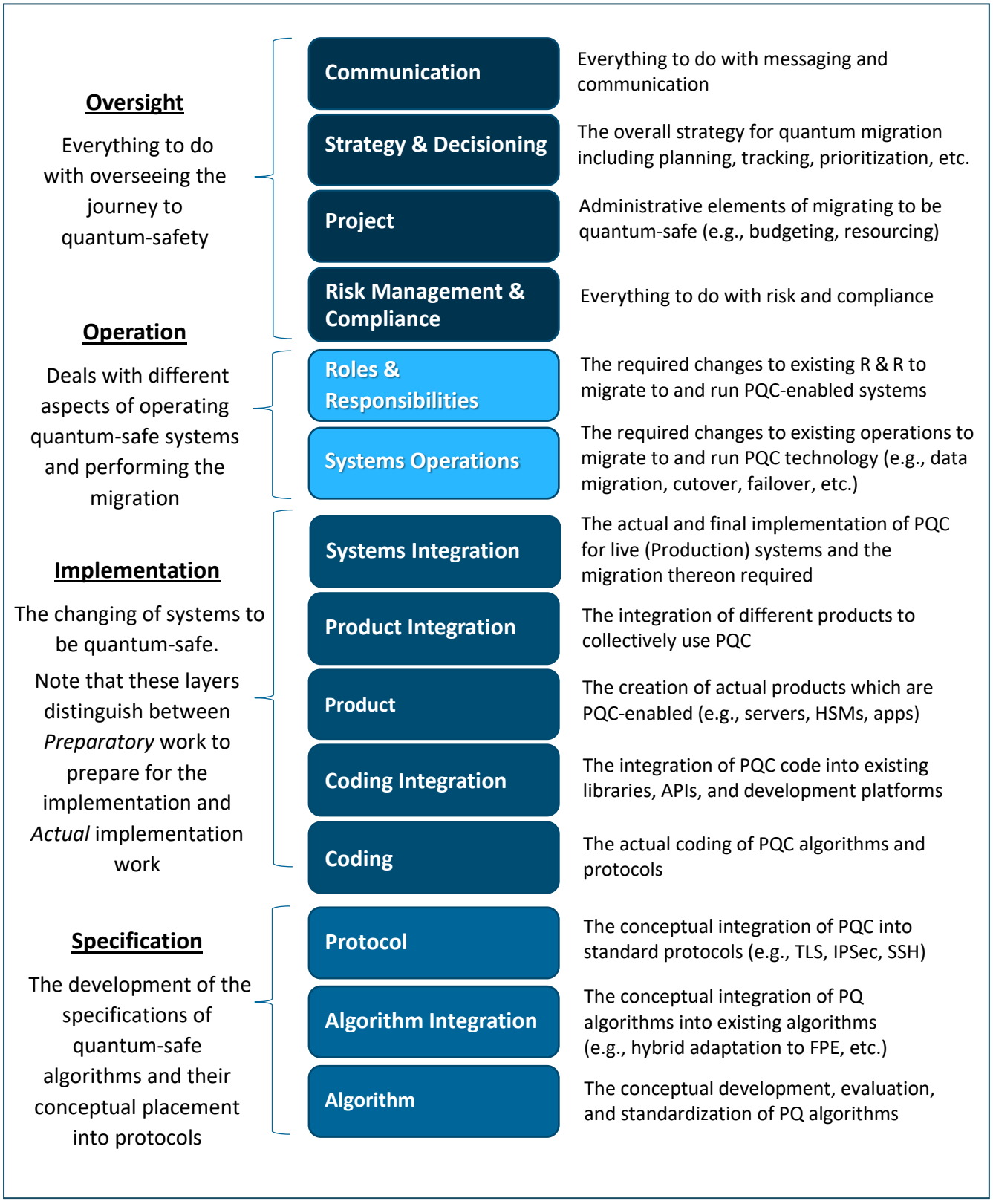
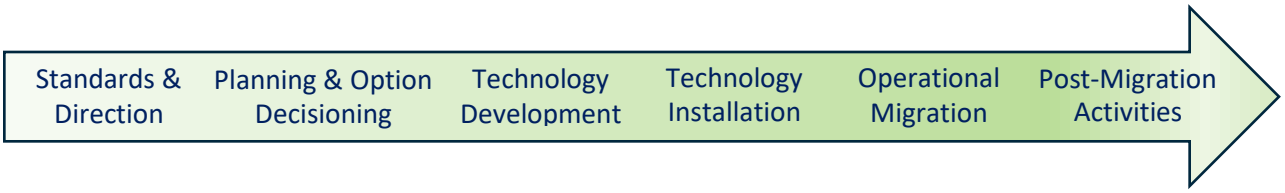


Figure J-2. Quantum Migration Conceptual Model

The model on the previous page allows us to visually describe the single level, or multiple levels, on which a particular action or event will occur. It was introduced during a *Global Financial Services PQC Congress* call organized by the FS-ISAC in 2022, to coordinate efforts across different quantum working groups, and was subsequently adopted by the CFDIR QRWG.



The visual representation of the migration model also facilitates an understanding of when each action in a migration will occur. By mapping each migration action to one (or more) of the different levels in the conceptual model introduced in Figure J-2, and in line with one of the six migration phases illustrated on the timeline above, an informative two-dimensional visualization of 'who will need to do what and when' can be developed for each quantum migration use case.

There are three points which are ever-present and apply to all use cases. They are presented here for reference so that the notes for each use case can focus on their own relevant content:

- The NIST standards work to approve post-quantum algorithms[23] is a precursor to all other standards work and is inherently present at the Algorithm level in every use case. All other standards work essentially build off of this work by NIST.
- Backward compatibility mainly focuses on the ability of devices and entities which have not been upgraded to continue to function as other devices and entities around them are migrated to PQC. For each use case, there will presumably a cutoff date whereby all instances of the old, quantum-vulnerable technology will no longer operate.
- There is a possibility and even probability that systems may be in a state of flux post-migration. This is due to the many unknown challenges which have yet to be encountered. It is important to keep abreast of cryptographic best practices during this time.

Two of the main migration actors are:

- Technology Providers (e.g., PKI and other technology vendors (e.g. HSMs, tokens) and/or suppliers); and
- Consumers (i.e., system owners and operators, including various different groups within their organizations, who use or rely on PKI technology).

In addition, depending on the use case, there may also be numerous other actors which have roles to play, including but not limited to:

[23] NIST Post-Quantum Cryptography Timeline | CSRC (nist.gov)

- Standards organizations and other entities that give direction (e.g., NIST, PKI Consortium, IETF, IANA, CA/Browser Forum, OASIS, W3C, FIDO Alliance);
- Additional Technology Providers (e.g., vendors of complementary technologies); and
- Use-case-specific teams or roles within an organization.

As with Annex I, these use cases are intended as a useful reference for the reader's preparation efforts with respect to quantum migration. In practice, the actors listed above will follow applicable industry guidance and may not be in position to take individual action based on the content of the use cases.

J.2 PQC MIGRATION USE CASES AND Findings

This Section contains the migration notes for the following thirteen use cases:

J.2.1: Use Case 1: Establishment of a Public Certification Authority (CA);

J.2.2: Use Case 2: Establishment of a Private Certification Authority (CA);

J.2.3: Use Case 3: End-Entity Certificate Migration;

J.2.4: Use Case 4: TLS Connections to General External Client Browsers;

J.2.5: Use Case 5: Non Browser-based TLS Connections;

J.2.6: Use Case 6: Internally Developed Applications;

J.2.7: Use Case 7: Code Signing (Private);

J.2.8: Use Case 8: Vault Encryption;

J.2.9: Use Case 9: S/MIME Secure E-mail;

J.2.10: Use Case 10: SAML or Other Federated Identity;

J.2.11: Use Case 11: IPsec and IKE;

J.2.12: Use Case 12: FIDO2 and Other User Authentication Methods;

J.2.13: Use Case 13: Mobile Device Management (MDM).

J.2.1 Use Case 1: Establishment of a Public Certification Authority (CA)

J.2.1.1 Description

This use case will cover the establishment of a public Certification Authority (CA) from the perspective of a subscriber. In particular, this CA will be designated by the organization as being allowed to issue certificates on domain names belonging to that organization. The actions of an individual entity during the certificate lifecycle are handled as part of a different use case (viz., End-Entity Certificate Migration as described in Section J.2.3).

J.2.1.2 Assumptions

1) It is believed that the Internet Engineering Task Force (IETF) will not need to change the PKIX/X.509 standard if a single algorithm is applied to the certificate signature. In some cases, including those involving hybrid cryptography, new additions to the X.509 Certificate will be needed (e.g., hybrid algorithms, extensions, ObjectID, and PublicKey types).
2) The Technology Provider's (viz., Entrust's) current process for requesting certificates will not change. There may only be some slight changes to the portal to reflect the use of new CAs and new algorithms.
3) It is expected that mechanisms for Certificate Transparency (CT) will remain largely the same, however the larger bandwidth requirements of Post-Quantum (PQ) signatures may prompt some redesign of the CT infrastructure. This work will be resolved between the public CA and browser vendors.
4) Mechanisms for certificate revocation checking (e.g., Certificate Revocation Lists (CRLs), Online Certificate Status Protocol (OCSP)) will remain largely the same with negligible changes.

J.2.1.3 Migration Activities

The migration to PQC for this use case can be modelled through the diagram on the next page. The corresponding individual migration steps would be as follows:

1) Standards and Direction:

Internet Engineering Task Force (IETF):

- The IETF's Limited Additional Mechanisms for PKIX and SMIME (lamps) working group [24] will need to update PKIX certificate standards to include new OIDs, to be defined by NIST, for each PQC algorithm standardized by NIST (e.g., FIPS 204 ML-DSA [25], FIPS 205 SLH-DSA [26]).

National Institute of Standards and Technology (NIST):

- Define new Object Identifiers (OIDs) for the new PQC algorithms as well as the binary encodings of public keys and signatures.

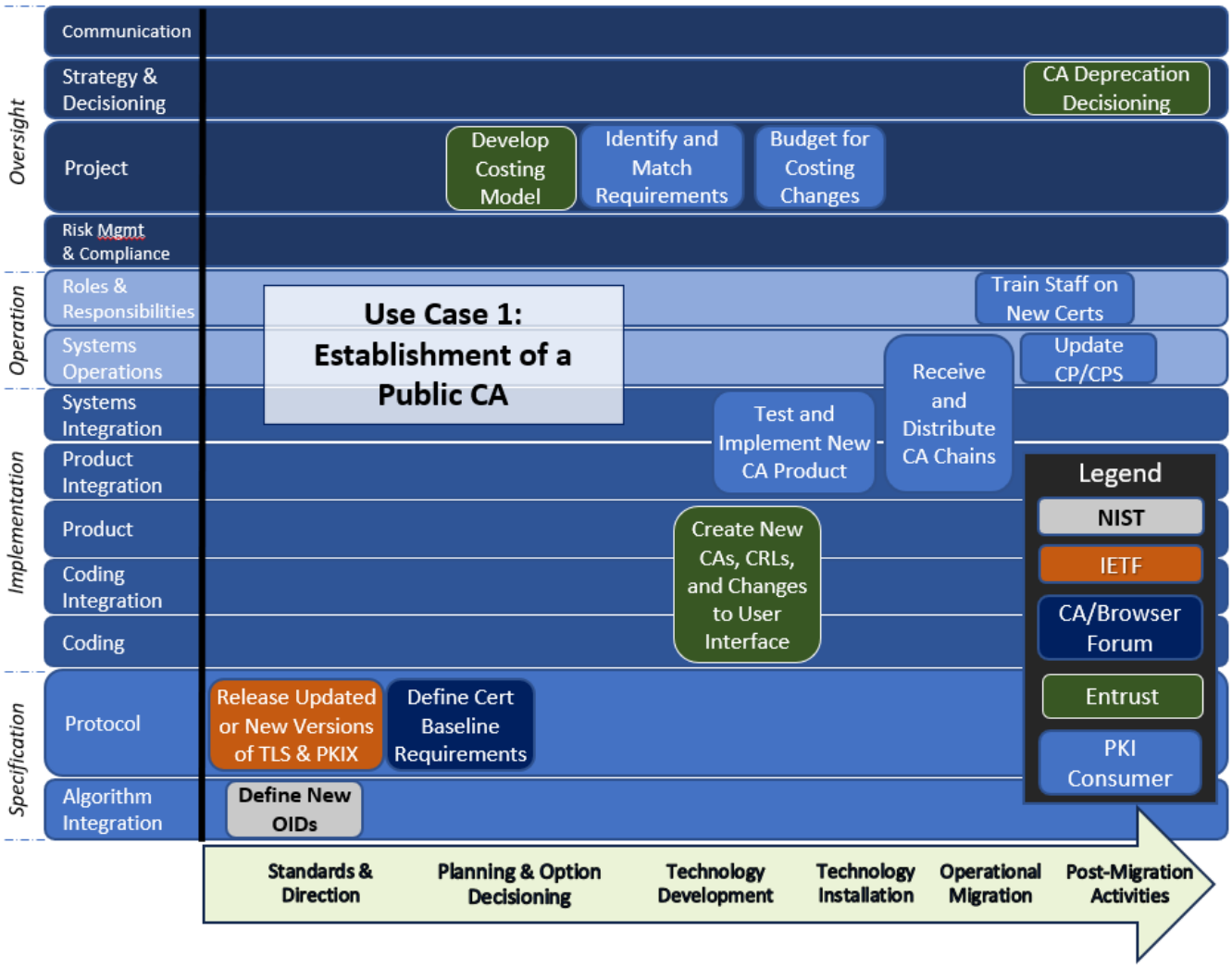
[24] Limited Additional Mechanisms for PKIX and SMIME (lamps) (ietf.org)

[25] Internet X.509 Public Key Infrastructure: Algorithm Identifiers for ML-DSA, draft-ietf-lamps-dilithium-certificates-04, July 2024

[26] Use of the SLH-DSA Signature Algorithm in the Cryptographic Message Syntax (CMS), draft-ietf-lamps-cms-sphincs-plus-07, July 2024

Certificate Authority Browser Forum (CA/Browser Forum):

- The CA/Browser Forum will need to define new policies (i.e., baseline requirements) for these certificates, such as which algorithms are to be used for which use case, the type of hybrid or PQC-only certificates which are to be made available, etc.



2) Planning and Decisioning:

Technology Provider (viz., Entrust):

- Define types of certificates and PQC algorithms available based on baseline requirements from CA/Browser Forum.
- Determine a costing model for PKI Consumers given that both classical and PQC certificates will be required for the same Fully Qualified Domain Name (FQDN) at the same time during the migration to PQC.

PKI Consumer (viz., Entrust Customer):

- Interact with Entrust to ensure their CA chains, use cases, and products meet the requirements of the Consumer's system (e.g., latency, throughput, and size and bandwidth required for using new PQC certificates and signatures).
- Ask Entrust for guidance on pricing and costing (for planning purposes) because the organization will need to have more certificates in use (as compared to today) during the migration to PQC.
- Include appropriate costing changes (if any) into the budget.

3) Technology Development:

Technology Provider (viz., Entrust):

- Establish new global public root certificates which are quantum-safe;
- Work with browser and operating system (OS) vendors to have the new public roots accepted into root stores;
- Establish new global public issuing certificates that are quantum-safe;
- Depending on events, establish a cross-signed issuing certificate corresponding to the old root certificate;
- Establish new CRLs and OCSP responders which will leverage quantum-safe signatures;
- Update Registration Authority (RA) capabilities to leverage the PQC algorithms; and
- Make appropriate changes to tooling (e.g., portal, approval procedure, etc.) to enable requesting of certificates from new CAs. Note that upon initial launch, the RA should still be able to request certificates from the old, classical CA.

4) Technology Installation:

Consumer (viz., Entrust Customer):

- Understand and test the changes to the requisite product including backward compatibility of certificate requests, CRLs, and other certificate management activities.

5) Operational Migration:

Consumer (viz., Entrust Customer):

- Receive and distribute new CA chains from Entrust;
- Request new RA certificate from Entrust;
- Train both Certificate Administrators and technology staff on the changes involved in using PQC certificates; and
- Update the organization's Certificate Policy / Certificate Practice Statements (CP/CPS) to reflect the new public CA.

OS/Browser Community:

- Validate the new CA certificates leveraging their internal processes or WebTrust audits as appropriate. Note that this will be time-based and may not occur before deployment of Root CAs.[27]

6) Post-Migration Activities:

Technology Provider (Entrust):

- Deprecate the old CA in conjunction with guidance from bodies such as NIST and CA/Browser Forum at an appropriate time.

J.2.1.4 Backward-Compatibility Considerations

This use case only establishes the capabilities of a new public CA. The actual issuance and use are covered in Section J.2.4 (Use case 4: TLS Connections to General External Browsers). As the original public CA will be accessible, there are no backward-compatibility concerns.

J.2.1.5 Potential Downgrade Attacks

This use case only establishes the capabilities of a new public CA. The actual issuance and use are covered in Section J.2.4. The only consideration is if an attacker were to convince the certificate requestor to accidentally request from the old public CA. This would be particularly pertinent where the two CAs share some common infrastructure (e.g., RA, portal, etc.). There would likely be other compensating controls in place to prevent this from happening.

J.2.1.6 Summary of Key Issues or Things You May Not Have Initially Thought Of

1) One of the main assumptions here is that TLS 1.3 and later versions will allow a certificate to be chosen in the same way that a cipher suite may be chosen. In particular, the client can tell the server what type of certificates (e.g., classical, hybrid, PQC-only) the client can accept. For this use case, it is assumed that the CA will issue two certificates (one classical and one involving PQC) for each use instead of one certificate.
2) The fact that there will be two certificates for each use introduces the issue of costing as described under Planning and Decisioning (i.e., migration step #2) in Section J.2.1.3 above. This will need to be considered as part of the business model.

[27] CA-Browser Forum Baseline Requirements for TLS Browser Certificates - Section 8 of TLS BR v2.0.5 on Compliance Audit and Other Assessments, https://cabforum.org/working-groups/server/baseline-requirements/documents/

J.2.2 Use Case 2: Establishment of a Private Certification Authority (CA)

J.2.2.1 Description

This use case will cover the creation of a Private Certification Authority (CA). Unlike the establishment of a Public CA described in Section J.2.1 (Use Case 1: Establishment of a Public CA) this is completely under the control of the organization that operates the system diagrammed in Figure J-1, with the exception of technological limitations and best practices.

There are two main paths to migration:

1. Creating new quantum-safe CAs leveraging existing Hardware Security Modules (HSMs) which offer classical protection through internal security measures.
2. Leveraging new quantum-safe HSMs to commission the quantum-safe CAs.

While Entrust provides the nShield Edge and nShield Connect HSMs for offline and online roots respectively, the migration will follow the same sequence and have the same considerations with any compatible HSM (e.g. Luna, Utimaco). Hence, the remainder of this use case will make reference to generic HSMs independent of manufacturer.

J.2.2.2 Assumptions

1. There is an existing classical private CA in place consisting of:
    - An air-gapped root CA leveraging an offline HSM attached to a dedicated device such as a laptop computer.
    - An online issuing CA leveraging an online HSM attached to a server or network.
    - One or more Registration Authorities (RAs) which leverage a classical RA certificate to perform either manual or automated request and lifecycle management.
    - One or more inspection CAs off of the root CA will exist on and be used by TLS inspection appliances.
2. The HSMs leverage PKCS #11 v3.1 for protection of keys. [28] This allows the use of stateful hash algorithms. It is anticipated that PKCS #11 v3.2 will have compatibility with other PQC algorithms.
3. Entrust's current process for requesting certificates will not change. There may only be some slight changes to the portal to reflect the use of new CAs.

J.2.2.3 Migration Activities

The migration to PQC for this use case can be modelled through the diagram on the next page. The corresponding individual migration steps would be as follows:

[28] OASIS PKCS 11 TC - OASIS (oasis-open.org)

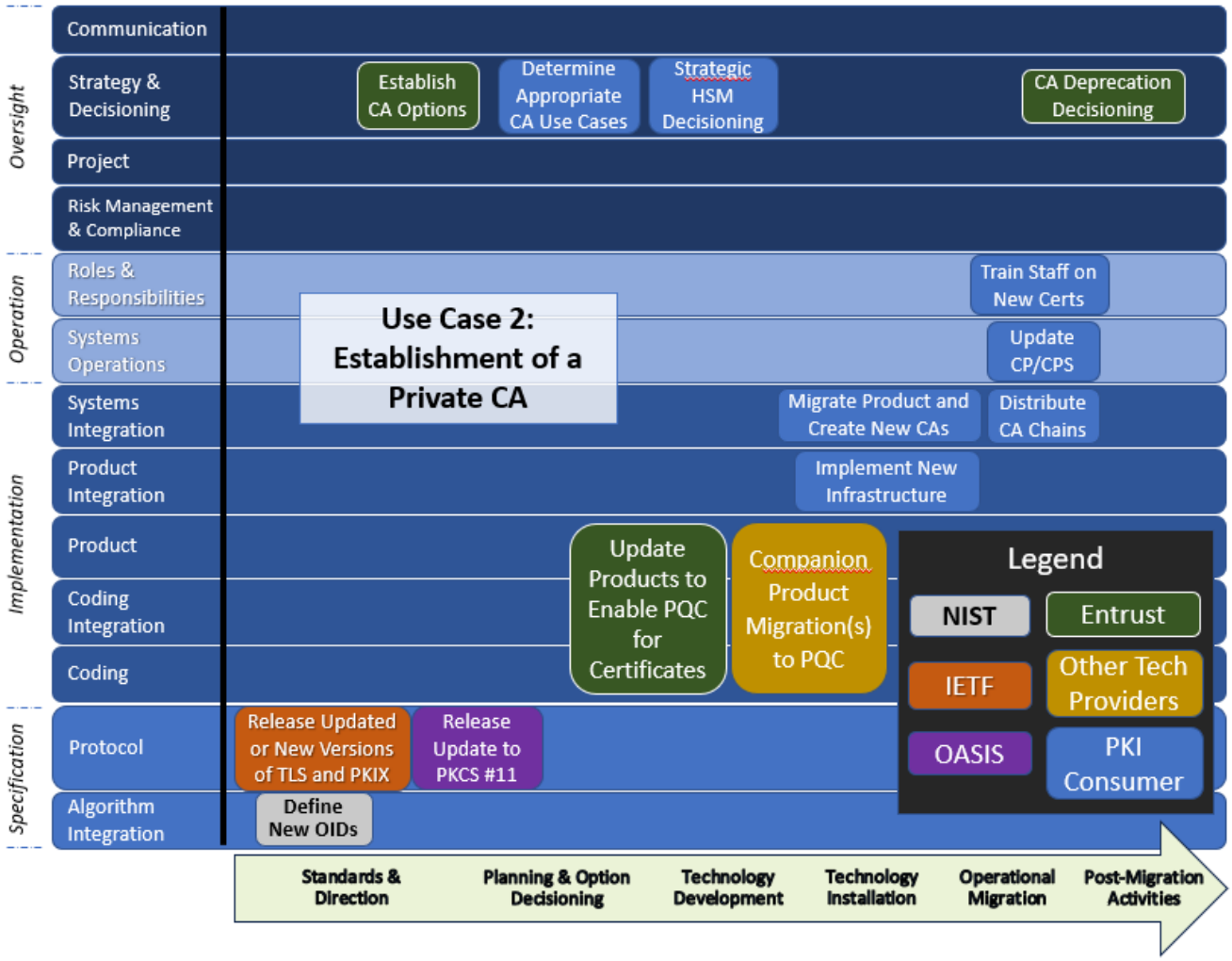
1) Standards and Direction:

National Institute of Standards and Technology (NIST):

- Define new Object Identifiers (OIDs) for the new PQC algorithms as well as the binary encodings of public keys and signatures.

Internet Engineering Task Force (IETF):

- The Limited Additional Mechanisms for PKIX and SMIME (lamps) working group [29] of the IETF will need to update PKIX certificate standards to include new OIDs, to be defined by NIST, for each PQC algorithm standardized by NIST (e.g., FIPS 204 [30], FIPS 205 [31]).



[29] Limited Additional Mechanisms for PKIX and SMIME (lamps) (ietf.org)

[30] draft-ietf-lamps-dilithium-certificates-03 - Internet X.509 Public Key Infrastructure: Algorithm Identifiers for ML-DSA

[31] Use of the SLH-DSA Signature Algorithm in the Cryptographic Message Syntax (CMS), draft-ietf-lamps-cms-sphincs-plus-07, July 2024

Organization for the Advancement of Structured Information Standards (OASIS):

- Provide an updated version of PKCS #11 (to version 3.2 at minimum) compatible with PQC algorithms.[32]

2) Planning and Decisioning:

Technology Provider (viz., Entrust):

- Establish options for PKI consumers on the following:
  - Type of certificates, certificate extensions available
  - PQC algorithms available including, where applicable, level of exhaustion and how they are used in the hierarchy (e.g., different algorithms at different levels)
  - PQC modes of operation (e.g., hybrid, PQC-only)
  - Cross-signing and/or multiple instances
  - Use cases appropriate for each type of CA (no Swiss-army knife)
  - Recommended expiration dates
  - Compatibility requirements for complementary vendor technology
  - Which options are available for customization

PKI Consumer:

- Determine appropriate use cases for which the CA will be needed and match against the options offered by Entrust.
- Make appropriate decisions with respect to certificate types, algorithms, mode of operations, hierarchy structure, instances, and expiration dates.
- Decide on one of the two paths of migration as listed in Section J.2.2.1.

3) Technology Development:

Technology Provider:

- Provide PQC firmware update for the HSMs to create CAs and perform CA operations for the different selected options.
- Provide new versions of the HSMs with new PKCS #11 PQC capabilities.
- Establish new mechanisms for RAs and CRLs, including capability for PQC RA certificates and appropriate changes to the user interface.
- Update protocol interfaces and APIs supporting management of certificates.

[32] OASIS PKCS 11 TC - OASIS (oasis-open.org)

Other Technology Providers (CA Complementary Technology):

- Code and enable the new algorithms in their CA complementary products [33] according to industry specifications and Entrust options.
- Enable appropriate protocol interfaces and APIs for PQC.

4) Technology Installation:

Consumer:

- Depending on the migration path chosen:
  - Receive the new version of the HSM firmware from Entrust and upgrade the HSMs, or
  - Upgrade to new model of the HSMs and, if required, migrate existing keys to be protected under the new HSM master keys.
- Migrate Entrust and compatible HSM technologies to compatible quantum-safe versions.
- Migrate RA to the new quantum-safe version.

5) Operational Migration:

Consumer:

- Create a new root CA, new issuing CA, and new CRLs according to selected parameters.
- Distribute root and issuing CAs to relevant devices and endpoints and other consumers.
- Train certificate administrators and technology staff on the changes involved in using PQC certificates.
- Deprecate old CA when all certificates have been migrated to new CA.
- Update the organization's Certificate Policy / Certificate Practice Statements (CP/CPS) to reflect the new private CA.

6) Post-Migration Activities:

Technology Provider (Entrust):

- Deprecate the old private CA when all certificates have been migrated to the new private CA.

[33] Examples of CA-complementary technology include devices or laptop computers used in lieu of or in conjunction with nShield Edge HSM, and servers leveraging nShield Connect HSM.

J.2.2.4 Backward-Compatibility Considerations

The following are considerations:

- The nShield Edge, nShield Connect, and compatible HSMs will be fully backward compatible with the old CA technology. It is highly probable the same will be true for most of the other supported technology elements.
- The use case only establishes the capabilities of a new Private CA. The actual issuance and use of the new CA is covered in a different use case. As the original Private CA will be accessible, there are no further backward-compatibility concerns.

J.2.2.5 Potential Downgrade Attacks

The HSMs as well as the software, drivers, and applications using them would be controlled by the organization, so there is no direct threat of downgrade in that respect. As with the Public CA, the only way of downgrading would be for an attacker to convince the requestor to accidentally request a certificate from the old CA. There would likely be other compensating controls to prevent this from happening.

J.2.2.6 Summary of Key Issues or Things You May Not Have Initially Thought Of

1) A migration can occur using existing infrastructure which can support PQC technology but may not involve PQC protection of the private keys. Alternatively, it is possible to obtain new infrastructure which is fully PQC compatible. If doing the former, a decision may need to be made to determine if a second migration would be needed from a risk perspective.
2) An important aspect to consider is the effect that migrating to PQC will have on TLS inspection capabilities if it is leveraging your Private CA. Inspection is inherently more difficult in TLS 1.3, compared with TLS 1.2, and may be harder still when TLS is updated for PQC.[34] As the details are still unknown, we will leave this out of scope but remark that this is a topic of consideration for this use case.

[34] Addressing Visibility Challenges with TLS 1.3 within the Enterprise, NIST Special Publication 1800-37A, Second Preliminary Draft, January 2024, 6 pages

J.2.3 Use Case 3: End-Entity Certificate Migration

J.2.3.1 Description

This use case will cover the migration of certificates throughout their lifecycle from an end-entity perspective. This would include Certificate Signing Request (CSR) generation and certificate loading as well as revocation and distribution, but not be tied to a protocol such as TLS.

The loading and renewal of certificates can occur either manually or using automation.

Many organizations leverage a Content Delivery Network (CDN) (e.g., Akamai, AWS, CloudFront) to deliver content externally. These often leverage certificates to assist with facilitation of services. From the perspective of this use case, they will simply be considered as an end entity requiring a certificate.

J.2.3.2 Assumptions

1) The organization's process for requesting certificates will not change. There may only be some slight changes to the portal to reflect the use of new Certification Authorities (CAs).
2) Certificate renewal can either be manual or automated. If it is automated, it will make use of an automated renewal protocol such as ACME.[35] While these protocols may already work with the PQC certificates, there may be some adjustments needed due to intricacies in the new PQC algorithms, new file formats, or changes in other protocols. We will assume that the standards need to have some updates.

J.2.3.3 Migration Activities

The migration to PQC for this use case can be modelled through the diagram on the next page.

The corresponding individual migration steps would be as follows:

1) Standards and Direction:

Internet Engineering Task Force (IETF):

- Update the PKCS #7/CMS, PKCS #10 and PKCS #12 standards for certificate file formats, requests and responses.
- Update additional standards for automated certificate renewal to support PQC algorithms (e.g., ACME, CMC, EST, CMP).[36]

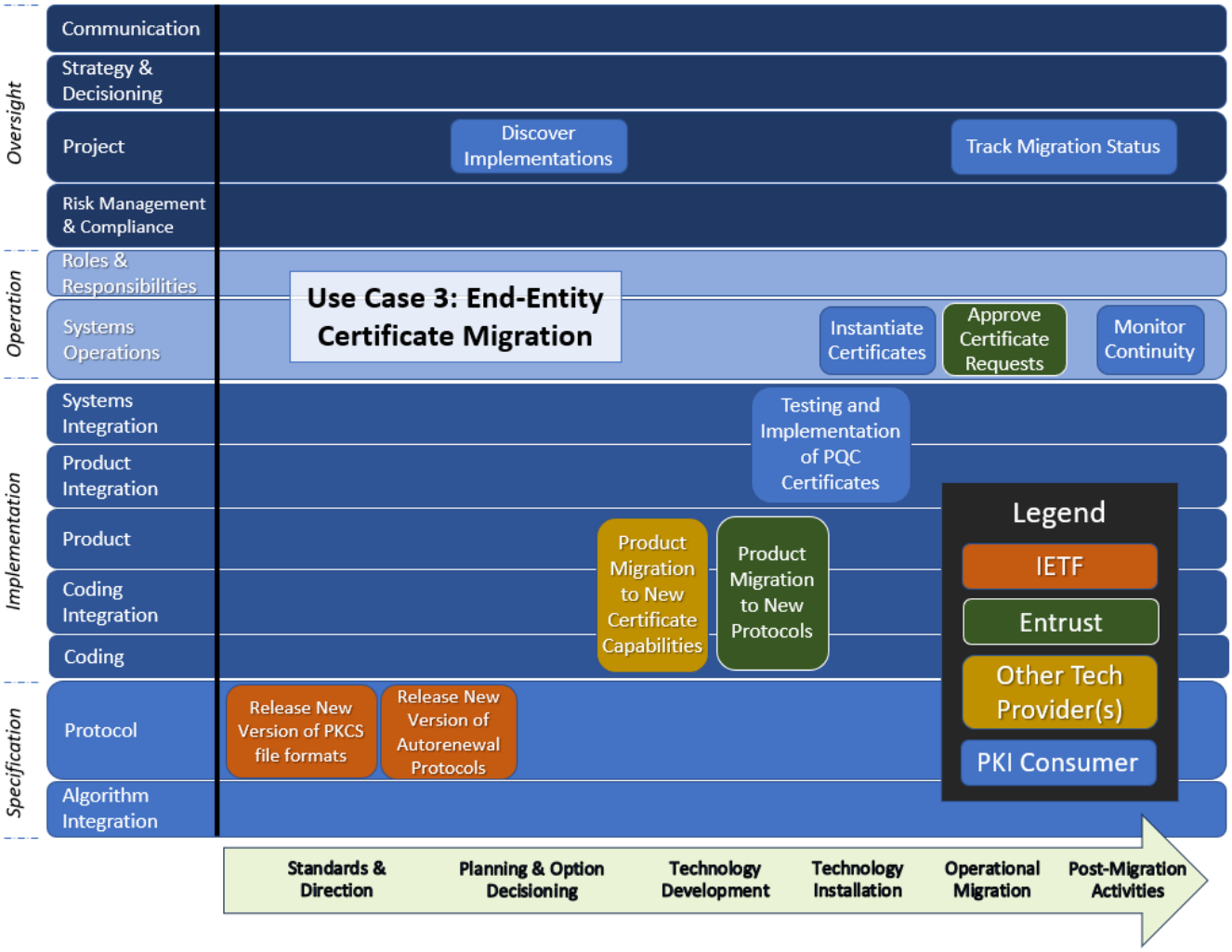
[35] Automatic Certificate Management Environment (ACME) - Wikipedia

[36] Automated Certificate Management Environment (acme) WG Documents (ietf.org)

2) Planning and Decisioning:

Consumer (viz., Entrust Customer):

- Discover and inventory each separate instance of certificate use.
  - This would include the full list of providers in the technology stack.
  - It would also include the certificates.
- Ensure systems can handle additional size of files as specified by new standards.



3) Technology Development:

Technology Provider (viz., Entrust):

- Implement products compatible with changes to new file formats and automated certificate request protocols.

Other Technology Providers:

- Code and enable the new algorithms in their products according to new certificate specifications.
- Ensure that there is support for both classical and quantum-safe CAs to work simultaneously.
- Ensure that there is a compatible PQC-enabled CSR generation tool available compatible with changes in file formats.
- Ensure that key stores for the new private keys are able to accept or import the new private key and certificate formats and store them using PQC-enabled protection mechanisms.
- Ensure the end-entity device can verify PQC-enabled CRLs.
- Ensure devices can properly implement new versions of the automated renewal mechanisms algorithms (e.g., ACME, CMC, EST, CMP).

4) Technology Installation:

Consumer:

- Implement the latest version of products from technology providers and Entrust which enable new file formats and protocols for automated certificate requests.

5) Operational Migration:

Technology Provider (viz., Entrust):

- Issue new certificates after approval by RA.

Consumer:

- Create new CSRs and submit to Entrust CA using the certificate request process.
- Use Entrust RA to approve certificate requests.
- Receive certificates from Entrust.
- Test and deploy the two-certificate model in production (as specified under the Planning and Decisioning migration step in Section J.2.1.3 (use case 1). Ensure that the correct certificate is chosen.

6) Post-Migration Activities:

Consumer:

- Monitor business continuity
- Track the status of each instance through the migration.

J.2.3.4 Backward-Compatibility Considerations

The following are considerations:

- CSR generation can still occur with the original CSR tools, so there are no issues in this respect.
- The method for requesting certificates from the old CA will continue to exist.
- The backward-compatibility considerations of the system leveraging the certificates is dependent on the use case for which this certificate will be used. This is explored in more detail in many of the subsequent use cases in this Annex.
- Some systems requesting certificates may only be able to accept a single root CA certificate. While a cross-signed root may assist, special care may be needed for these types of systems.
- There may be issues which arise from the TLS protocol itself. For example, if a legacy client does not properly set the appropriate TLS ClientHello extensions and gets served the server's (default) PQ certificate, then connection failure may occur, even if the server has a legacy certificate.[37]

J.2.3.5 Potential Downgrade Attacks

Downgrade attacks would presumably exist, but the details would be dependent on the particular use of a certificate. Consumers and technology providers will need to pay special attention to allowing clients with legacy certificates to be automatically moved to a PQ certificate upon renewal.

J.2.3.6 Summary of Key Issues or Things You May Not Have Initially Thought Of

1) Changes in file formats such as .pem, .pfx, and .crt files need to be taken into account when handling PQC certificates, as well as ancillary tools that help with managing these files (viewers, linters, installers, etc.).
2) The assumption that there are two certificates for an end entity may have interesting side effects in this use case. For example, upon renewal, one certificate may install properly, but the other may have issues. This leads to decisioning on whether to roll back changes completely or go ahead with the certificate that was installed.
3) When more than one certificate is installed, the choice of which certificate to use in the communication is a non-trivial decision. This migration will put a lot of emphasis on a part of the TLS 1.3 handshake (the signature algorithm negotiation) that may never have really been properly exercised in a world where everything recognizes RSA.

[37] draft-ietf-tls-hybrid-design-10 - Hybrid key exchange in TLS 1.3

J.2.4 Use Case 4: TLS Connections to General External Client Browsers

J.2.4.1 Description

This use case will cover the use of certificates in Transport Layer Security (TLS) connections to general external client browsers. A migration would consist of two pieces which can be done separately:

- Sub-Use Case 4(a): Cipher Suite Migration for Key Establishment; and
- Sub-Use Case 4(b): Cryptographic Migration for Authentication.

The particulars of these two sub-use-cases are described after the assumptions, below.

J.2.4.2 Assumptions:

1) The Consumer has identified a particular instance of a TLS connection which they are hosting and can get the requisite details when required. What those details are will be determined in this exercise.
2) This use case need not consider lower-level protocols (e.g., IP layer) which facilitate the connection. They would just be a pass-through.
3) From the perspective of the PKI Consumer, the first consideration is the version of TLS being used. It has been decided (by the IETF community) that no new features will be added to existing standards for TLS 1.2.[38] Given that support for PQC is not defined within TLS 1.2, it will be incumbent upon the Consumer to, at minimum, upgrade to TLS 1.3 to secure public TLS connections to browsers with PQC.

J.2.4a Sub-Use Case 4a: (TLS) Cipher Suite Migration for Key Establishment

J.2.4a.1 Description of this Sub-Use Case

This sub-use case covers the establishment of TLS ciphers suites. In particular, ephemeral keys are established under perfect forward secrecy to encrypt the connection. Migrating this portion of the TLS connection to quantum-safe ephemeral keys would protect against passive Harvest-Now-Decrypt-Later (HNDL) or Steal-Now-Decrypt-Later (SNDL) attacks.[39]

J.2.4a.2 Assumptions of this Sub-Use Case

No additional assumptions beyond those listed above in Section J.2.4.2.

J.2.4a.3 Migration Activities

The migration to Post-Quantum Cryptography (PQC) for this sub-use case can be modelled through the diagram on the next page.

---

[38] "TLS 1.2 is in Feature Freeze", draft-ietf-tls-tls12-frozen-00, April 2024

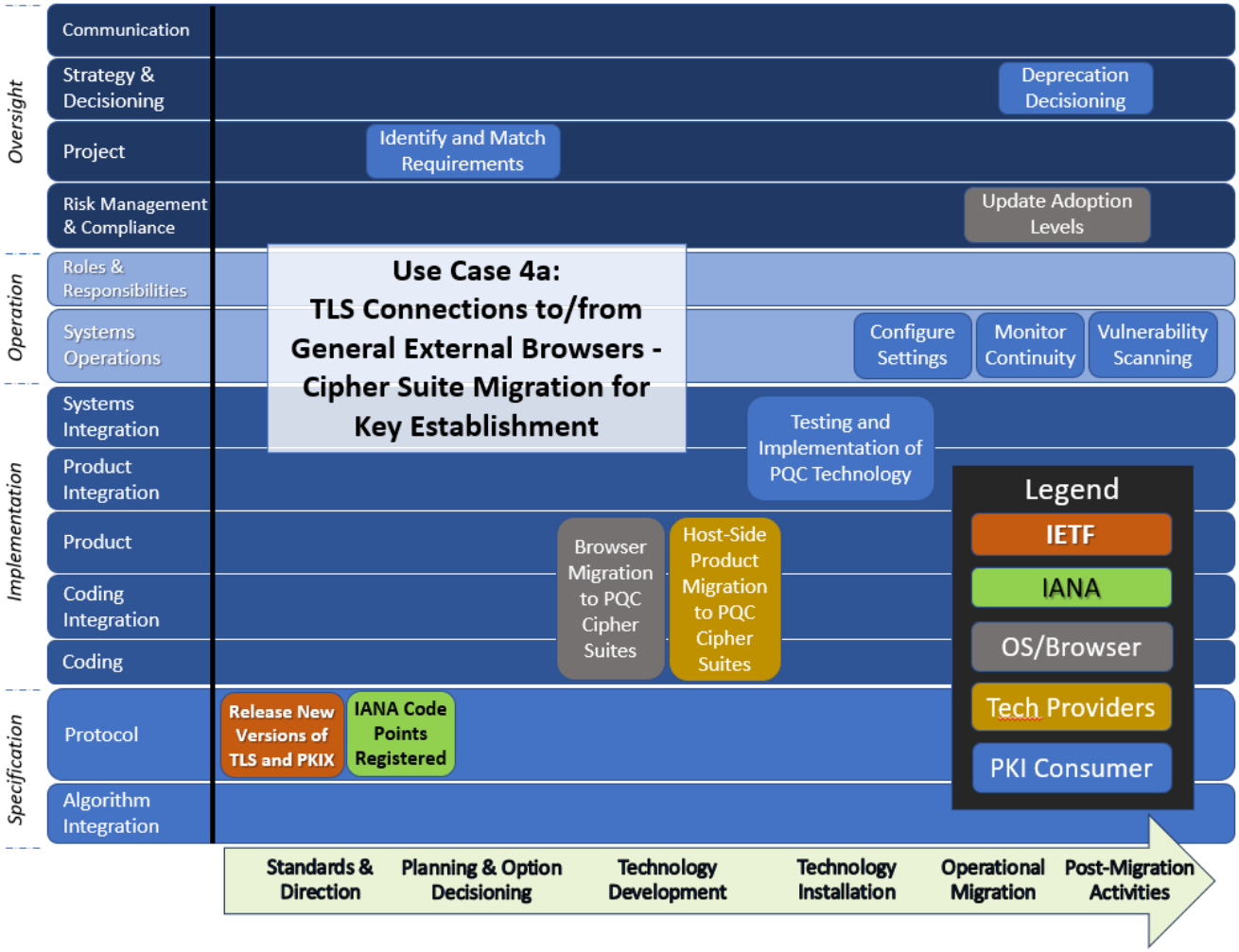
[39] "Steal-Now-Decrypt-Later" (SNDL) is a synonym for HNDL.

The corresponding individual migration steps would be as follows:

1) Standards and Direction:

IETF or Other Standards-Related Entity:

- The IETF will need to release an update to TLS 1.3 or else a new version TLS 1.4 which can incorporate new algorithms and methods to implement PQC as recommended by NIST, potentially including hybrid cryptography.[40] This will include the encoding of the public key and ciphertext.



- The IETF will also need to request that IANA register new code points, to be published within a new RFC or Internet-Draft (I-D).

Internet Assigned Numbers Authority (IANA):

- Register new code points for the new PQC algorithms.

[40] See Annex H for an *Overview of Hybrid Cryptography.*

2) Planning and Decisioning:

OS/Browser Community

- Select PQC algorithms to be supported.

Consumer:

- Interact with Technology Provider(s) to ensure their products meet the requirements of the systems (e.g., latency and throughput using new PQC algorithms).

3) Technology Development:

OS/Browser Community:

- Code and enable the new PQC algorithms in their browsers.
- Roll out the new version of compatible browsers, presumably leveraging the existing practice of browser updates.

Technology Providers:

- Code and enable the new PQC algorithms in their products according to specifications.

4) Technology Installation:

Consumer:

- Understand and test the changes to the requisite products including backward compatibility.
- Implement new technology in production systems.

5) Operational Migration:

Consumer:

- Configure settings on hosts to ensure quantum-safe cipher suites are active and/or preferred.

6) Post-Migration Activities:

OS/Browser Community:

- Provide periodic updates regarding adoption level.

Consumer:

- Monitor connections to ensure business continuity and back out if necessary.
- Test connections for quantum-safety and/or cryptographic agility.
- Perform vulnerability scanning on implementations to determine if the new cipher suites are being utilized.

- Keep abreast of OS/Browser adoption level. When a certain threshold is met, deprecate the old non-quantum-safe cipher suites in the configuration.

J.2.4a.4 Backward-Compatibility Considerations

The following are considerations:

- This use case has the characteristic that backward compatibility is automatically maintained as a result of cipher suite negotiation. New connections may negotiate quantum-safe cipher suites but can always fall back to legacy cipher suites during the migration period.
- Each cluster of hosts can be migrated independently of each other.

J.2.4a.5 Potential Downgrade Attacks

There have, historically, been many downgrade attacks in this use case such as DROWN [41] and POODLE [42]. There is definite risk that future downgrade attacks could occur for PQC cipher suites as well. This should be monitored across the industry.

J.2.4a.6 Summary of Key Issues or Things You May Not Have Initially Thought Of

1) This sub-use case can be performed independently of the other sub-use case (viz., Cryptographic Migration for Authentication) described in Section J.2.4b. The cipher suite migration will take away the Harvest-and-Decrypt (HNDL) risk against data encryption which exists today. Subsequently, the certificate migration at a later date would prevent an active quantum authentication attack.
2) Migrating cipher suites should be the easier of these two sub-use cases and would mitigate the most pressing quantum risk as the authentication attack could only occur when a quantum computer is available.

[41] Attack of the week: DROWN – A Few Thoughts on Cryptographic Engineering, March 1, 2016

[42] SSL 3.0 Protocol Vulnerability and POODLE Attack | CISA, September 30, 2016

J.2.4b Sub-Use Case 4b: (TLS) Cryptographic Migration for Authentication

J.2.4b.1 Description of this Sub-Use Case

This sub-use case will cover the implementation of quantum-safe certificates for authentication. This will prevent active attacks such as man-in-the-middle (MITM) or impersonation attacks due to quantum recovery of the static keys within certificates.

J.2.4b.2 Assumptions of this Sub-Use Case

1) The latest version of TLS will allow processing of both the classical and PQC-enabled certificate to be negotiated during the TLS handshake. All parties will work under this assumption.

J.2.4b.3 Migration Activities

The migration to PQC for this sub-use case can be modelled through the diagram on the following page.

The corresponding individual migration steps would be as follows:

1) Standards and Direction:

National Institute of Standards and Technology (NIST):

- Define new Object Identifiers (OIDs) for new PQC algorithms as well as the binary encodings of public keys and signatures.

Internet Assigned Numbers Authority (IANA):

- Register new code points for the new PQC algorithms.

IETF or Other Standards-Related Entities:

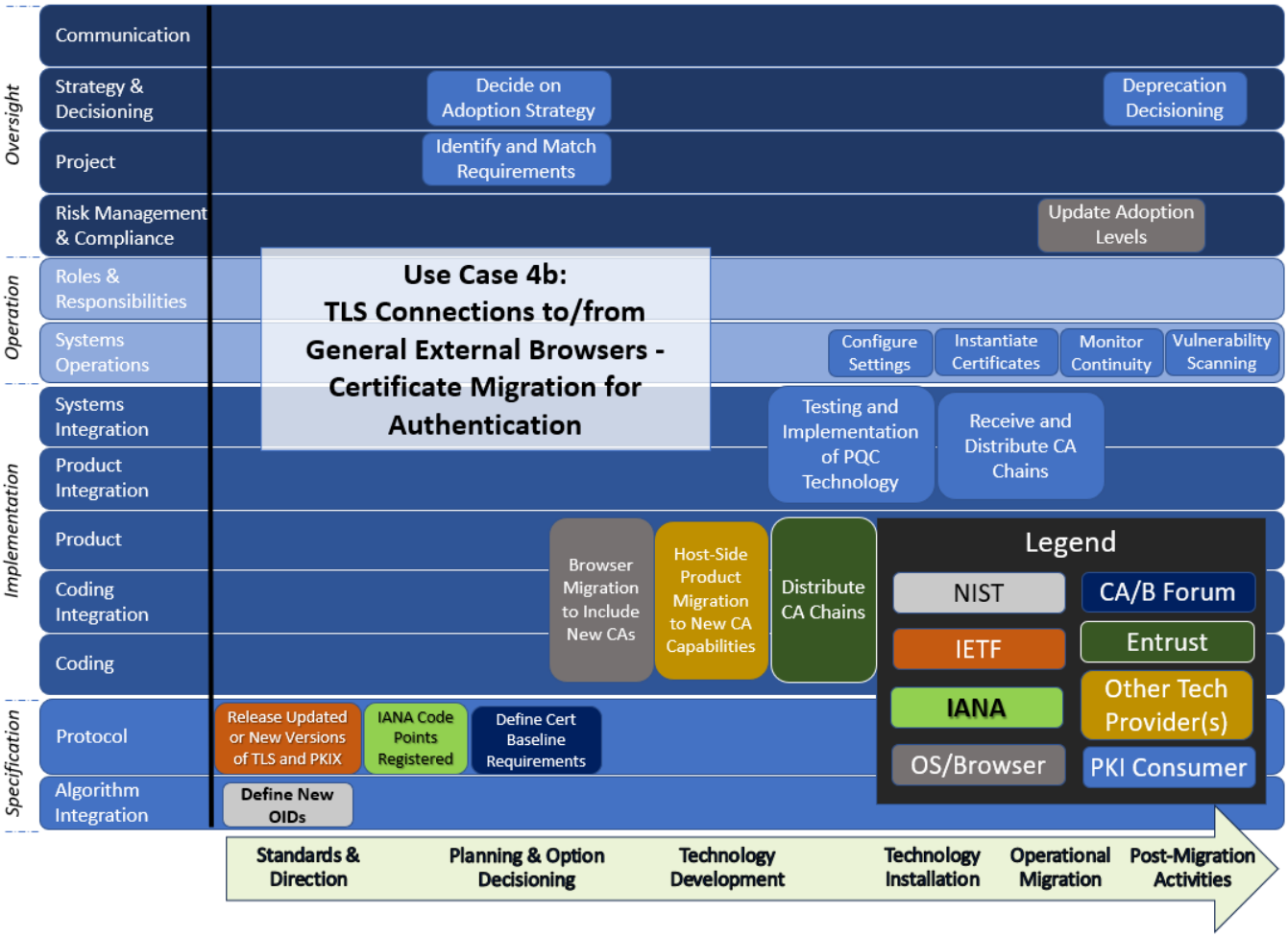
- The IETF will need to release an update to TLS 1.3 or else a new version TLS 1.4 which can incorporate new algorithms and methods to implement post-quantum cryptography as recommended by NIST, potentially including hybrid cryptography. This will include the encoding of the signature; and
- The LAMPS working group of the IETF will need to update PKIX certificate standards to include new OIDs, to be defined by NIST, for each PQC algorithm standardized by NIST (e.g., FIPS 204 ML-DSA[43], FIPS 205 SLH-DSA[44]).

[43] Internet X.509 Public Key Infrastructure: Algorithm Identifiers for ML-DSA, draft-ietf-lamps-dilithium-certificates-04, July 2024

[44] Use of the SLH-DSA Signature Algorithm in the Cryptographic Message Syntax (CMS), draft-ietf-lamps-cms-sphincs-plus-07, July 2024

CA/Browser Forum:

- Define new policies (i.e., baseline requirements) for these certificates such as which algorithms are to be used for which use case and the type of hybrid or PQC-only certificates which are to be made available, etc. [45]



2) Planning and Decisioning:

Technology Provider (viz., Entrust):

- Select parameters as provided by the CA/Browser Forum and create new CA chains.
- Distribute new CA roots of trust to the OS/Browser community to make them publicly available.

[45] CA/Browser Forum - Certificate Issuers, Certificate Consumers, and Interested Parties Working to Secure the Web (cabforum.org)

OS/Browser Community:

- Validate the new CA certificates leveraging their internal processes or WebTrust audits as appropriate. Note that this will be time-based and may not occur before deployment of Root CAs.[46]

Consumer:

- Interact with Technology Provider(s) to ensure their products meet the requirements of the systems (e.g., latency and throughput using new PQC algorithms).
- Decide on the adoption strategy for the enterprise.

3) Technology Development:

OS/Browser Community:

- Make the appropriate changes to browsers which will enable the new cryptographic algorithms in their browsers and the publishing of the new root CAs in root stores.
- Roll out the new version of compatible browsers, presumably leveraging the existing practice of browser updates.

Technology Provider (viz., Entrust):

- Code and enable the new PQC algorithms in their products according to specifications.
- Ensure there is support for both classical and quantum-safe CAs to work simultaneously.

4) Technology Installation:

Consumer:

- Allow browsers to make appropriate updates.
- Understand and test the changes to the requisite products including backward compatibility.

5) Operational Migration:

Consumer:

- Receive and distribute new CA chains from Entrust.
- Deploy keys and certificates and implement new technology in production.
- Configure settings to ensure quantum-safe certificates are active and/or preferred.

[46] CA-Browser Forum Baseline Requirements for TLS Browser Certificates, Section 8 of TLS BR v2.0.5 on Compliance Audit and Other Assessments; https://cabforum.org/working-groups/server/baseline-requirements/documents/

6) Post-Migration Activities:

OS/Browser Community:

- Provide periodic updates regarding adoption level.

Consumer:

- Monitor connections to ensure business continuity and back out if necessary.
- Test connections for quantum safety and/or cryptographic agility.
- Perform vulnerability scanning on implementations to determine the certificates being utilized.
- Keep abreast of OS/Browser adoption level. When a certain threshold is met, deprecate the old non-quantum-safe certificates in the configuration.

J.2.4b.4 Backward-Compatibility Considerations

The following are considerations:

- This use case has the characteristic that backward compatibility is automatically maintained as long as both classical and PQC certificates can be used simultaneously. It is assumed that new connections can negotiate quantum-safe certificates, but can always fall back to existing certificates during the migration period.
- Each cluster of hosts can be migrated independently of each other.

J.2.4b.5 Potential Downgrade Attacks

The main risk here lies in the assumption that both classical and PQC certificates will be simultaneously allowed in TLS. The risk is that an attacker may somehow force the use of the classical certificate when PQC certificates are supported by all parties in a communication.

J.2.4b.6 Summary of Key Issues or Things You May Not Have Initially Thought Of

1) The migration of certificates for authentication, per this sub-use case, can be performed independently of the cipher suite migration sub-use case described in Section J.2.4a. The migration of certificates would prevent active quantum authentication attacks, whereas the migration of cipher suites to PQC would protect against HNDL attacks.
2) Migrating cipher suites should be the easier of the two sub-use cases.
3) Migrating the certificates used for authentication could be done later, to mitigate the risk of authentication attacks when a quantum computer is available.

J.2.5 Use Case 5: Non Browser-Based TLS Connections

J.2.5.1 Description

This use case will cover the Transport Layer Security (TLS) connections which do not involve browsers. Examples include service-to-service backend Applications Programing Interface (API) calls or programmatic connections from apps and other kinds of client devices. These may be internal or external and also one-way or mutual TLS.

As with use case 4, migration would consist of two pieces which can be done separately:

- Sub-Use Case 5(a): Cipher Suite Migration for Key Establishment; and
- Sub-Use Case 5(b): Cipher Migration for Authentication.

The particulars of these two sub-use-cases are described below.

Note: Use Case 5 from Annex I is "Vendor Appliances Establishing TLS Connections". It combines the non-browser use case with the vendor use case. We will focus on the non-browser part.

J.2.5a Sub-Use Case 5a: Cipher Suite Migration for Key Establishment

J.2.5a.1 Description of this Sub-Use Case

This sub-use case covers the establishment of TLS ciphers suites. In particular, ephemeral keys are established under perfect forward secrecy to encrypt the connection. Migrating this portion of the TLS connection to quantum-safe ephemeral keys would protect against passive Harvest-Now-Decrypt-Later (HNDL) or Steal-Now-Decrypt-Later (SNDL) attacks.[47]

J.2.5a.2 Assumptions of this Sub-Use Case

None.

J.2.5a.3 Migration Activities

The migration to PQC for this sub-use case can be modelled through the diagram on the next page.

The corresponding individual migration steps would be as follows:

1) Standards and Direction:

IETF or Other Standards-Related Entity:

- The IETF will need to release an update to TLS 1.3 or else a new version TLS 1.4 which can incorporate new algorithms and methods to implement post-quantum cryptography

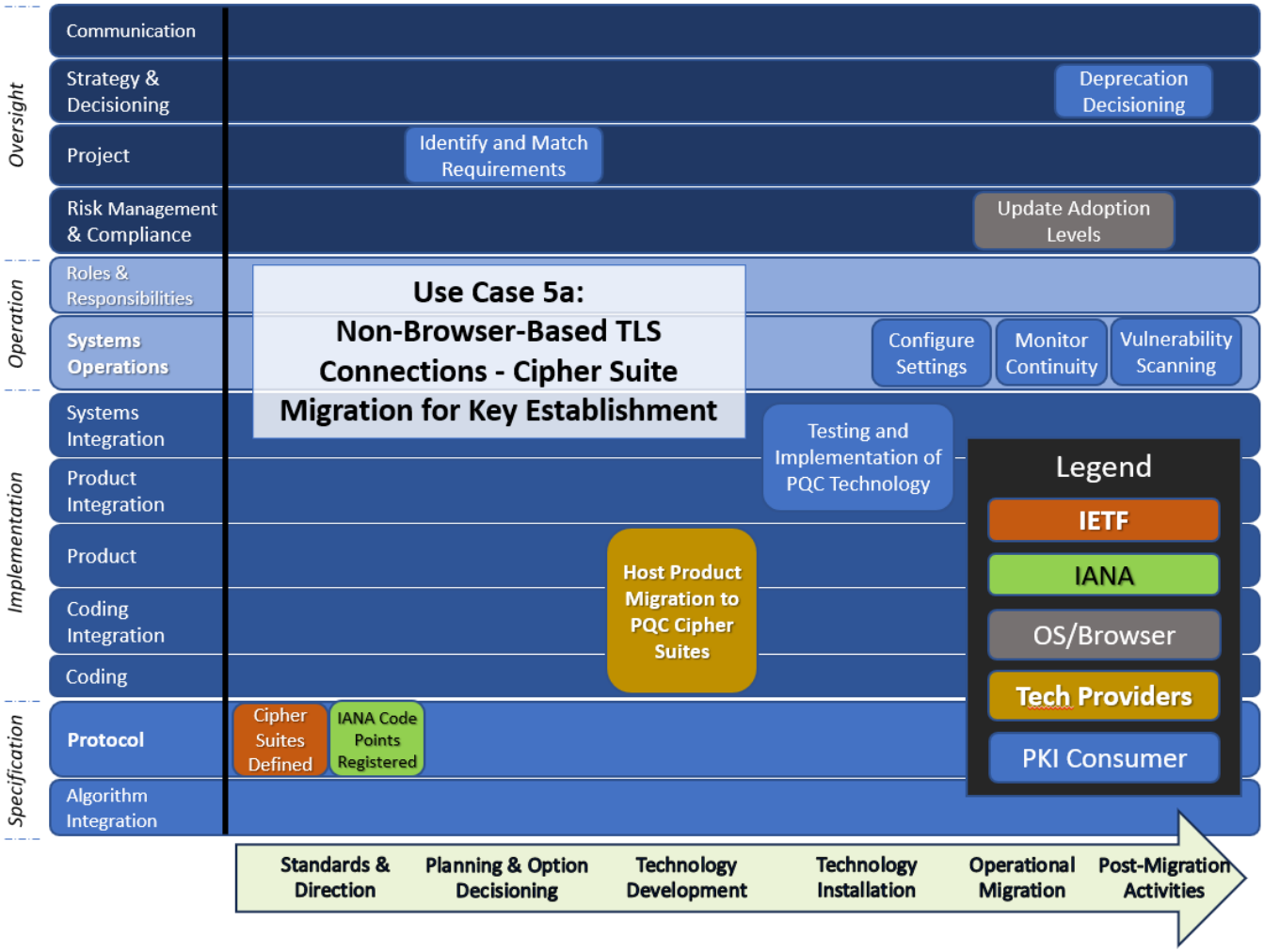
[47] Steal-Now-Decrypt-Later (SNDL) is a synonym for HNDL.

as recommended by NIST, potentially including hybrid cryptography.[48] This will include the encoding of the public key and ciphertext.

- Send an initiation request to IANA to register code points for new PQC algorithms. This could potentially take the form of an RFC or an Internet-Draft (I-D) published by the IETF.

Internet Assigned Numbers Authority (IANA):

- Register new code points for the new PQC algorithms.



2) Planning and Decisioning:

Consumer:

- Interact with Technology Provider(s) to ensure their products meet the requirements of the systems (e.g., latency, bandwidth, and throughput using the new PQC algorithms).

[48] See Annex H for an *Overview of Hybrid Cryptography.*

3) Technology Development:

Technology Providers:

- Code and enable the new PQC algorithms in their products according to specifications.

4) Technology Installation:

Consumer:

- Understand and test the changes to the requisite products including backward compatibility.
- Implement new technology in production systems.

5) Operational Migration:

Consumer:

- Configure settings to ensure quantum-safe cipher suites are active and/or preferred.
- Ensure consuming services enable compatible PQC cipher suites.

6) Post-Migration Activities:

Consumer:

- Monitor connections to ensure business continuity and back out if necessary.
- Test connections for quantum safety and/or cryptographic agility.
- When given the go-ahead, deprecate the old non-quantum-safe cipher suites in the configuration,
- Perform vulnerability scanning on implementations to determine if the new cipher suites are being utilized.

J.2.5a.4 Backward-Compatibility Considerations

The following are considerations:

- This use case has the characteristic that backward compatibility is automatically maintained as a result of cipher suite negotiation. New connections may negotiate quantum-safe cipher suites but can always fall back to legacy cipher suites during the migration period.
- Each cluster of hosts can be migrated independently of each other.

J.2.5a.5 Potential Downgrade Attacks

There have, historically, been many downgrade attacks in this use case such as DROWN [49] and POODLE. [50] There is definite risk that future downgrade attacks could occur for PQC cipher suites as well. This should be monitored across the industry.

J.2.5a.6 Summary of Key Issues or Things You May Not Have Initially Thought Of

1) This sub-use case (Cipher Suite Migration for Key Establishment) case can be performed independently of the Cryptographic Migration for Authentication sub-use case described in Section J.2.5b. Migrating the cipher suite can mitigate the HNDL risk against data encryption which exists today.
2) Cryptographic migration for authentication can be performed at a later date to prevent active quantum authentication attacks.
3) Migrating cipher suites should be the easier of the two sub-use cases described here.

J.2.5b Sub-Use Case 5b: (TLS) Cryptographic Migration for Authentication

J.2.5b.1 Description of this Sub-Use Case

This sub-use case will cover the implementation of quantum-safe certificates for authentication. This will prevent active attacks such as man-in-the-middle (MITM) or impersonation attacks due to quantum recovery (i.e., decryption) of the static keys within certificates.

J.2.5b.2 Assumptions of this Sub-Use Case

1) Version 1.3 of TLS (and newer) will allow processing of both the classical and PQC-enabled certificate to be negotiated during the TLS handshake. All parties will work under this assumption.

J.2.5b.3 Migration Activities

The migration to PQC for this sub-use case can be modelled through the diagram on the next page. The corresponding individual migration steps would be as follows:

1) Standards and Direction:

National Institute of Standards and Technology (NIST):

Define new Object Identifiers (OIDs) for new PQC algorithms as well as the binary encodings of public keys and signatures.

---

[49] Attack of the week: DROWN – A Few Thoughts on Cryptographic Engineering, March 1, 2016

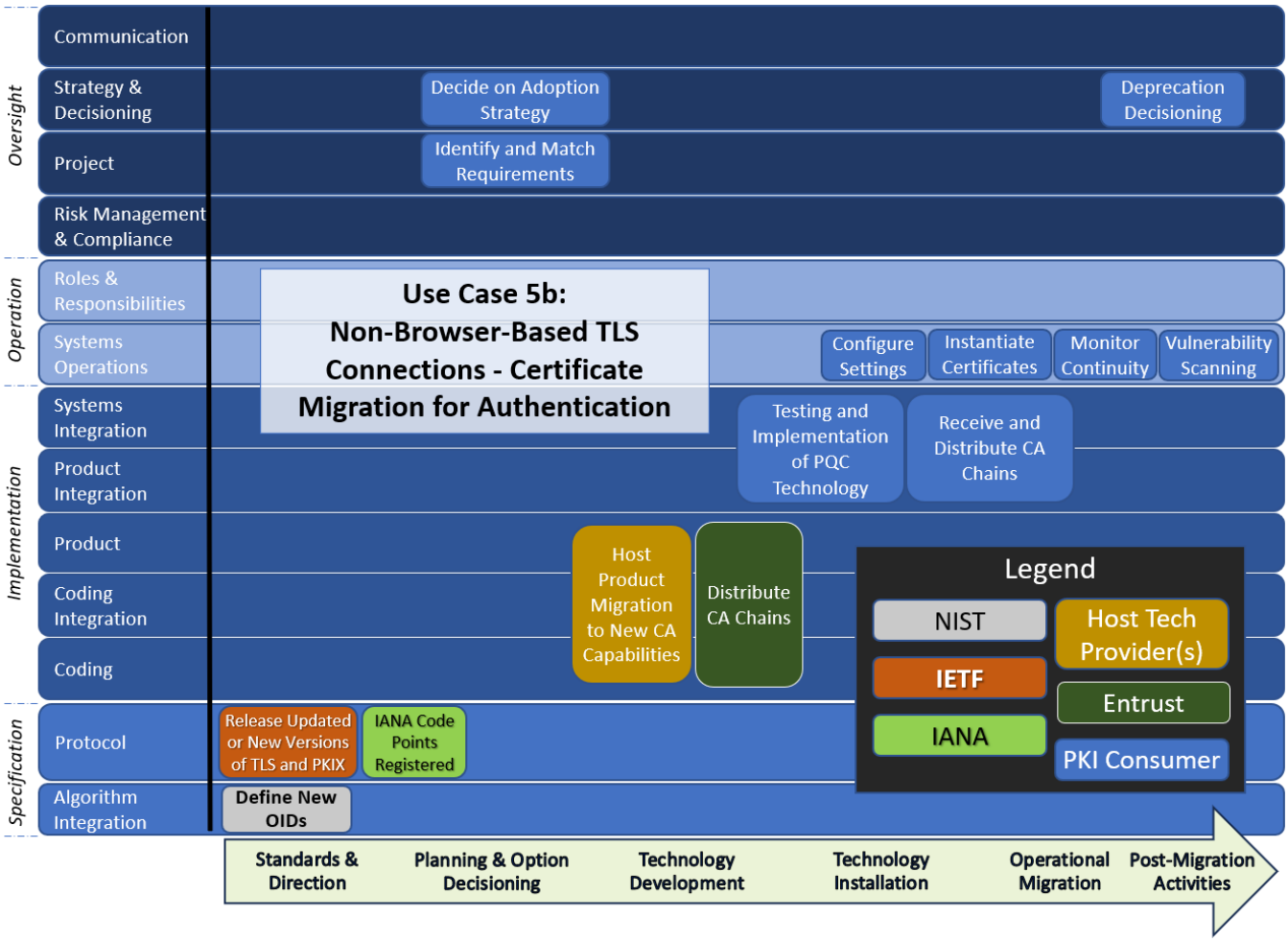
[50] SSL 3.0 Protocol Vulnerability and POODLE Attack | CISA, September 30, 2016

Internet Assigned Numbers Authority (IANA):

- Register new code points for the new PQC algorithms.

Internet Engineering Task Force (IETF):

- Release an update to TLS 1.3 or else a new version TLS 1.4 which can incorporate new algorithms and methods to implement PQC as recommended by NIST, potentially including hybrid cryptography. This will include the encoding of the signature; and
- The LAMPS working group of the IETF will need to update PKIX certificate standards to include new OIDs, defined by NIST, for each PQC algorithm standardized by NIST (e.g., FIPS 204 ML-DSA [51], FIPS 205 SLH-DSA [52]).



[51] Internet X.509 Public Key Infrastructure: Algorithm Identifiers for ML-DSA, draft-ietf-lamps-dilithium-certificates-04, July 2024

[52] Use of the SLH-DSA Signature Algorithm in the Cryptographic Message Syntax (CMS), draft-ietf-lamps-cms-sphincs-plus-07, July 2024

2) Planning and Decisioning:

Technology Provider (viz., Entrust):

- Distribute new CA chains to all entities.

Consumer:

- Interact with Technology Provider(s) to ensure their products meet the requirements of the systems (e.g., latency and throughput using new PQC algorithms).
- Decide on the adoption strategy for the enterprise.

3) Technology Development:

Technology Providers:

- Code and enable the new PQC algorithms in their products according to specifications.
- Ensure there is support for both classical and quantum-safe CAs to work simultaneously.

4) Technology Installation:

Consumer:

- Understand and test the changes to the requisite products including backward compatibility.

5) Operational Migration:

Consumer:

- Receive and distribute new CA chains from Entrust.
- Deploy certificates and implement new technology in production systems.
- Configure settings to ensure quantum-safe certificates are active and/or preferred.

6) Post-Migration Activities:

Consumer:

- Monitor connections to ensure business continuity and back out if necessary.
- Test connections for quantum safety and/or cryptographic agility.
- Perform vulnerability scanning on implementations to determine the certificates being utilized.
- When given the go-ahead, deprecate the old non-quantum-safe certificates in the configuration.

J.2.5b.4 Backward-Compatibility Considerations

The following are considerations:

- This use case has the characteristic that backward compatibility is automatically maintained as long as both classical and PQC certificates can be used simultaneously. It is assumed that new connections can negotiate quantum-safe certificates but can always fall back to existing certificates during the migration period.
- Each cluster of hosts can be migrated independently of each other.

J.2.5b.5 Potential Downgrade Attacks

The main risk here lies in the assumption that both classical and PQC certificates will be simultaneously allowed in TLS. The risk is that an attacker may somehow force the use of the classical certificate when PQC certificates are supported by all parties in a communication.

J.2.5b.6 Summary of Key Issues or Things You May Not Have Initially Thought Of

1) As with the previous sub-use cases described in Section J.2.4, this sub-use case can be performed independently of the cipher suite migration sub-use case. Migrating certificates at a later date would prevent an active quantum authentication attack, whereas migrating cipher suites to PQC would protect against HNDL attacks.
2) Migrating cipher suites should be the easier of the two sub-use cases.
3) Migrating the certificates used for authentication could be done later, to mitigate the risk of authentication attacks when a quantum computer is available.

J.2.6 Use Case 6: Internally Developed Applications

J.2.6.1 Description

This use case will cover the migration of internally developed applications which leverage cryptographic capabilities. The idea is to replace, as directly as possible, the current cryptography with PQC algorithms. While the principles of the use case apply in general, this exploration will focus on the Entrust Security Toolkit for the Java Platform as the main software development tool.

J.2.6.2 Assumptions

1) The organization has well-defined processes for software development, a team of application developers, a set of complementary software development tools, and a code repository.

2) This use case will be independent of the type of development philosophy used in the Software Development Lifecycle (SDLC), such as DevOps[53] or Continuous Integration and Continuous Delivery (CI/CD).[54]

3) As asymmetric PQC algorithms have some fundamental differences from classical algorithms in terms of applicability (e.g., limitations on applications, different characteristics making them appropriate for different use cases), an Application Developer might not be fully capable of properly coding PQC algorithms. To that end, specific PQC Developers might be needed to augment Application Developers for certain applications.

4) Quality Assurance (QA) testers may need to test additional items corresponding to PQC functionality, but the testing process will remain the same. In the case of extremely fault-sensitive applications, some expert knowledge may be required to construct tests that reproduce rare failure cases.

5) Application Security (AppSec)[55] services will look for new types of vulnerabilities and misconfigurations as a result of PQC, but the remediation process will remain the same.

6) Some existing code has not been written in a cryptographically-agile way and might be difficult to migrate to use PQC libraries, or to find and remove hard-coded references to specific algorithms and key sizes.

7) PQC algorithms need solid Random Number Generator (RNG) functions. The organization should verify the leveraged RNGs for running PQC algorithms within their software development pipeline.

J.2.6.3 Migration Activities

The migration to PQC for this use case can be modelled through the diagram on the next page. The corresponding individual migration steps would be as follows:

1) Standards and Direction:

Internet Engineering Task Force (IETF):

- Complete work on integrating algorithms into the protocols and message formats used by the toolkits.

2) Planning and Decisioning:

Consumer:

- Analyze and assess the benchmarking, performance, and risk associated with coding using PQC toolkits.

---

[53] DevOps - Wikipedia

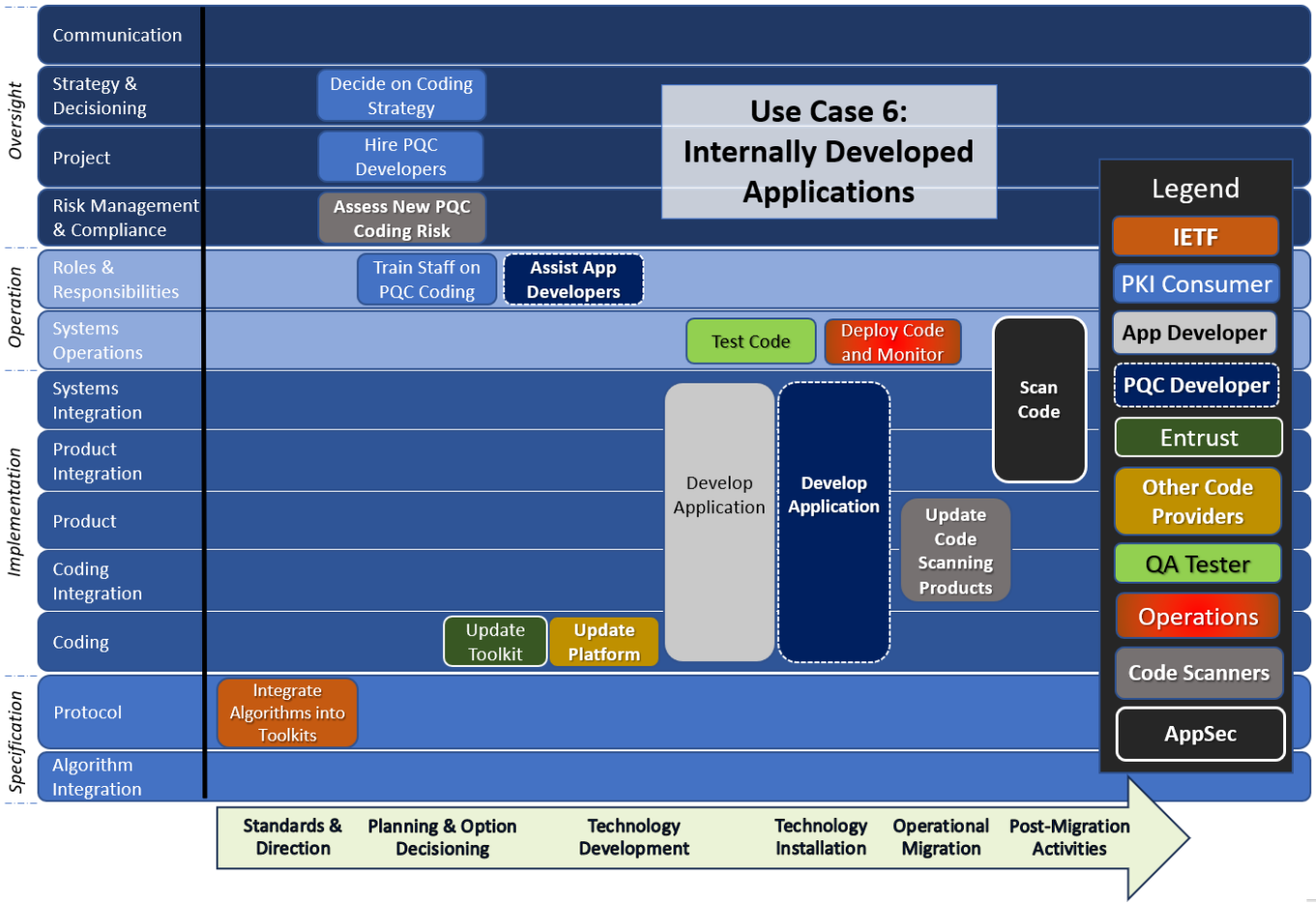
[54] CI/CD - Wikipedia

[55] Application security - Wikipedia

3) Planning and Decisioning:

Consumer:

- Analyze and assess the benchmarking, performance, and risk associated with coding using PQC toolkits.
- Create organizational guidance and oversight to handle risks such as crypto agility, performance implications, key exhaustion and decryption failure. Embed this into the SDLC pipeline. In particular, include guidance that code should be written in a way to more easily enable changes to cryptography.
- Educate Application Developers and QA testers on the differences of coding PQC.
- Hire additional PQC Developers as appropriate to fill gaps.



3) Technology Development:

Toolkit Provider (viz., Entrust):

- Develop new version of Java toolkit to make PQC algorithms available. This would require that any open-source or dependent toolkits also be PQC-enabled.
- Provide install kit to update the software.

- Create release notes regarding options for implementation, benchmarking, performance, and special considerations such as key exhaustion and decryption failure.
- Create sample implementation code for Application Developers.
- Distribute code and documentation package for public consumption.

Other Code Providers:

- Update platforms to allow compatibility with corresponding changes to Entrust and other PQC toolkits.
- Distribute these updates for public consumption.

Code Scanning Vendors:

- Update products to find new vulnerabilities which may arise from PQC.
- Update products to find instances of classical cryptography.
- Optionally, update product to suggest patterns to make code quantum-safe.

4) Technology Installation:

Consumer (viz., Application Developer):

- Install updates from Entrust and other code providers.

5) Operational Migration:

Consumer (viz., Application Developer):

- Analyze requirements for PQC and applicability to a particular use case to understand the best choice for algorithms and integration patterns (leveraging the expertise of a PQC Developer as appropriate).
- Decide whether or not to re-architect software to be cryptographically agile beyond just making the current required changes.
- Code PQC in applications.
- Merge PQC code into existing codebase and code repository.

Consumer (viz., PQC Developer):

- Provide guidance to Application Developers where appropriate.
- When coding, follow the steps listed for an Application Developer.
- Analyze requirements for PQC and applicability to a particular use case to understand the best choice for algorithms and integration patterns (e.g., when there are more ‘knobs and switches’).
- Code PQC in applications.
- Merge PQC code into existing codebase and code repository.

Consumer (viz., QA Tester):

- Regression test existing code when the updated PQC-enabled version is installed.
- Test that PQC code accomplishes the desired business functionality using standard process.
- Test that PQC is actually being used instead of classical cryptography.
- Test that the new code is cryptographically agile and can be easily re-configured to accommodate future changes to cryptographic policy.

Consumer (viz., Operations):

- Implement, manage, and troubleshoot new code in production.

6) Post-Migration Activities:

Consumer (viz., Operations):

- Monitor business continuity.

Consumer (viz., Application Security):

- Implement new code scanning services and work to remediate findings through normal processes.

J.2.6.4 Backward-Compatibility Considerations

The following are considerations:

- The update of the Entrust PKI Toolkit (as well as other toolkits) will be fully backward compatible and allow old code to compile and run, and to continue to support processing data created with older versions of the Entrust Toolkit (as well as other toolkits).
- The backward compatibility of the developed code will be dependent upon the particular application which is being developed.

J.2.6.5 Potential Downgrade Attacks

The types of downgrades can be divided into two categories:

- **Compile-Time Downgrade**: At the time of compilation, an older, classical crypto library or cryptographic configuration may be selected. This may be caused more by indifference than maliciousness as the most likely scenario would be programmers who simply ignore or do not realize that quantum-safe cryptography should be leveraged. This vulnerability may be mitigated by compile failure as older toolkits may not contain newer APIs which applications would potentially leverage.
- **Run-Time Downgrade**: At run time, an attacker may be able to exploit a vulnerability in the code to force the use of quantum-vulnerable cryptography.

J.2.6.6 Summary of Key Issues or Things You May Not Have Initially Thought Of

1) Making use of PQC algorithms will likely not be as straightforward as classical algorithms as there is a certain amount of decisioning required and a number of necessary coding differences depending on the algorithms chosen. In particular, Key Encapsulation Mechanism (KEM) algorithms do not inherently give the same security properties as Diffie-Hellman based algorithms. While examples may be available, there may still be a need for access to a PQC Developer to who can consult on PQC-specific coding. This should be taken into account from a resource perspective.
2) The use of PQC algorithms might result in states which do not occur in with classical cryptography. For example, some PQC encryptions may not be able to be decrypted. These failures are expected to be exceptionally rare but should be taken into account in cases where loss of data is unacceptable. This needs to be taken into account with appropriate design and exception handling to ensure business continuity potentially including adding new mechanisms whereby the decryptor can request re-encryption and re-transmission of the data.
3) The existing code might not be crypto agile in the sense that the code may not be easily able to add new cryptographic algorithms or add new mechanisms to the protocol message formats such as new message types for establishing connections with KEM algorithms. A decision may need to be made regarding the extent to which affected code should be rewritten to this effect.
4) As with any code base, it is important to ensure that these considerations also apply to all of the open-source and dependent libraries.
5) Writing code to be cryptographically agile may not be as straightforward as it sounds.[56]

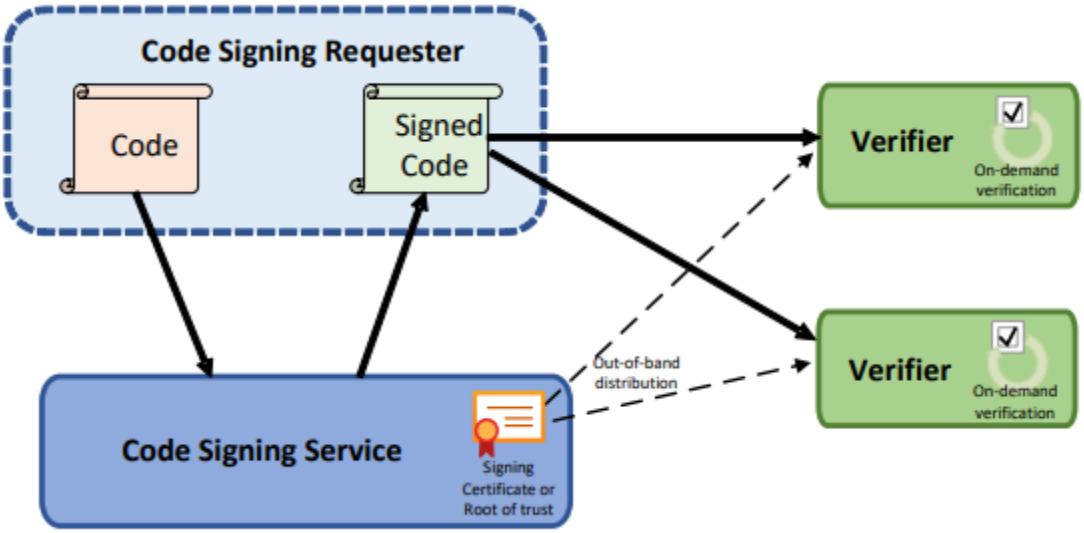
J.2.7 Use Case 7: Code Signing (Private)

J.2.7.1 Description

This use case covers the migration of a private code-signing service. The focus is on an individual organization's process for code signing by and for entities within the organization and/or those in close relation with the organization. Public code-signing services, such as those used for desktop operating systems and mobile devices, are a different use case and are not covered herein.

[56] On the State of Crypto-Agility, Presented at BSI Kongress 2022, https://eprint.iacr.org/2023/487.pdf

We note that NIST has published a standard for hash-based signatures (viz., SP 800-208) in preparation for this use case.[57]



J.2.7.2 Assumptions

1) The structure of the certificate will not have a significant effect upon the migration. For example, Windows systems use X.509 certificates whereas many Linux package managers leverage PGP certificates. In either case, the important assumption is that a certificate exists, is backed by appropriate cryptographic hardware, and can adequately sign regardless of its structure.
2) To handle the change in cryptography, classical and PQC signatures will be logically independent. The existing method of classical signature will remain in place, while a new set of PQC signatures will be created. There may be several options in creating PQC signatures (e.g., signing just the code or nesting the signatures). The code verifier will accept if one of the signatures verifies correctly. This was how the migration from SHA-1 to SHA-2 was handled. Note that it will further be possible for a particular type of signature to labelled invalid in order to enforce algorithm and certificate deprecation.
3) Any new code-signing algorithms will be compliant with existing code-signing standards and follow the existing code-signing paradigm as follows. In particular, there will be a time-stamping service available for use by the signers. This time-stamping service may itself need to be migrated to PQC.

[57] Recommendations for Stateful Hash-Based Signature Schemes, October 29, 2020, https://csrc.nist.rip/publications/detail/sp/800-208/final

4) The new versions of the technology product which enable code signing and verifying will be fully backward compatible with the existing code. Note that this is a product decision and is independent of the cryptography available.
5) The code signer may belong to the organization or may be a third-party service. For the purpose of this use case, the ownership of it will be immaterial.

J.2.7.3 Migration Activities

The migration to PQC for this use case can be modelled through the diagram on the next page. The corresponding individual migration steps would be as follows:

1) Standards and Direction:

- None

2) Planning and Decisioning:

Technology Provider (viz., Entrust)

- Determine a set of options available in terms of algorithms supported for both signing and verifying code.
- Create release notes on the different types of options.

Consumer:

- Make decisions with regards to the options for signing algorithm based on release notes.
- Decide on rollout plan (e.g., go-forward basis, signing existing code, etc.).

Code Signers / Verifiers:

- Make decisions with regards to the options for the signing algorithm to be used, based on release notes.

3) Technology Development:

Toolkit Provider (viz., Entrust):

- Develop new version of code-signing and verifier software including coding of new options. Note the code-signing ecosystem may require coordination among Technology Providers. Note these details are beyond the scope of this discussion.
- Make the product and release notes available to consumers.

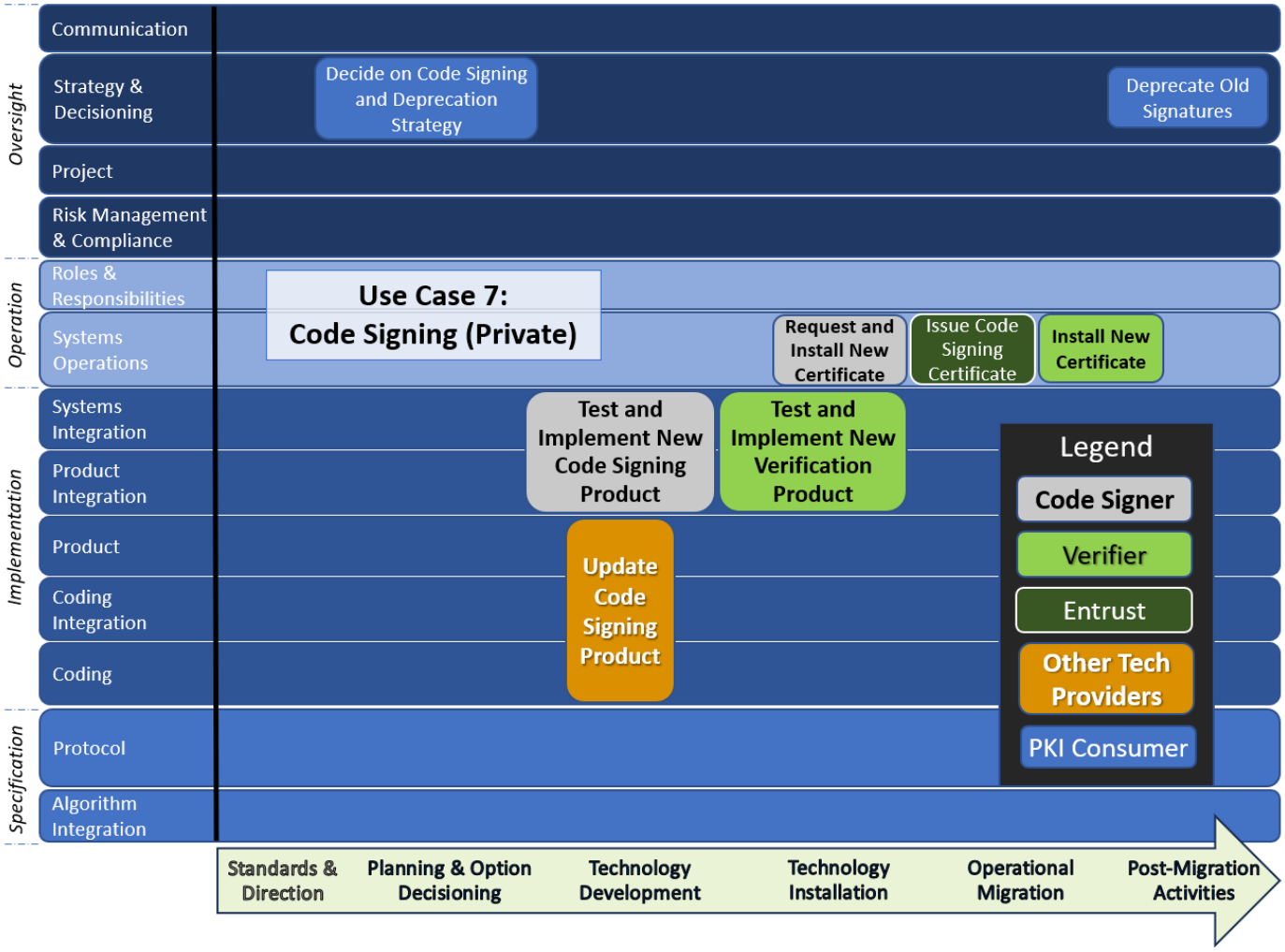
4) Technology Installation:

Consumer (viz., Code Signer):

- Implement new version of code-signing software.

Consumer (viz., Verifiers):

- Update products to new version.



5) Operational Migration:

Consumer:

- Obtain new code-signing certificate.

Technology Provider (viz., Entrust):

- Issue new code-signing certificate.

Consumer (viz., Code Signer):

- Submit new certificate request and obtain code-signing certificate.
- Sign code according to roll out plan.

Consumer (viz., Verifiers):

- Install new code-signing certificate.
- Test verification with newly signed code and backward compatibility.

6) Post-Migration Activities:

Consumer

- Decide when code signers will no longer produce old signatures.

Consumer (viz., Verifiers):

- Monitor continuity.
- Deprecate old signatures according to organizational directive.

J.2.7.4 Backward-Compatibility Considerations

The following are considerations:

- Under the assumption that different signatures will be detached from each other, backward compatibility will not be an issue. Older verifiers can simply validate older signatures, although it should be tested that unrecognized signatures will not cause an outage.
- Under the assumption that the two signatures are decoupled, it should be straightforward to continue providing signatures on both algorithms for an extended period of time until all verifiers support the PQ algorithms, much as was done during the RSA-SHA1 to RSA-SHA2 transition.
- When implementing the new PQC signature algorithms, care must be taken that these are the signatures being verified. It would be somewhat easy to mistakenly continue verifying the older classical signatures. This could potentially cause an outage when classical signatures are deprecated or there are security issues for older code.

J.2.7.5 Potential Downgrade Attacks

The use of parallel certificates and parallel signatures could be leveraged by an attacker to have the Verifier accept forged quantum-vulnerable signatures.

It would be somewhat easy to mistakenly code a Verifier to prefer the verify the PQ signature, when present, but continue to accept code which is signed with only a legacy signature, especially if binaries built many years ago are supposed to continue working on upgraded systems. Verifiers may be incapable of distinguishing between legitimate legacy binaries, and modern binaries which have had their PQ signature removed maliciously. Solutions to this problem will be environment-dependent and could, for example, include the use of Trusted Timestamp Authorities (TTAs) to assert the build time of the binary, and verifiers coded to accept legacy signatures only if the binary was produced prior to a certain date.

J.2.7.6 Summary of Key Issues or Things You May Not Have Initially Thought Of

1) The key assumption here is that signatures will be independent. This is not a given. The industry might be dependent on organizations such as Microsoft to set the direction.
2) Some code-signing services might be completely managed by a third party such as a SaaS solution. In the past, these services have been compromised. Code signing with PQC certificates might present a new avenue of attack to insert malicious code.
3) There may be particular attention needed for code-signing services with respect to hash-based signatures. Public code-signing and timestamping services are particularly susceptible to DoS-via-key-exhaustion since these services are hard to rate-limit.
4) If an HSM is used in the code-signing system, backup of the HSM may not be straightforward if the signing occurs using hash-based signatures. There is currently discussion on how this is to be handled.[58]

J.2.8 Use Case 8: Vault Encryption

J.2.8.1 Description

This section will cover the specific use case of protecting and accessing keys within a key vault. The same principles would apply to other types of storage encryption.

The vault itself could potentially leverage hardware encryption through HSMs, or alternatively, software-based key protection.

As with use case 2 (viz., Establishing a new Private Certification Authority with PQC, as described in Section J.2.2.1) there are two paths to migration:

1) Creating and storing PQC-related keys leveraging existing technology that offers classical protection.
2) Leveraging new quantum-safe protection mechanisms to create and store keys.

It is possible to do the former with largely existing technology and, hence, in advance of the latter. However, this comes with the risk that the classical protection of the vault will be compromised. If choosing this strategy, the organization will need to decide if it wishes to accept this risk or perform a second (subsequent) migration when quantum-safe protection technology is available.

[58] Hash-based Signatures: State and Backup Management, February 2024, https://www.ietf.org/archive/id/draft-wiggers-hbs-state-00.html

The Key Management Interoperability Protocol (KMIP) standard handles formatting and transmission of keys. KMIP is governed by OASIS which, at the time of this writing, is working on a revision of this standard for PQC. [59]

J.2.8.2 Assumptions

1) While technically different, the software and hardware version of the vault will be similar enough in structure that the same steps would apply in both scenarios.
2) The KeyControl product will be fully backwards compatible. In other words, upon upgrade, all of the existing keys in the vault, functions, and API calls by requesting entities will function seamlessly even if there are no PQC elements involved.
3) The KeyControl product will automatically imply that the new version of KMIP is available upon installation, including the requisite protocol interfaces and APIs. This will be seamless to the consumer.
4) Consuming applications will need to be migrated separately to leverage the new version of KMIP and PQC keys. This is out of the scope of this use case.
5) The data-migration capability needed to migrate encryption keys within the vault will be automated.
6) KeyControl might leverage classical cryptography from an access control and authentication perspective. The particulars are similar to those in the TLS use cases described in Sections J.2.2.4 and J.2.2.5. We will not go deep into detail regarding these considerations here.
7) The deprecation of classical keys is dependent upon the applications which consume them. This is out of the scope of this use case.

J.2.8.3 Migration Activities

The migration to PQC for this use case can be modelled through the diagram on the next page.

The corresponding individual migration steps would be as follows:

1) Standards and Direction:

Organization for the Advancement of Structured Information Standards (OASIS):

- Update the KMIP protocol to account for post-quantum keys and algorithms. [60]

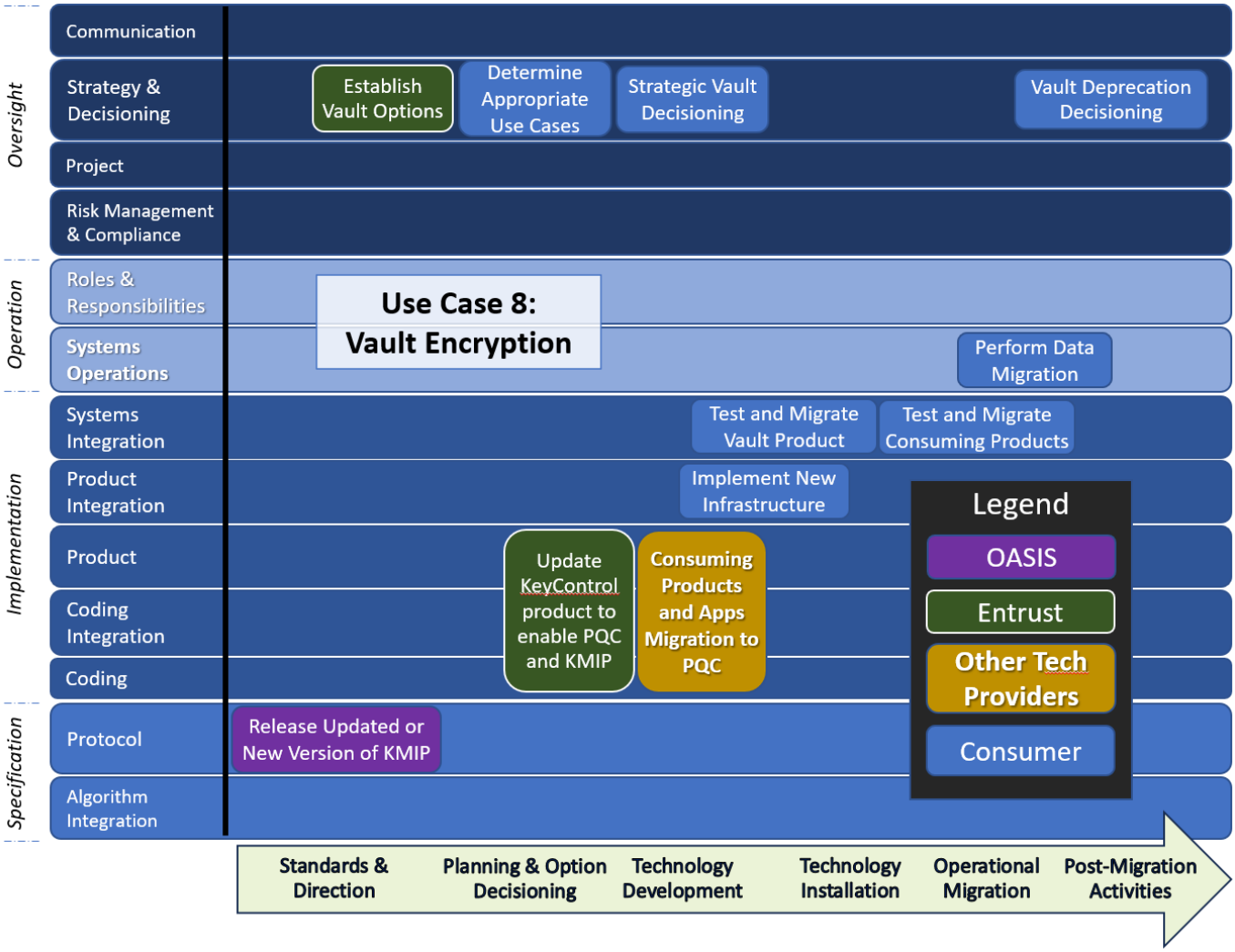
[59] OASIS Key Management Interoperability Protocol (KMIP) TC - OASIS (oasis-open.org)

[60] ibid

2) Planning and Decisioning:

Toolkit Provider (viz., Entrust):

- Establish options for consumers on the following regarding KeyControl:
  - Type of keys and certificates which are supported;
  - Use cases appropriate for each type of key (no Swiss Army knife);
  - Options for access control and authentication mechanisms;
  - Compatibility requirements for complementary vendor technology; and
  - Which options are available for customization.



Consumer:

- Determine appropriate use cases for KeyControl which will be needed and match against the options given by Entrust.
- Establish enterprise policy with respect to key types, algorithms, mode of operations, access control and authentication, instances, and deprecation of classical keys.

3) Technology Development:

Technology Provider (viz., Entrust):

- Provide firmware update for nShield Connect to create CAs and perform PQC KMIP operations for the different selected options (similar to use case 2).
- Provide new versions of the nShield Connect with new SecurityWorld and PKCS #11 PQC capabilities (similar to use case 2).
- Code and provide new version of KeyControl for operations, access control and authentication, and support by nShield Edge, nShield Connect, and software-based vault. This would include a utility to perform a data migration if necessary.
- Update protocol interfaces and APIs supporting KMIP.

Other Technology Providers (Consuming Applications):

- Code and enable the new algorithms in their products according to KMIP specifications and Entrust options.
- Enable appropriate protocol interfaces and APIs for KMIP PQC.

4) Technology Installation:

Consumer:

- Choose one of the following activities:
  - Receive the new version of the nShield Connect firmware from Entrust and upgrade the HSM, or
  - Receive the new version of the FIPS-validated software vault which enables support for PQC keys, or
  - Upgrade to the new model of the nShield Edge and/or nShield Connect and, if required migrate existing keys to the new SecurityWorld, or
  - Receive the new version of the FIPS-validated software vault which enables both support for PQC keys and PQC protection mechanisms.
- Install new version of Entrust KeyControl to compatible quantum-safe versions. This implies that the latest version of KMIP is installed in the package.
- Encode enterprise policy in KeyControl.
- If necessary, create new protection keys for the vault.
- Regression test existing operations.

5) Operational Migration:

Consumer:

- Test and migrate consuming application to be able to leverage PQC keys and new access control and authentication mechanisms using KeyControl.

- If necessary, perform a manual or automated data migration.

6) Post-Migration Activities:

Consumer:

- Monitor connections to ensure business continuity.
- Deprecate capabilities for classical keys when the enterprise strategy calls for it.

J.2.8.4 Backward-Compatibility Considerations

The following are considerations:

- The nShield Connect HSM and FIPS-validated software vault provided by Entrust will be fully backward compatible with the old vault technology. Similarly, quantum-safe HSMs will also provide backwards compatibility. Hence, the installation will be agnostic to consuming applications.
- The backward-compatibility considerations of the consuming applications are particular to that application. This is out of the scope of this use case.
- If changing the protection mechanisms to PQC, then a data migration involving the protection keys may need to take place. This can be done manually or automatically by leveraging the Entrust KeyControl capability.
- New PQC keys can be added, and old classical keys can be removed, at the convenience of the consuming applications subject to enterprise directive.
- Existing access-control and authentication mechanisms will be available during the migration. They are non-persistent, so they can simply be re-initialized by the consuming application. The exact details would be dependent upon the consuming application and so is out of scope of this use case. The mechanisms involving classical cryptography can be turned off when all applications are migrated.
- When all classical keys have been removed, the appropriate KeyControl capability can be turned off in the policy to prevent future provisioning.

J.2.8.5 Potential Downgrade Attacks

The protection mechanisms, once established, would not easily lend themselves to downgrade attacks after a data migration has occurred. Although they are out of scope for this section, we will remark that the authentication mechanisms for access could be subject to downgrade attacks depending on their details.

J.2.8.6 Summary of Key Issues or Things You May Not Have Initially Thought Of

1) As described in use case 2 (for Establishing a Private CA with PQC per Section J.2.2.6), a migration of vault encryption can occur using existing infrastructure which can support PQC technology, but it might not involve PQC protection of the private keys. An alternative strategy would be to obtain new vault infrastructure which is fully PQC compatible. If the

former strategy is adopted, then a decision may need to be made later to determine if a second migration would be needed from a risk perspective. This may apply to general database or storage technology depending on the characteristics of those technologies.

2) If changing the protection mechanisms involved in the vault, there may need to be a data migration of the data elements within the vault to translate encryption to different encryption keys. This may also apply to more general use cases in database and storage encryption depending on the characteristics of those technologies.

J.2.9 Use Case 9: S/MIME Secure Email

J.2.9.1 Description

This section will cover the specific use case of sending and receiving email messages through the S/MIME protocol.[61] This use case may be viewed as a specific sub-use case of Use-case 3 (i.e., End-Entity Certificate Migration) as described in Section J.2.3.

Entrust's S/MIME product consists of a plugin to an existing email technology such as Outlook in order to simplify the interface to S/MIME emails. The consumer will have its own directory server or an adaptation of an existing directory service such as Active Directory or LDAP. This directory service will contain the public key certificates for all internal organization users.

Entrust's product has long used the practice of "dual-usage" certificates which take advantage of the fact that the RSA algorithm can be used for both signing and encryption, and therefore users can be provisioned with a single certificate for both purposes. There are, however, S/MIME products from other vendors that use the same certificate for signing and encryption. With the migration to post-quantum cryptography, S/MIME users will require separate signing and encryption certificates since there is no "dual-use" PQ algorithm. Fortunately for the Entrust product, the default certificate profiles for S/MIME within the Entrust PKI have always provisioned users with separate signing and encryption certificates, so this should not result in any behaviour or policy changes for most S/MIME deployments using the Entrust PKI. Other vendors may have to alter the product to enable dual usage.

Public-key certificates for users external to the organization can be stored in a particular user's local certificate store, either by specific request or automatically through receipt of an email. This store can be viewed by a local application such as MMC.

Devices which are not issued by the organization (e.g., employee-owned BYOD mobile computing devices that meet the technical requirements of the enterprise) are out of scope for this use case.

[61] Secure / Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification, IETF RFC 8551, April 2019

J.2.9.2 Assumptions

1) Entrust's S/MIME product will be fully backwards compatible. In other words, upon upgrade, all of the existing messages and keys will be accessible and messages can still be sent with existing cryptography.
2) All technologies will be fully backwards compatible in that all existing functionality will still work upon installation of the new updated products.
3) The organization is leveraging a Private Entrust CA to provision certificates for S/MIME. The organization will stand up a new PQC-enabled Private Entrust CA, as described in use case 2 (viz., Section J.2.2). This would include the ability to automatically provision certificates for organizational users.
4) The consumer has existing policies to onboard and decommission users for S/MIME and dictate which users are able to perform which actions. These policies will not change. The permissions will likely be encoded in the user's registry and possibly the CA.
5) All protocols and technology will support PQC certificates working alongside classical certificates for both signing and encryption. Note that this would mean that, at least for a time, each user would have two signing and two encryption certificates.
6) The organization has an existing policy for retention of S/MIME-encrypted emails which deals with handling old or archived emails from different public keys. This might or might not include key escrow.

J.2.9.3 Migration Activities

The migration to PQC for this use case can be modelled through the diagram on the next page. The corresponding individual migration steps would be as follows:

1) Standards and Direction:

Internet Engineering Task Force (IETF):

- Update the S/MIME, Cryptographic Message Syntax (CMS), and enrolment protocol Certificate Message Protocol (CMP) standards to enable PQC.

National Institute of Standards and Technology (NIST):

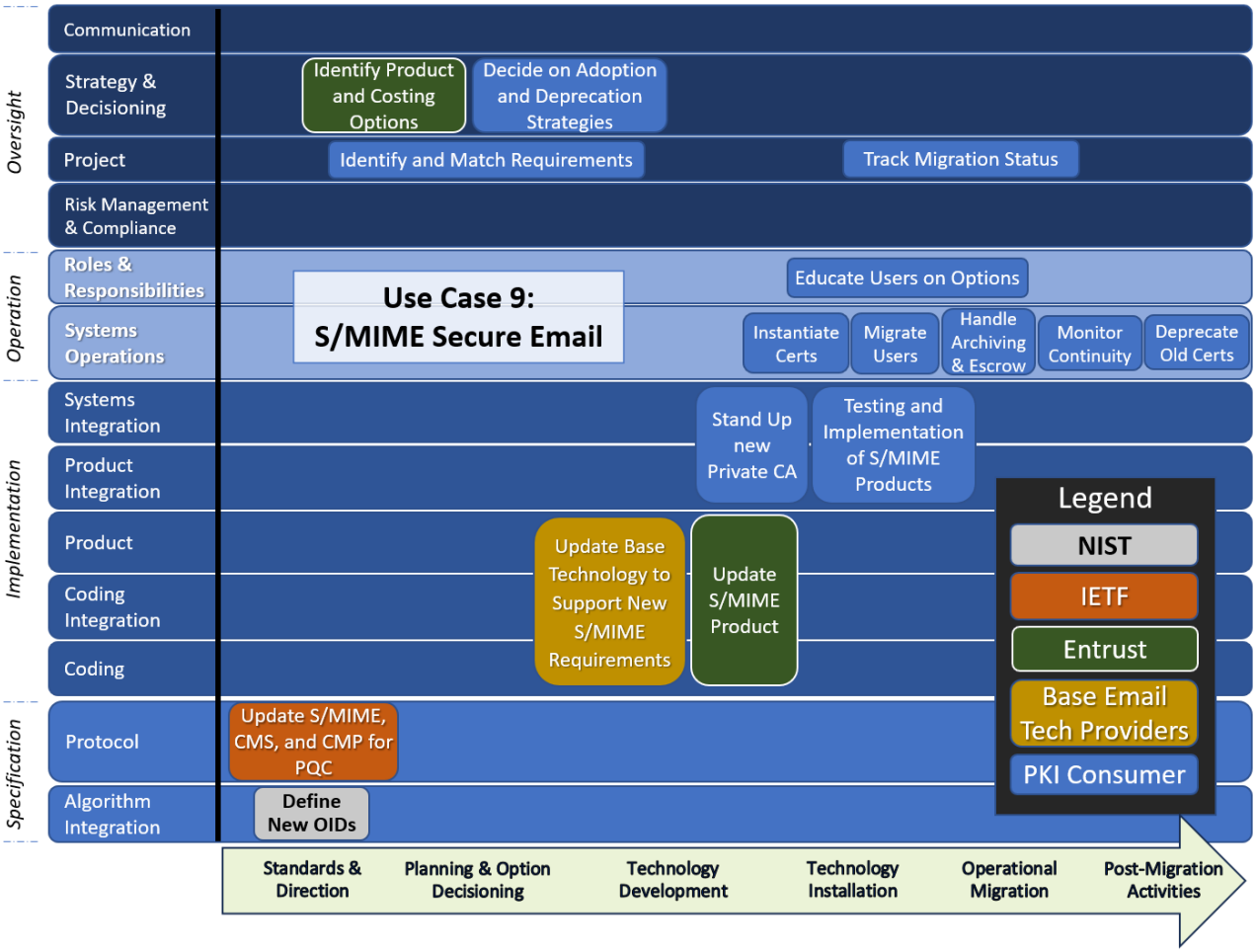
- Define new Object Identifiers (OIDs) for the new algorithms as well as the binary encodings of public keys and signatures.

2) Planning and Decisioning:

Technology Provider (viz., Entrust):

- Establish options for consumers on the following regarding S/MIME:
  - Type of PQC algorithms, modes, and key sizes supported for both encryption and signatures;

- Organizational and user options for sending and receiving quantum-vulnerable emails;
- Messaging and warnings on sending and receiving emails which are quantum-vulnerable or selectable;
- Cost model for multiple certificates per user as outlined in the Private CA use case (viz., Section J.2.2); and
- Options for handling old or archived emails. This might include options for escrow of old keys as well as whether or not nesting of encryption and signatures will be used in this process.



Consumer:

- Incorporate the new costing model into existing budget.
- Develop organizational policy given the options provided from Entrust.
- Develop a rollout strategy for user application upgrades. This would include describing behaviour for situations where:
  - The email sender has upgraded but not the recipient;

  - The email recipient has upgraded, but not the sender; and
  - Both have upgraded.
- Develop policy for handling certificate-verification special cases such as when one signature within a hybrid-signed message verifies correctly and another signature does not verify correctly.
- Develop a rollout strategy for new PQC certificates. This would include how the first PQC certificates are to be provisioned (e.g., sign CSR under old private key or password).
- Develop an organizational policy for handling and retention of old or archived emails under classical public keys. Note that while the existing retention policy can be leveraged, a decision would have to be made as to whether or not maintaining older emails under classical quantum-vulnerable keys will be an acceptable risk.
- Develop a strategy for eventual deprecation of old classical keys.
- Develop a strategy for communication to external users.
- Communication and education of users.

3) Technology Development:

Other Technology Providers (Base Applications):

- Update email application to support changes required for S/MIME plugin.
- Update local key store to support parallel PQC certificates for external users.
- Update the directory service to enable parallel PQC certificates (if this directory service is leveraged).

Entrust:

- Update plugin to enable PQC certificates for S/MIME.
- Update directory service to enable parallel PQC certificates (if Entrust directory service is leveraged).
- Enable Private CA functionality to allow the new PQC CA to accept S/MIME certificate requests under the old classical CA.
- Update email archiving and key escrow protocols.

4) Technology Installation:

Consumer:

- Set up new Private CA, as described in use case 2 (viz., Section J.2.2);
- Enable capability of provisioning S/MIME certificates through the Private CA. This would include accepting certificate requests under the old classical CA.
- Install new version of directory service.
- Roll out update of certificate stores to all users.

- Roll out new version of email application to all users.
- Roll out new version of S/MIME plugin to all users.
- If necessary, supply organizational users with new devices with updated software.

5) Operational Migration:

Consumer:

- Create new certificates for users and roll them out according to the enterprise strategy. This might leverage automated methods or manual enrolment. It might also occur in batches of certain volumes or at renewal.
- Verify that existing archived emails under classical keys can still be accessed.
- Provide IT training for users on how to use the new S/MIME product with the organizational policy.
- Instruct users to reach out to their external contacts with whom they frequently exchange email to communicate upgrade plans, new certificates, and to request they move to quantum-safe S/MIME.
- Execute organizational strategy on old and archived emails under classical quantum-vulnerable keys

6) Post-Migration Activities:

Consumer:

- Track status of migrating certificates used for S/MIME.
- At a certain date, perform a forced update for any users who have not yet updated.
- Deprecate the old certificates and use of classical cryptography in S/MIME.

J.2.9.4 Backward-Compatibility Considerations

The following are considerations:

- The S/MIME product itself can be built in a way that when upgraded, it will still seamlessly allow the use of classical cryptography and old emails will still be accessible. Therefore, the product will still function.
- If the email sender, but not the recipient, has been upgraded, the sender's software will still notice that only a classical certificate is still available in the directory. The software can then encrypt only using classical cryptography so that it can be decrypted by the recipient. The sender's software can sign either with just a classical or with both signatures. If the former occurs, the operation should work as normal. If the latter, it is not clear if the existing receiving software would accept the signature or what kind of warning message it would give. Note that the sender's software would have to handle user messaging or policy enforcement somehow.

  - While BYOD devices are out-of-scope for this use case, it is worth considering the implications of users with email clients on multiple devices where some are capable of decrypting emails protected with PQ algorithms while their other devices are not.

- If the recipient, but not the sender, has been upgraded, then the sender will notice a PQC certificate in addition to the classical certificate of the recipient in the directory service. It is assumed that the sender's software will select only the classical certificate and work as normal. However, this would need to be verified. It is further assumed that if the recipient would only have a PQC certificate, then the sender's software would not be able to process it and so either no email would be sent or the message would be sent in the clear or unsigned.
- For external users (i.e., users external to the organization with whom the user frequently exchanges secure email via manually or automatically imported certificates), the same would apply except the user's local store will be searched for the public keys. These keys could be updated either manually or when the external users sends an email using PQC. It is assumed that the local certificate store would still be able to store a PQC certificate, even if it could not be accessed by the S/MIME client.
- Presumably, the upgraded software would allow access to old and archived emails when older keys are present. However, the organization may deem that the risk of a quantum attack is too great and may require deletion or migration of these emails. The backward-compatibility considerations would depend on the organization's strategy. We note here that the migration of emails would involve a data migration and thus be inherently complex.

J.2.9.5 Potential Downgrade Attacks

There are several potential paths to downgrade in this use case:

- The use of parallel certificates for both sender and recipient lends itself to downgrade attacks by convincing the other party that PQC certificates do not exist. If there is only a legacy certificate for the recipient, how does the sender know if they are a truly legacy client, or if they are a PQ-aware client, but whose PQ certificate is not present? For internal users, this would be mitigated by the fact that the organization has control of the directory service. For external users, this can be mitigated via a signed message containing an S/MIME capabilities attribute identifying support for PQC. At time of writing this Annex, the IETF was working on several extensions to X.509 standards to allow discovery of related certificates.
- Another important consideration is the behaviour when a PQC certificate is not recognized. The user may be prompted to send the email without encryption or signature.

J.2.9.6 Summary of Key Issues or Things You May Not Have Initially Thought Of

1) There are several methods regarding how encryption, signatures, and certificates will work together within an organizational S/MIME deployment. It is possible to combine classical algorithms so that both are required to encrypt or sign a message or have them work independently.
2) As a Private CA is being used, the structure of the certificates could theoretically be set by the CA without involvement from external groups such as the CA/Browser Forum. However, as external clients will be leveraging a different CA, there is a possibility that differences in certificate structure may lead to interoperability issues. This may point to involvement from organizations such as the CA/Browser Forum in order to standardize the certificates, although this will be left as an open consideration.
3) For S/MIME email in particular, there is a non-trivial trade-off between backwards compatibility and security. The straightforward solution of provisioning every end user with two sets of certificates maximizes backwards compatibility but does nothing to prevent downgrade attacks. On the other extreme, Post-Quantum / Traditional composites and other mandatory hybrid modes maximize security at the expense of backwards compatibility. Solutions exist which offer both, but at the cost of increased deployment and management complexity. Customers should work with their vendors to find the right balance of backwards compatibility, forward security, and deployment complexity.
4) There are several methods for rolling out changes to individual email users. This can be done all at once or in batches at scheduled times or at the time that an existing certificate comes for renewal, taking into account both administrative burden and load on the certificate-signing service. Note that in many cases, the user will have to be online for both the software updates and the new certificate generation; special consideration should be paid if organizations have users who operate in an offline mode for long periods of time and may not be able to receive an upgraded certificate before the cutoff date.
5) The organization can set policies for what capabilities different users have in terms of signing, encrypting, being able to send and receive messages from other recipients, and algorithm selection. The organization would further have to make decisions on selection of certificates and determination of behaviour for verification.
6) As a special case of the previous point, the organization would have to make decisions on how to handle special cases of verification which did not exist before. For example, if the signature from a classical certificate verifies but the PQC signature fails (or vice-versa), should the email message be accepted or rejected? If an email is received that is only signed with a classical algorithm, but the recipient knows from the directory that the sender has a PQ certificate, is that a downgrade / stripping attack, or simply that the sender has multiple email clients that have not been upgraded at the same time? Should the sender's email client downgrade itself to use classic-only encryption versus failing to send be dependent on

external factors such as whether it was able to reach the directory to check for an upgraded version of the recipient's certificate?

7) The organization must also decide on how the first PQC certificate request is generated. The user would have only a classical, quantum-vulnerable certificate available to generate the CSR. The organization must decide if this is acceptable to use or if the PQC certificate should be generated using a different method such as passphrase-based or through an adaptation of the original user onboarding protocol where no inherent certificate exists. Perhaps a sunset date needs to be established after which classical certificates can no longer be used to authenticate PQC certificate requests.

8) If it is deemed that classical certificates are acceptable to use in verifying PQC CSRs, then the new PQC CA would have to have the ability to verify certificate requests under the old CA. This is something that may need to be turned off at some point during deprecation.

9) At some point the organization may also need to decide on a forced key update for all of its users, especially if users have long-lifetime certificates and therefore will not naturally initiate a certificate renewal within the desired timeframe. Internal users can be forced to migrate to PQC as a part of organizational strategy. External users might need to have their emails rejected in order to incentivize them to migrate.

10) The archiving of old emails will be a particularly difficult issue to solve. It is often necessary to have these emails available for specified retention periods which can be years in length. Unfortunately, this would mean that they would be vulnerable to a quantum attack if attackers gained access to the archive, which posses a different risk profile compared with harvesting of emails as they are exchanged over the network. The organization must first decide if the risk of having these emails retained is acceptable. If not, they would have to decide how they will handle this situation. It would be possible to migrate all old emails to encryption under the email's recipient's new PQC certificate, but this would not be straightforward. The old emails might exist in any number of places, and it might not be possible to find, re-encrypt, and delete the original copies of all of them. We note that re-encryption is a data migration and inherently complex.

11) The point above regarding the archive of emails as originally encrypted requires older private keys for decryption to be maintained. This implies that some sort of key escrow would be in place. Having these keys in service represents a risk to the organization. The organization must decide to how to handle all old and escrowed keys.

12) When certificates are being migrated to PQC, a natural question to ask is what this means for digital identities. It is unclear how these will be impacted.

J.2.10 Use Case 10: SAML and Other Federated Identity Services

J.2.10.1 Description

Security Assertion Markup Language (SAML) is a standardized set of protocol messages based on XML syntax which enables authorization of a user to access a particular service. It inherently verifies identity, authenticates users, and authorizes services. Depending on the version of SAML, some inherent elements might be encrypted.

SAML relies on three major parties:

- **User** – the entity requesting access to a service;
- **Identity Provider (IdP)** – the entity which verifies the identity of the user;
- **Service Provider (SP)** – the entity providing the service.

The main asset is a SAML token provided by the IdP which is verified by the SP to allow the User to access the SP's resource.

Please note that the method of verification to authenticate the User to the IdP is beyond the scope of this Annex. In addition, the cryptographic security of transmissions between entities in SAML is also out of scope for this use case. This is most often provided by a standard protocol and so will fall into the use case of that particular protocol (e.g., TLS as described in use cases 4 and 5, Sections J.2.4 and J.2.5).

An alternative SAML access flow can occur when the User performs an initial login to the IdP to get a SAML token and then is free to use the token with any SP. An example would be a User logging into their computer at the beginning of a workday and then accessing services throughout the day via Single Sign On (SSO).

SAML also has several different methods of flow. They are:

- **Bearer** – the presence of any valid SAML token will grant access to the resource;
- **Holder of the Key** – Like Bearer, but the SAML token is bound to the User and the User must verify to the SP that they are the entity identified in the SAML token; and
- **Sender Vouches** – Like Bearer, but there is an additional entity called an Intermediary which handles all processing on behalf of the User and additionally signs messages to the SP.

Holder-of-the-key is not very common, but will still be included. The details of this mode of operation would not have a significant impact on migration.

Other frameworks used to provide identity authentication, such as OpenID Connect [62], a commonly used extension of the OAuth 2.0 authorization standard, have certain differences but follow a similar theme. Therefore, many of the activities for these frameworks would be similar.

Entrust offers an Identity as a Service (IDaaS) product which offers a variety of ID services. We will focus on the SAML part of the product, although similar activities and considerations will apply to their other services. In particular, the Entrust IDaaS will hold the role of a the IdP, the consumer's staff will take the role of Users, and there will be various resources, both internal and external, which will act as SPs.

SAML makes use of access control mechanisms to authenticate its users. As with use case 8 (viz., Vault Encryption, described in Section J.2.8), we will leave this out of scope for this case.

In order to establish federation, both users and SPs must register with the IdP. This means that there is an inherent inventory listing out all participants in this use case.

Finally, we note that while SAML is governed by OASIS, it depends upon the XML standard which is governed by the World Wide Web Consortium (W3C). There will likely need to be changes made to XML before any changes to the SAML protocol.

J.2.10.2 Assumptions

1) The IdP will have capability to generate its own signing certificate. It may leverage an Entrust CA if it wishes, but this will not affect the functionality. When holder-of-the-key is used, the same will be true of Users.
2) The technology will be backward compatible in the sense that it will still allow for use of classical cryptography when it is installed.
3) There is an existing registration and decommissioning process for both Users and SPs. For SPs, this is often a manual process accomplished by sending a SAML Assertion from the IdP. This will not fundamentally change when migrated to PQC.
4) PQC-enabled tokens will automatically be issued when a PQC-enabled User requests a token for a PQC-enabled SP. There will be no need for action to enable this capability.
5) It is still uncertain if the old IdP will be extended or if a new one would be created. In either case, the IdP will issue a PQC token if both User and SP are registered for PQC. It will issue a classical token otherwise.
6) Some SPs may actually be a cluster of devices functioning as a single SP called a SAML Gateway. Note that migration could only occur after all devices have had the new software installed.

[62] https://openid.net/developers/specs/

7) Users, after they have been migrated, will be able to accept both classical and PQC tokens. The token would depend on the relying party.
8) Non-registered Users will not receive tokens. Non-registered SPs will not accept tokens.

J.2.10.3 Migration Activities

The migration to PQC for this use case can be modelled through the diagram on the next page. The corresponding individual migration steps would be as follows:

1) Standards and Direction:

World Wide Web Consortium (W3C)

- Update the XML encryption and signature syntax to enable PQC in the general markup language.

Organization for the Advancement of Structured Information Standards (OASIS):

- Update the SAML protocol to enable PQC in its token signatures.[63, 64]

2) Planning and Decisioning:

Technology Provider (viz., Entrust):

- Determine a set of options available in terms of algorithms supported for both signing and verifying.
- Decide on whether or not an existing IdP can be extended to leverage PQC or a net new IdP would have to be created by consumers.
- Determine a rollout strategy for migrating Users. This could be done all at once or in batches.
- Determine requirements for SPs to leverage this service.
- Decide how to handle SPs who cannot migrate or lag behind anticipated migration timelines.

Consumer:

- Ensure inventory of Users and SPs is available.
- Select from the options for signing algorithm provided by Entrust for its IdP.
- Decide the appropriate time and conditions to decommission the old classical capabilities.

---

[63] Security Assertion Markup Language (SAML) v2.0, https://www.oasis-open.org/standard/saml/

[64] Post-quantum XML and SAML Single Sign-On, https://eprint.iacr.org/2024/828.pdf

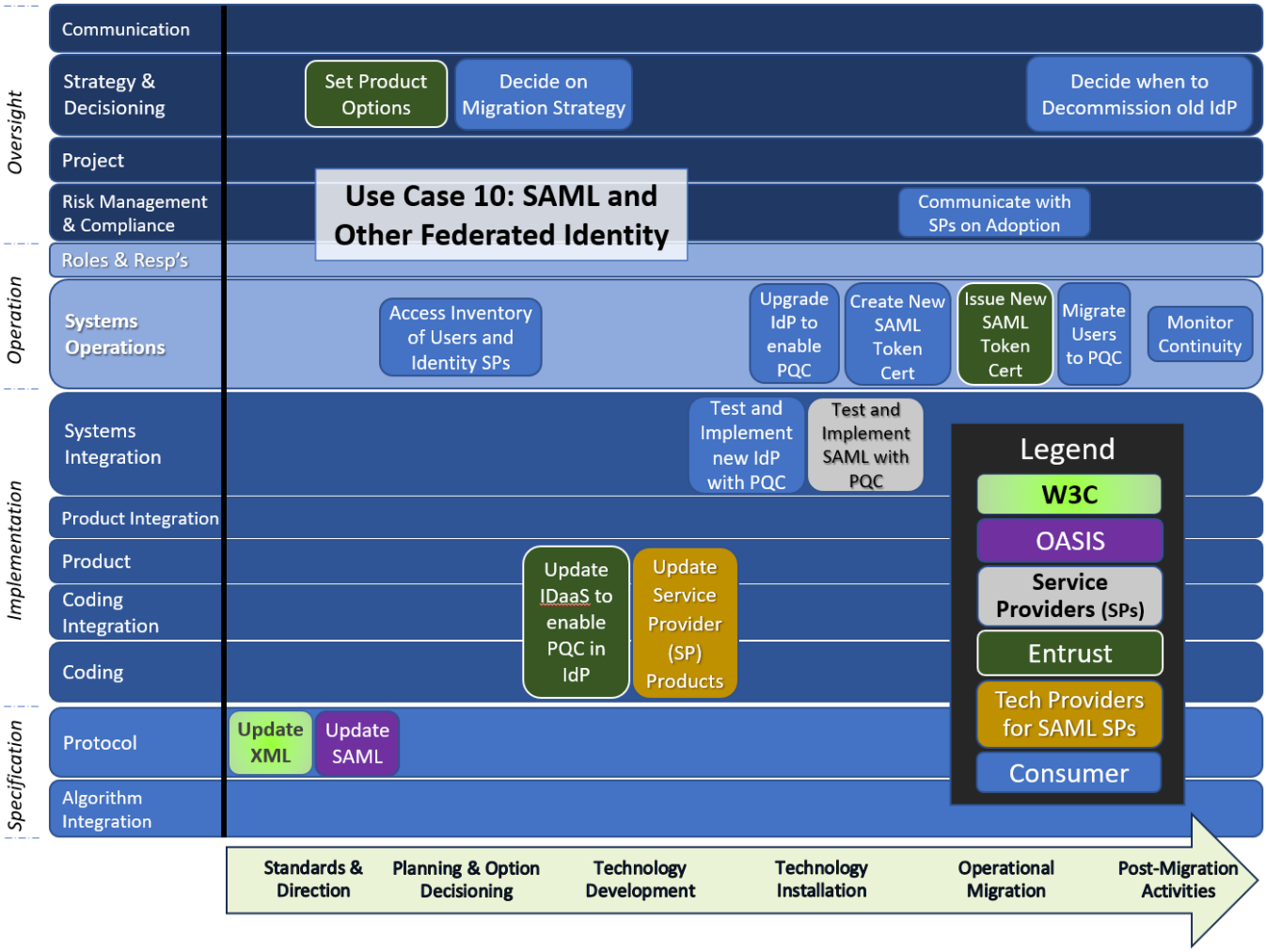
3) Technology Development:

Technology Provider (viz., Entrust):

- Develop a new version of IDaaS to enable PQC in the IdP service.
- Develop a new version of User software to support IDaaS for Users including holder-of-the-key.

Other Technology Providers (of SAML SPs):

- Code new versions of SAML capabilities as well as additional requirements from Entrust for Service Providers (SPs) into their products.



4) Technology Installation:

Consumer:

- Test and implement new version of IdP leveraging PQC.
- Test, implement and roll out new version of user software to all Users.

Identity Service Providers:

- Update products to a new version of SP allowing PQC.

5) Operational Migration:

Consumer:

- Create new, or update existing, IdP to enable PQC capabilities.
- Create and then install new SAML token certificate issued by Entrust.
- Migrate each User to PQC, creating a new public key if holder-of-the-key is used.
- Re-register all existing SPs to PQC capabilities.

Identity Service Providers:

- Re-register with the IdP.

6) Post-Migration Activities:

Consumer:

- Monitor continuity. Prepare to backout or take other action if outages occur.
- Decommission old IdP or remove its classical capability when all entities are migrated and sufficient confidence is obtained in resilience of PQC.

J.2.10.4 Backward-Compatibility Considerations

The following are considerations:

- The important operation is when the IdP begins issuing PQC SAML tokens. It is unclear if SPs will be able to accept these tokens. This may cause outages.
- Tokens from the old IdP would naturally expire. The halting of token issuance from the old IdP would set a natural timeline for when no such token would be valid.

J.2.10.5 Potential Downgrade Attacks

Since the IdP will issue tokens based on registration, the possibility of downgrade attacks is minimal.

J.2.10.6 Summary of Key Issues or Things You May Not Have Initially Thought Of

1) The requirement for registration with IdP would inherently give an inventory of all Users and SPs. This will simplify migration and tracking and protect against downgrade attacks.
2) The so-called Golden SAML attack is ever-present when dealing with SAML-based systems including those implementing PQC. [65] We note here that leveraging a quantum computer to

[65] https://www.cyberark.com/resources/threat-research-blog/golden-saml-newly-discovered-attack-technique-forges-authentication-to-cloud-apps

compromise the IdP's private key would enable issuance of a Golden SAML token on the classical system. In this respect, the PQC migration would mitigate the quantum attack and, hence, help prevent a Golden SAML attack on the classical system.

3) The non-persistence of tokens makes the operational migration much simpler.

J.2.11 Use Case 11: IPsec and IKE

J.2.11.1 Description

This use case will cover IP-layer security through the IPsec suite of protocols[66, 67], with both Authentication Header (AH) and Encapsulated Security Payload (ESP). It will also cover the Internet Key Exchange (IKE) protocol which is used to establish the symmetric keys used for IPsec.

This use case is, in many ways, similar to use case 5 (viz., Non-Browser Based TLS Connections) as described in Section J.2.5. In particular, migration would consist of two phases which can be done separately:

- Cipher Suite Migration for Key Establishment
- Cipher Migration for Authentication

The particulars of these two use cases are described in the two sub-use cases below.

J.2.11a Sub-Use Case 11a: (IKE) Cipher Suite Migration for Key Establishment

J.2.11a.1 Description of this Sub-Use Case

This sub-use case covers the cipher suites used for encryption, authentication, and integrity within IP packets. This establishment is generally handled through the use of the Internet Key Exchange (IKE) protocol where a Diffie-Hellman group is selected and ephemeral keys are exchanged. The basic idea is to perform a similar exchange using PQC. Migrating this portion of the Transport Layer Security (TLS) connection to quantum-safe ephemeral keys would protect against passive HNDL or SNDL attacks.[68]

---

[66] Security Protocol for the Internet Protocol (IPsec), IETF RFC 4301, December 2005

[67] IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap, IETF RFC 6077, February 2011

[68] Steal-Now-Decrypt-Later (SNDL) is a synonym for HNDL.

J.2.11a.2 Assumptions of this Sub-Use Case

1) IKE version 1 is deprecated as per IETF RFC 9395[69], so it is assumed systems currently use IKEv2 specified in IETF RFC 7296 and its updates.[70]

J.2.11a.3 Migration Activities

The migration to PQC for this sub-use case can be modelled through the diagram on the next page. The corresponding individual migration steps would be as follows:

1) Standards and Direction:

Internet Engineering Task Force (IETF):

- Update the Internet Key Exchange (IKE) protocol to support PQC for use as ephemeral keys.

Internet Assigned Numbers Authority (IANA):

- Register new code points for the new PQC algorithms.

2) Planning and Decisioning:

Consumer:

- Interact with Technology Provider(s) to ensure their products meet the requirements of the systems (e.g., latency and throughput using new PQC algorithms).

3) Technology Development:

Technology Providers:

- Code and enable the new PQC algorithms in their products according to specifications.

4) Technology Installation:

Consumer:

- Understand and test the changes to the requisite products including backward compatibility.
- Implement new technology in production systems.

---

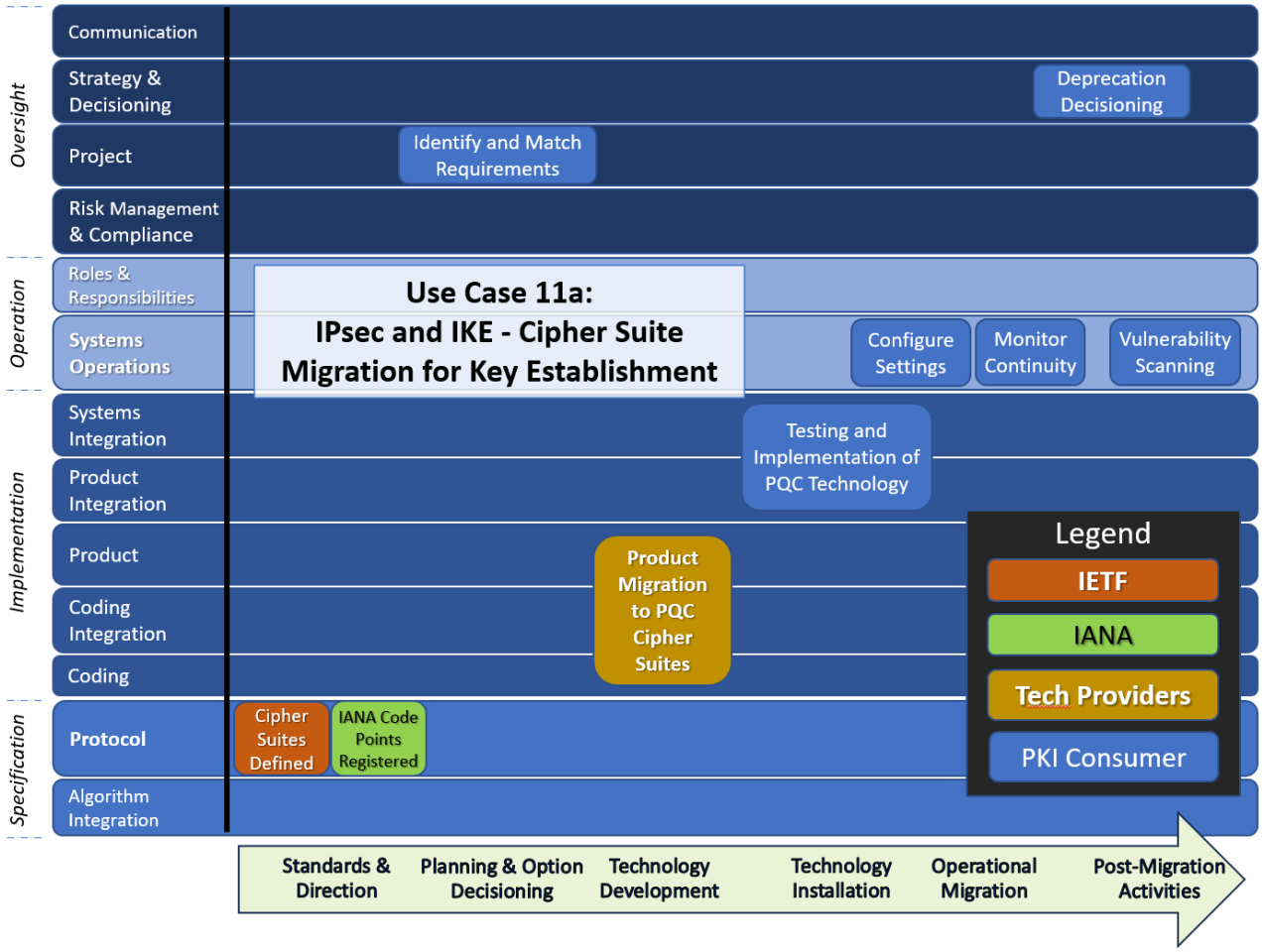
[69] Deprecation of the Internet Key Exchange Version 1 (IKEv1) Protocol and Obsoleted Algorithms, IETF RFC 9395, April 2023

[70] Internet Key Exchange Protocol Version 2 (IKEv2), IETF RFC 7296, October 2014

5) Operational Migration:

Consumer:

- Configure settings to ensure quantum-safe "**transforms**" are active and/or preferred. [71]
- Ensure consuming services enable compatible PQC transforms.



6) Post-Migration Activities:

Consumer:

- Monitor connections to ensure business continuity and back out if necessary.
- Test connections for quantum-safety and/or cryptographic agility.
- When given the go-ahead, deprecate the old non-quantum-safe transforms in the configuration.

---

[71] "A **transform** set is an acceptable combination of security protocols, algorithms and other settings to apply to IP Security protected traffic", per Solved: Transform sets vs. IKE policy attributes - Cisco Community, February 2015

- Perform vulnerability scanning on implementations to determine if the new transforms are being utilized.

J.2.11a.4 Backward-Compatibility Considerations

The following are considerations:

- This use case has the characteristic that backward compatibility is automatically maintained as a result of ephemeral-key and security-association negotiation. New connections may negotiate quantum-safe transforms, but can always fall back to existing or legacy transforms during the migration period.

J.2.114a.5 Potential Downgrade Attacks

As with TLS, the main risk in downgrade attacks during IKE involves convincing an entity that the other cannot handle PQC transforms. This would potentially apply to future downgrade attacks as well. This should be monitored across the industry.

J.2.11b Sub-Use Case 11b: (IKE) Cryptographic Migration for Authentication

J.2.11b.1 Description of this Sub-Use Case

This sub-use case will cover the implementation of quantum-safe certificates for authentication. This will prevent active attacks such as man-in-the-middle (MITM) or impersonation attacks due to decryption (by quantum computers) of the static keys within certificates.

J.2.11b.2 Assumptions of this Sub-Use Case

1) The latest version of IKE will allow processing of either classical or PQC-enabled certificates. All parties will work under this assumption.

J.2.11b.3 Migration Activities

The migration to PQC for this sub-use case can be modelled through the diagram on the next page. The corresponding individual migration steps would be as follows:

1) Standards and Direction:

National Institute of Standards and Technology (NIST):

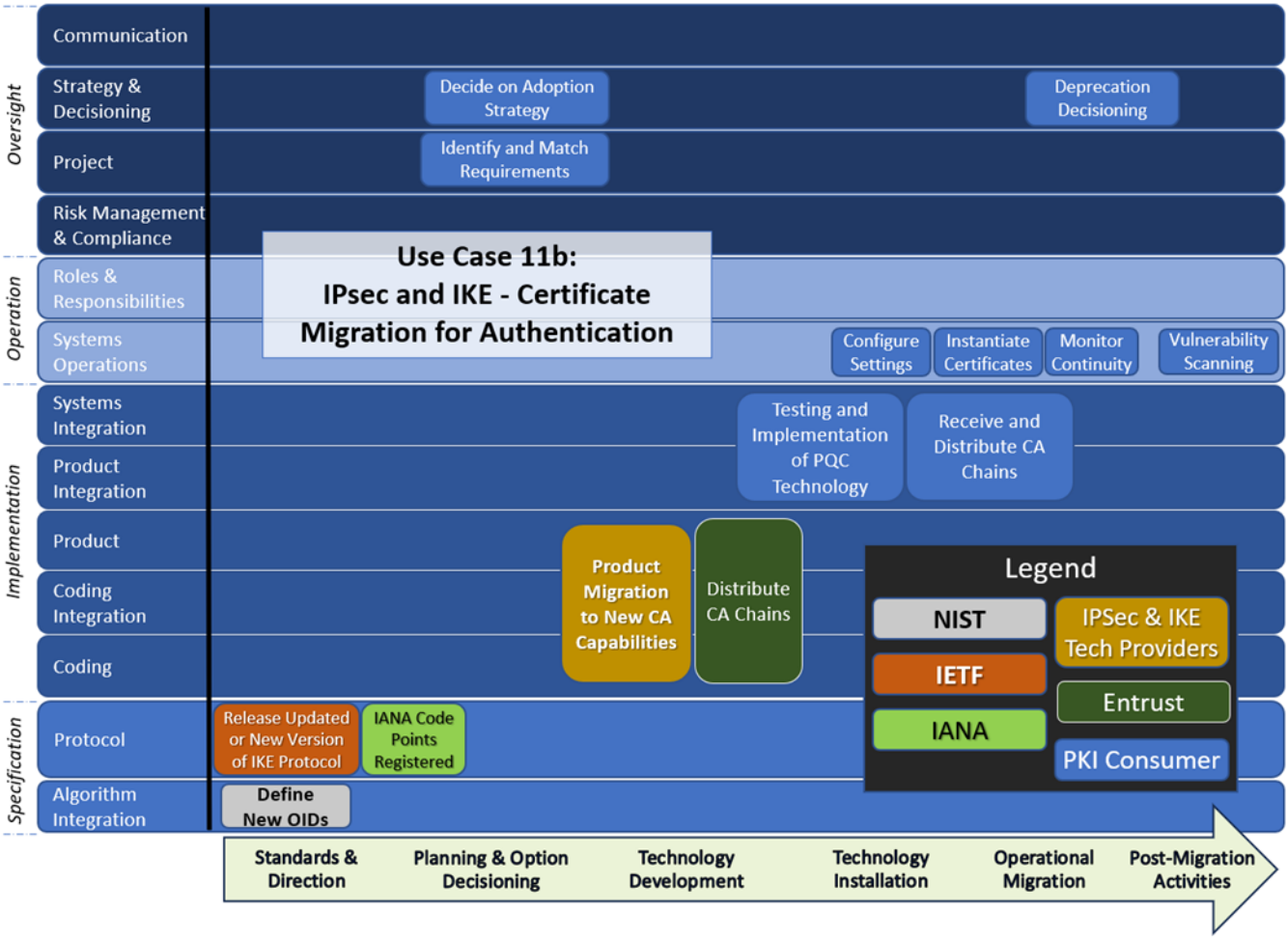
- Define Object Identifiers (OIDs) for new Post-Quantum Cryptography (PQC) algorithms as well as the binary encodings of public keys and signatures.

Internet Assigned Numbers Authority (IANA):

- Register new code points for the new PQC algorithms.

Internet Engineering Task Force (IETF):

- Release an update to IKEv2 which can incorporate new algorithms and methods to implement PQC as recommended by NIST. This will include the encoding of the public key and signature. There is also a possibility that payload notifications will allow peers to announce supported authentication mechanisms. [72]
- The LAMPS working group of the IETF will need to update PKIX certificate standards to include new OIDs, to be defined by NIST, for each PQC algorithm standardized by NIST (e.g., FIPS 204 ML-DSA[73], FIPS 205 SLH-DSA[74]).



[72] Announcing Supported Authentication Methods in IKEv2, draft-ietf-ipsecme-ikev2-auth-announce-10, April 2024

[73] Internet X.509 Public Key Infrastructure: Algorithm Identifiers for ML-DSA, draft-ietf-lamps-dilithium-certificates-04, July 2024

[74] Use of the SLH-DSA Signature Algorithm in the Cryptographic Message Syntax (CMS), draft-ietf-lamps-cms-sphincs-plus-07, July 2024

2) Planning and Decisioning:

Technology Provider (viz., Entrust):

- Distribute new CA chains to all entities.

Consumer:

- Interact with Technology Provider(s) to ensure their products meet the requirements of the systems (e.g., latency and throughput using new PQC algorithms).
- Decide on the adoption strategy for the enterprise.

3) Technology Development:

Technology Providers:

- Code and enable the new PQC algorithms in their products according to specifications.
- Ensure that there is support for both classical and quantum-safe CAs to work simultaneously.

4) Technology Installation:

Consumer:

- Understand and test the changes to the requisite products including backward compatibility.

5) Operational Migration:

Consumer:

- Receive and distribute new CA chains from Entrust.
- Deploy certificates and implement new technology in production.
- Configure settings to ensure quantum-safe certificates are active and/or preferred

6) Post-Migration Activities:

Consumer:

- Monitor connections to ensure business continuity and back out if necessary.
- Test connections for quantum safety and/or cryptographic agility.
- Perform vulnerability scanning on implementations to determine the certificates being utilized.
- When given the go-ahead, deprecate the old non-quantum-safe certificates in the configuration.

J.2.11b.4 Backward-Compatibility Considerations

The following are considerations:

- This use case has the characteristic that backward compatibility is automatically maintained as long as both classical and PQC certificates can be used simultaneously. It is assumed that new connections may negotiate quantum-safe certificates, but can always fall back to existing certificates during the migration period.

J.2.11b.5 Potential Downgrade Attacks

The main risk here lies in the assumption that both classical and PQC certificates will be simultaneously allowed in IKE. The risk is that an attacker may somehow force the use of the classical certificate when a PQC certificate is available.

J.2.11.6 Summary of Key Issues or Things You May Not Have Initially Thought Of

1) Migrating cipher suites used to establish keys (described in Section J.2.11a) can be performed independently of migrating the cryptography used for authentication (described in Section J.2.11.b).
2) Migrating security associations should be the easier of the two sub-use cases and migrating that portion of IKE connections to quantum-safe ephemeral keys would mitigate the risk of HNDL or SNDL attacks.
3) Mitigating the certificates used for authentication could be done later, to mitigate the risk of authentication attacks when a quantum computer is available.

J.2.12 Use Case 12: FIDO2 and Other User Authentication Methods

J.2.12.1 Description

Several protocols such as KMIP (described in Section J.2.8) and SAML (described in Section J.2.10) require a user to personally authenticate to a server or resource. While this has traditionally been handled by userid and password, there are now options to move to password-less technologies, some of which employ asymmetric cryptography and so are vulnerable to quantum attacks.

FIDO2 is one such protocol.[75] It involves registering a user's public key with the entity performing the authentication and then using signed messages to verify the user. We note that this public key is unique to each user / authenticating entity pair, although it may exist across

[75] The FIDO2 specifications are the World Wide Web Consortium's (W3C) Web Authentication (WebAuthn) specification and the FIDO Alliance's corresponding Client-to-Authenticator Protocol (CTAP); https://fidoalliance.org/fido2/, accessed April 8, 2024.

many user devices. As other protocols based on public keys would have a similar methodology, we will focus on FIDO2. We note that Entrust leverages FIDO2 as part of their IDaaS service.

There are elements of FIDO2 and other authentication methods apart from public keys. We will consider these out of scope for this use case as they are not affected by quantum.

We also note that other protocols such as Transport Layer Security (TLS) may be used in some user authentication methods and that these protocols would therefore need to be made quantum-safe as well. However, this will be out of scope as it falls under a different use case.

J.2.12.2 Assumptions

1) An inventory of users for each resource is available for each authenticating entity.
1) There is an existing protocol to register new users to FIDO2 and to rotate public keys. Note that this might entail sending a new physical USB stick or similar object to the user.

J.2.12.3 Migration Activities

The migration to PQC for this use case can be modelled through the diagram on the next page. The corresponding individual migration steps would be as follows:

1) Standards and Direction:

World Wide Web Consortium (W3C) and FIDO Alliance:

- Update the WebAuthn web standard and the FIDO2 open standard to support PQC.

Internet Assigned Numbers Authority (IANA):

- Register new code points in the JSON Web Signature (JWS)[76] registries for the PQC algorithms.

Internet Engineering Task Force (IETF):

- Update JWS and JSON Web Token (JWT)[77] specifications to support PQC.

2) Planning and Decisioning:

Technology Provider (viz., Entrust):

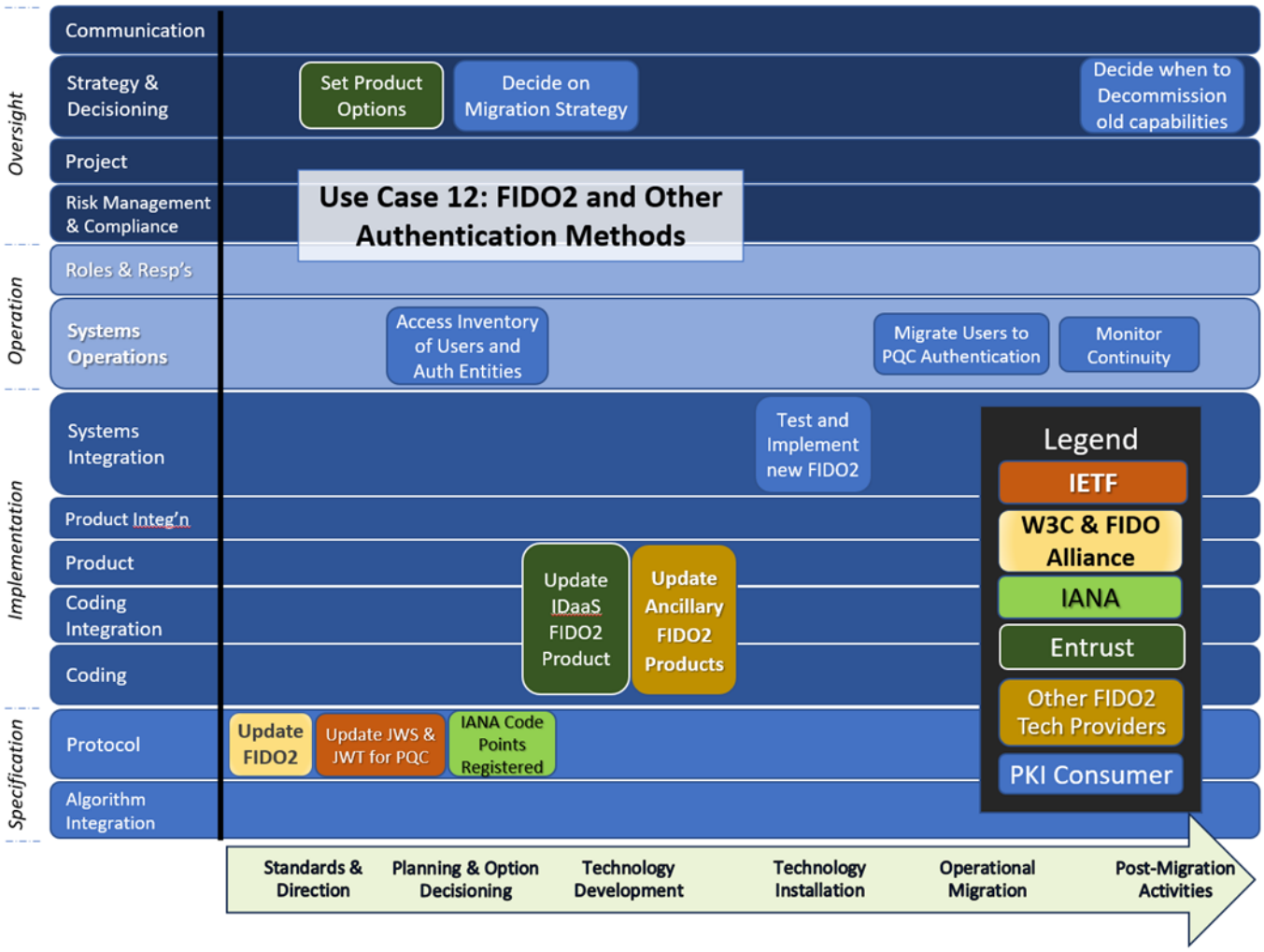
- Determine a set of options available in terms of algorithms supported in the FIDO2 protocol.

[76] JSON Web Signature (JWS), IETF RFC 7515, May 2015

[77] JSON Web Token (JWT), IETF RFC 7519, May 2015

Consumer:

- Ensure inventory of users is available.
- Select from the options for signing algorithms offered by Entrust.
- Decide on a migration strategy for users.
- Decide the appropriate time and conditions to decommission the old classical capabilities.



3) Technology Development:

Technology Provider (viz., Entrust):

- Develop new version of IDaaS to enable PQC for use with the FIDO2 protocol.

Other Technology Providers:

- Code new versions of FIDO2 capabilities into any user products (e.g., USB keys, thick clients, mobile device passkeys).

4) Technology Installation:

Consumer:

- Test and implement new version of FIDO2 leveraging PQC on its resources.
- Test, implement and roll out new version of user software to all users.

5) Operational Migration:

Consumer:

- Create new PQC public/private keys for all FIDO2 users and re-register PQC public keys to authenticating entities using existing processes.
- Synchronize public/private keys between user devices using existing processes.

6) Post-Migration Activities:

Consumer:

- Monitor continuity. Prepare to backout or take other action if outages occur.
- Track migration of users.
- After a certain point in time, force remaining users to migrate.
- Deprecate old classical authentication.

J.2.12.4 Backward-Compatibility Considerations

The following are considerations:

- Since the relationship between user and authenticating entity is unique and pairwise, once a new public key is registered, all authentication will occur using that public key going forward.
- Since authentication is non-persistent, there are no backward- compatibility issues.

J.2.12.5 Potential Downgrade Attacks

Once a new public key is registered, the old one would become deprecated. The only downgrade attacks possible would be if the authenticating entity still allows users to register using classical cryptography.

J.2.12.6 Summary of Key Issues or Things You May Not Have Initially Thought Of

1) The fact that the relationship between user and authenticating entity is pairwise, unique, and non-persistent greatly simplifies the transition and minimizes the risk of downgrade attack.
2) Migrating users can occur all-at-once, in batches, or gradually, based on some pre-existing condition. It is up to the organization to decide on a strategy.
3) One must pay attention to synchronization of the user's public/private key pair on different devices. There may be cases where one of their devices has not migrated. There would need to be a way to recognize this and/or deal with this if it happens.

J.2.13 Use Case 13: Mobile Device Management

J.2.13.1 Description

Mobile Device Management (MDM) deals with the management of applications and configurations on a user's mobile device. It is responsible for installing, updating, and removing applications (apps) on the device according to an organization's policies.

The apps on user's devices may or may not need to be migrated to PQC. However, this migration will be out of scope as we will assume that each application and its corresponding ecosystem will have their own migration plan. The MDM itself may also perform some cryptographic operations such as cryptographically signing instructions to the device. However, this is vendor dependent and its migration will also be out of scope for this use case.

Instead, this use case will solely be concerned with the update to the MDM itself as well as its actions which put PQC technology in place for other apps. As such, this use case is fundamentally different from all of the other use cases in this Annex. It will not concentrate on migrating cryptography, but instead will focus on leveraging the MDM to migrate other applications to use PQC technology.

Part of this migration will involve obtaining certificates from a Private Certification Authority (CA). The management of PQC certificates through a Private CA is described in use case 2 (in Section J.2.2) and so is out of scope for this use case. The interface to the CA is in scope.

We note here that the secure connection between the MDM and its base station will likely need to be migrated to be quantum-safe. However, this will just be considered another app and so is also out of scope for this use case.

J.2.13.2 Assumptions

1) The organization already has a mechanism in place to provision, update, or remove the MDM from a mobile device. This mechanism can be leveraged to make any updates to the device required for PQC migration.
2) Updates of the Operating System (OS) and MDM will still allow all old applications to function as normal for backward-compatibility purposes.
3) The organization may issue its own devices or implement Bring-Your-Own-Device (BYOD), or both, as part of its MDM strategy. For the purpose of this use case, there is no material difference between the two.
4) There is already a process in place for the MDM to update the certificate chains, individual certificates, and other forms of cryptography for the apps on the device. This mechanism can be leveraged to make updates with regards to PQC certificates. The parameters for these certificates might be app-dependent and so are out of the scope of this use case. The connection to the Private CA to obtain the certificates is in scope.
5) Any cryptographic implementation on the device which is not managed by the MDM will not leverage the MDM for its migration and so will be out of scope for this use case.
6) The MDM will be dependent upon the OS which the device uses. There is an existing process for the OS migration to enable PQC technology, if required.

J.2.13.3 Migration Activities

The migration to PQC for this use case can be modelled through the diagram below. The corresponding individual migration steps would be as follows:

1) Standards and Direction:

Internet Engineering Task Force (IETF):

- Update certificate request protocols (e.g., SCEP [78]) with regards to PQC on mobile devices.

2) Planning and Decisioning:

Consumer:

- Develop a new MDM policy which incorporates the aspects of PQC for the various apps to be supported on mobile devices.
- Develop a rollout strategy for updating devices such as in batches, or at time of re-issue.
- Develop a strategy for the migration of individual apps to PQC.

[78] Simple Certificate Enrolment Protocol (SCEP), IETF RFC 8894, September 2020

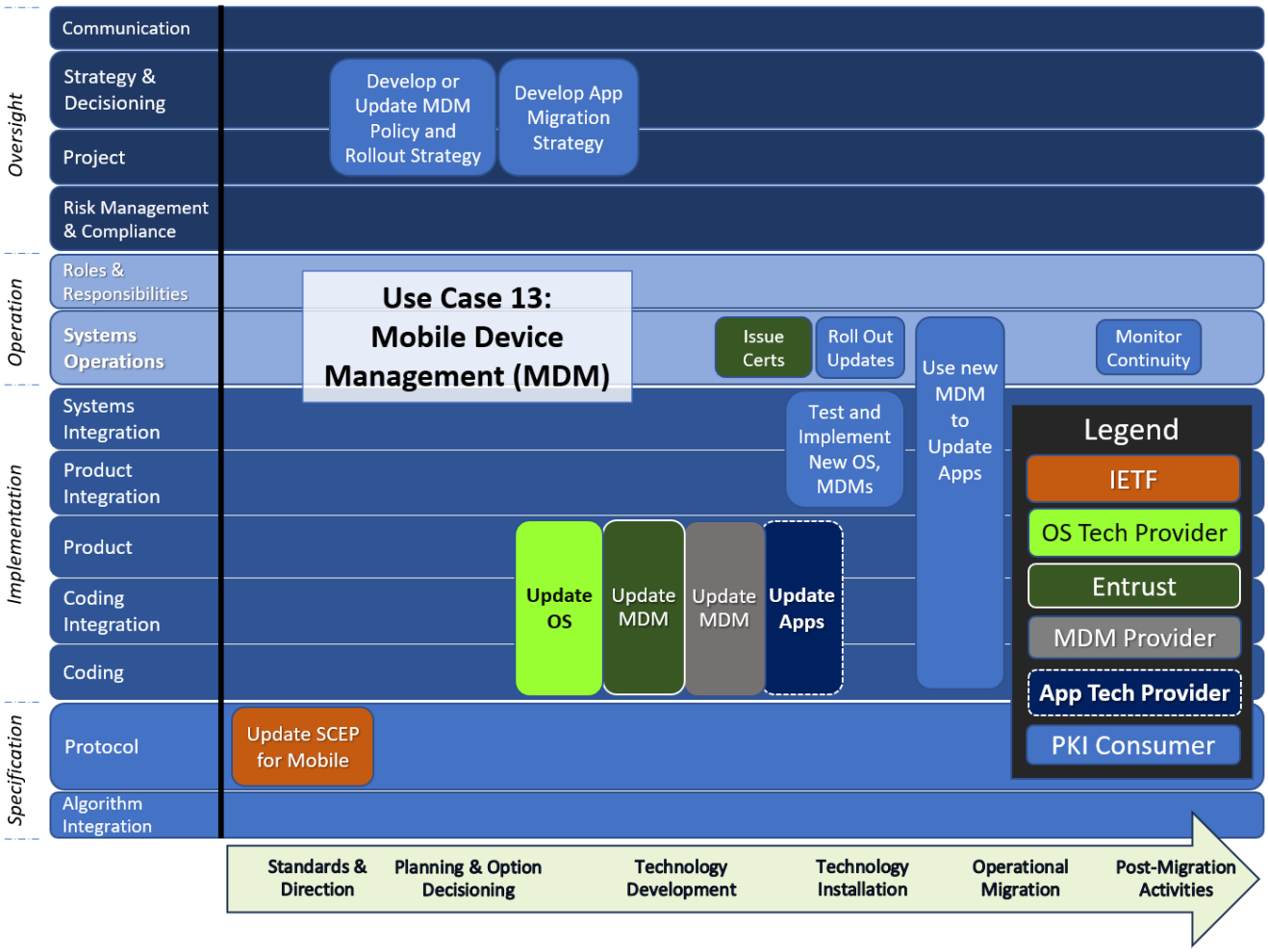
3) Technology Development:

OS Technology Provider:

- Develop new version of the Operating System which will enable PQC technology for MDM.

App Technology Providers:

- Develop a new version of the app ecosystem which will leverage PQC technology.



MDM Provider:

- Develop a new version of the MDM solution which can facilitate PQC technology for certificates and other cryptographic implementations on the Operating System, the certificate request protocol, and the apps on the device.

PKI Technology Provider (viz., Entrust):

- Update the interface to the Private CA and certificate request protocol implementations to allow the MDM to obtain PQC certificates.

4) Technology Installation:

Consumer:

- Test and implement new version of OS for mobile devices.
- Test, implement and roll out new version of the MDM including using the existing process. The implementation will include the new version of the certificate request protocol to be used by the MDM.
- Test and roll out an update to the CA, if applicable, to enable the MDM to request and receive certificates.

5) Operational Migration:

PKI Technology Provider (viz., Entrust):

- Issue the requisite certificates for use by the MDM.

Consumer:

- Make updates to shared device locations such as certificate chains.
- Obtain the necessary certificates for MDM use.
- Execute migration strategies for individual or collective applications in the ecosystem according to the organizational strategy. This would include, but not be limited to, the provisioning of PQC certificates for apps using the certificate request protocol.

6) Post-Migration Activities:

Consumer:

- Track and monitor device migration as well as app migration.
- After a certain point in time, execute a forced update for devices which have still not been migrated to PQC.

J.2.13.4 Backward-Compatibility Considerations

The following are considerations:

- The OS and MDM can be developed so that installation will not affect existing apps and MDM functionalities prior to any PQC migration. Thus, there should be no backward-compatibility issues with installation.
- Placing certificate chains, certificates, or other common cryptographic objects which are PQC-enabled in shared locations on the device theoretically should not cause any issues. However, there is always the possibility that it may cause an issue with a particular application. While this should be tested, the issue is app-dependent.
- The migration of apps may have issues with backward compatibility. However, these are dependent on the app.

J.2.13.5 Potential Downgrade Attacks

To be determined.

J.2.13.6 Summary of Key Issues or Things You May Not Have Initially Thought Of

1) As the MDM is merely a facilitator of PQC technology in this use case, its installation should be straightforward. The real work will occur when the app ecosystem is migrated to PQC.
2) The MDM will likely be an important tool in the migration of mobile-device-related use cases. Not only is it a tool which can facilitate the migration, but it will also likely have access to many details of the devices, the apps on it, and possibly the associated certificates, all of which will help facilitate the strategy and migration.

J.3 Glossary of Terms Used in this Annex

Term	Definition or Description
ACME	Automated Certificate Management Environment
API	Application Programming Interface
Application Developer	A software developer who does not have specialist knowledge in cryptography in general nor post-quantum cryptography in particular
CA	Certification Authority - a trusted entity that issues and revokes public key certificates
CA/Browser Forum	Certification Authority Browser Forum
Certificate	A set of data that uniquely identifies a public key (which has a corresponding private key) and an owner that is authorized to use the key pair
CMC	Certificate Management over CMS (CMC) is a network protocol for managing certificates and is defined in IETF RFC 5272
CMP	The Certificate Management Protocol (CMP) is a network protocol for managing certificates, and is defined in IETF RFC 4210
Code Signer/Verifier	A code signer is an entity which signs a software package in order to prove its authenticity and integrity; typically this will be the vendor or distributor of that software. A verifier is an entity that verifies the signature on a software package prior to installing and running the software; typically this is an OS.
Consumer, Consumers	System owners and operators, including various different groups within their organizations, who use or rely on PKI in some way.
CP	Certificate Policy - A named set of rules that indicate the applicability of a certificate to a particular community and/or class of applications with common security requirements

Term	Definition or Description
CPS	Certificate Practice Statement - A statement of the practices that a Certification Authority employs in issuing and managing public key certificates
CRL	Certificate Revocation List
Cross-signed root	Cross-certification is the process of issuing a cross-certificate, which is a certificate issued by one CA to another CA that contains a CA signature key used for issuing certificates. Ref: https://datatracker.ietf.org/doc/html/rfc5280 Section 3.5
Crypto Agility	Cryptographic agility (sometimes abbreviated as Crypto agility) is the ability to easily implement, update, and replace cryptographic components within Information Technology (IT) systems, without affecting their functionality, with no significant changes to the infrastructure, and without disruptions to running systems
Cryptographic Algorithm	Well-defined procedure or sequence of rules or steps, or a series of mathematical equations used to describe cryptographic processes such as encryption/decryption, key generation, authentication, signatures, etc.
CSR	Certificate Signing Request
DoS	Denial of Service
Ephemeral key	A cryptographic key is called ephemeral if it is generated for each execution of a key establishment process.
EST	Enrollment over Secure Transport (EST) is a network protocol for managing certificates and is is defined in https://datatracker.ietf.org/doc/html/rfc7030
FIDO2	FIDO2 enables users to leverage common devices to easily authenticate to online services in both mobile and desktop environments
FIPS	Federal Information Processing Standards
HNDL	Harvest Now, Decrypt Later

Term	Definition or Description
HSM	Hardware Security Module
Hybrid	One of several types of cryptographic techniques that combines two or more cryptographic algorithms for increased security. Within the context of this document, a hybrid refers to a post-quantum / traditional hybrid
IANA	Internet Assigned Numbers Authority
IDaaS	Identity as a Service is cloud-based authentication built and operated by a third-party provider
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange protocol
JSON	JavaScript Object Notation is a lightweight data-interchange format
JWS	JSON Web Signature
JWT	JSON Web Token
KEM	Key Encapsulation Mechanism
Key Escrow	An arrangement in which keys needed to decrypt encrypted data are held in escrow so that, under certain circumstances, an authorized third party may gain access to those keys
KMIP	Key Management Identity Protocol specified by OASIS
Linter	A linter will scan source code looking for errors, defects, stylistic issues, and questionable constructs. The term "linter" stems from the origins of a tool known as "lint," which was initially developed by Stephen C. Johnson in 1978 at Bell Labs.
MDM	Mobile Device Management
MMC	The Microsoft Management Console is used to create, save and open administrative tools (called consoles) which manage the hardware, software, and network components of Microsoft's Windows operating system.

Term	Definition or Description
OASIS	Organization for the Advancement of Structured Information Standards
OCSP	Online Certificate Status Protocol
OID	Object Identifier
OS	Operating System
OS/Browser Community	The community software venders who produce operating systems and web browsers, which often act as consumers of cryptography and PKI
Perfect forward secrecy	Perfect Forward Secrecy (PFS), also called Forward Secrecy (FS), refers to an encryption system that changes the keys used to encrypt and decrypt information frequently and automatically. This ongoing process ensures that even if the most recent key is hacked, a minimal amount of sensitive data is exposed.
PKC	Cryptography that uses separate keys for encryption and decryption; also known as asymmetric cryptography
PKCS	Public Key Cryptography Standards
PKI	Public Key Infrastructure (PKI) defines the foundation for most modern security systems and provides Confidentiality, Integrity, Authenticity and Non-repudiation. It relies on public key cryptography, a universally accepted asymmetric technology that enables entities to securely communicate using an insecure transport or media, reliably link the data to the signatory and protects the integrity of the data while proving guarantees about the existence of the data at the time of signature creation
PKIX	Public-Key Infrastructure (X.509) standards developed by the IETF
Post-Quantum Algorithm	An asymmetric cryptographic algorithm that is intended to be secure against attacks using quantum computers as well as classical computers

Term	Definition or Description
Post-Quantum Cryptography (PQC)	New cryptography that uses quantum-resistant primitives, with the goal of keeping existing public key infrastructure intact in a future era of quantum computing (i.e., to be secure against both quantum and classical computers) and to be deployable without drastic changes to existing communication protocols and networks
PQC Developer	A software developer who has specialist knowledge in developing applications and cryptographic protocols using post-quantum cryptography
Public Key Cryptography Algorithm	An asymmetric cryptographic algorithm that uses two related keys: a public key and a private key. The two keys have the property that determining the private key from the public key is computationally infeasible for classical computers
QA	Quality Assurance
Quantum-vulnerable Cryptography	Cryptographic algorithms that may not be secure against quantum cryptographic attacks
Quantum-vulnerable Email	Email digitally signed and/or encrypted with quantum-vulnerable cryptography
Quantum-Safe Cryptography	New cryptographic algorithms (usually public-key algorithms) and/or technologies (such as Quantum Key Distribution) that are believed to be secure against cryptanalytic attacks by quantum computers.
RA	Registration Authority
RNG	Random Number Generator
RSA	Public-key algorithm developed by Rivest, Shamir, & Adleman that is used for key establishment and the generation and verification of digital signatures
RSA-SHA1	A digital signature scheme which uses both the RSA and SHA1 algorithms
RSA-SHA2	A digital signature scheme which uses both the RSA and SHA2 algorithms

Term	Definition or Description
SaaS	Software as a Service
SAML	Security Assertion Markup Language
SCEP	Simple Certificate Enrollment Protocol
SDLC	Software Development Life Cycle
Security Association	Security Associations (SAs) are security policies defined for communication between two or more entities. A set of algorithms and mutually agreed-upon keys are used and represented by both parties when attempting to establish a connection.
SHA-1	Secure Hashing Algorithm version 1
SHA-2	Secure Hashing Algorithm version 2
S/MIME	Secure/Multipurpose Internet Mail Extensions
TLS	Transport Layer Security protocols developed by the IETF. Also sometimes referred to by the older name "Secure Socket Layer (SSL)"
Transform sets	A transform set is an acceptable combination of security protocols, algorithms and other settings to apply to IP Security-protected traffic, During the IPSec security association negotiation, the peers agree to use a particular transform set when protecting a particular data flow.
TSA	Time Stamp Authority - an entity that is trusted to provide accurate time information
Vault	A KMIP server than can serve requests from external KMIP clients

APPENDIX A: QUANTUM-READINESS MYTHS AND FAQS

	Myth	Reality
1	The Quantum Threat applies only to a small set of organizations within Canada.	The Quantum Threat is of national significance and impact. The risks to information security as well as health and safety, across domains including Critical Infrastructure, 5G, Cloud, AI/ML, and IoT, will require actions at a national scale, and efforts and actions from both government and organizations.
2	The Quantum Threat: For my organization, it is an Information Technology (IT) problem.	For the Organization, the threats and risks posed by Quantum Computing are, first and foremost, a BUSINESS problem.
3	The Information and Communications Technology (ICT) sector and related industry organizations will solve this. My organization / sector don't have to do anything… or not much?	It is true that the vast array of quantum stakeholders, including standards organizations, ICT sector organizations, academia, and others are working diligently to try to address the threats posed by the future of quantum computing. However, at the end of the day, individual organizations and sectors are ultimately accountable for ensuring the confidentiality, integrity, and availability of all key data of value that is stored, processed, and transmitted.
4	This is not a pressing issue at this time. Getting prepared for Quantum … that can wait?	The process of Quantum Risk Assessment and Quantum Migration may take many years, if not even longer. The timelines for organizations and sectors will depend on many factors, including but not limited to: numbers, types, complexities, and interdependencies (intra-org and inter-org) of products, systems, interfaces, and solutions employing various cryptographic systems; trusted supply chain of cryptographic systems (hardware & software); Skilled resources' availability; etc.

	Myth	Reality
5	NIST is still in the process of standardizing Post-Quantum Cryptography. Should one wait until that is done, before starting to prepare for PQC?	From a planning perspective, there are no direct dependencies on the outcomes of NIST's Post-Quantum Cryptography Standardization process that should prevent or delay an organization or industry sector from planning how to mitigate the risks, posed by quantum computing, to their uses of cryptography. From an QSC migration perspective, the future implementations must be based on standardized PQC algorithms within certified technology products and solutions.
6	The risk is low within the organization / sector, because cryptography usage is low ?	Cryptography is pervasive and embedded across all aspects of Information and Communications Technology, to help ensure the confidentiality, Integrity of information that is stored, processed, and transmitted.
7	The confidentiality of current sensitive information is safe for now. Getting Quantum-Prepared can wait ?	One of the key threat scenarios is the capture of data today (including encrypted data as well as cryptographic information such as cryptographic key exchanges), and then decrypting the captured data in the future using quantum technologies.
8	Preparing for the impacts of Quantum technologies seems simple and straightforward for my organization / industry sector. So this preparation can wait, can't it?	That depends. Quantum readiness depends on many factors, including but not limited to: the quantities and types of valuable data to be protected from attacks; the length of time that valuable data must be kept confidential and intact; the number and types of systems that store, process and transmit the data; the number and complexity of interfaces to other systems; inter-organization dependencies and more. A Quantum-Readiness assessment should be undertaken to provide insight to the scope of the effort the organization should be prepared to expect.

	Myth	Reality
9	Migrating to Post-Quantum Cryptography (PQC) could be as simple as installing some software upgrades that support new cryptographic algorithms. Right ?	No. As described in Annex J in this document, migrating an organization's IT systems to use standardized PQC will required much more than some "simple monthly software updates". A detailed technical review of current products, systems, infrastructure, and architectures that leverage cryptographic modules will be needed to determine where hardware upgrades, software upgrades, application upgrades, or even complete system replacements, may be required.
10	The steps needed for Quantum Readiness appear to be overwhelming.	While the detailed technical aspects of Quantum threats and cryptographic aspects are beyond the skills of most, the vast majority of Quantum-Readiness steps are typically incremental steps on existing business as well as technical strategic and operational processes and procedures. Open source information, such as this Quantum-Readiness Best Practices guide, plus exemplars, can inform organizations and sectors on how to get started immediately.
11	For symmetric cryptography, all that needs to be done is to ensure that the key length is sufficiently large to provide PQC assurance ; it's that simple, right ?	Strictly speaking, from the "narrow" perspective of symmetric cryptography, yes. If the key length is sufficiently large, then the symmetric cryptography may be deemed safe. However, depending on the use case, in support of the symmetric cryptography, there may be also be a need for key exchange and key management of the symmetric keys, and those techniques may involve the use of asymmetric cryptography. So if this is the case, then the system will be vulnerable to Quantum-based cryptographic attacks.
12	We may use some cryptography that will not be standardized. That's OK, right ?	Using any proprietary or non-standard cryptography, or any algorithm that has not received substantial review, is a big security risk.

APPENDIX B: QUANTUM-SAFE POLICIES, REGULATIONS AND STANDARDS

B.1 Quantum-Safe Policies

The Canadian Centre for Cyber Security introduced new guidance on preparing for Post-Quantum Cryptography (PQC) in the following document published during 2024:

- **Cryptographic Algorithms for Unclassified, Protected A, and Protected B Information (Version 3)** ITSP.40.111, March 18, 2024, 30 pages

  This document identifies and describes recommended cryptographic algorithms and appropriate methods of use that organizations can implement to protect sensitive information.

  Section 13 Preparing for post quantum cryptography

  NIST is expecting to finalize the first set of standards (for PQC) in 2024. In the meantime, we recommend the following high-level steps:

  - *Evaluate the sensitivity of your organization's information and determine its lifespan to identify information that may be at risk (e.g., as part of on-going risk assessment processes).*
  - *Review your IT lifecycle management plan and budget for potentially significant software and hardware updates.*
  - *Educate your workforce on the quantum threat.*
  - *Consider using Stateful Hash-based Signature schemes if you meet the criteria in Section 5.5.*

  For more detailed information on how to prepare, see Preparing Your Organization for The Quantum Threat to Cryptography - ITSAP.00.017.

  Organizations should wait until standards for quantum-resistant public-key encryption and signature schemes are finalized before using any candidate algorithm to protect information or systems.

Cryptographic algorithms for UNCLASSIFIED, PROTECTED A, and PROTECTED B Information
CCCS, March 18, 2024, Page 24

B.2 Quantum-Safe Regulations

Canada has not enacted any regulations related to quantum-readiness or quantum-safe cyber security to date.

B.3 Quantum-Safe Standards

The U.S. National Institute of Standards and Technology (NIST) began work on new standards for PQC in 2015. NIST's goals continue to include publishing a first set of PQC standards in 2024. [79]

During the summer of 2023, NIST invited public comments on three candidate (PQC) Federal Information Processing Standards (FIPS), for a public-key encryption/key-encapsulation mechanism (KEM) and for digital signatures, as follows:

- ***FIPS 203, Module-Lattice-Based Key-Encapsulation Mechanism Standard***
- ***FIPS 204, Module-Lattice-Based Digital Signature Standard***
- ***FIPS 205, Stateless Hash-based Digital Signature Standard***

*These proposed standards specify key establishment and digital signature schemes that are designed to resist future attacks by quantum computers, which threaten the security of current standards. The three algorithms specified in these standards are each derived from different submissions in the NIST post-quantum cryptography standardization project (see: **https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization**).*

Request for Comments on Draft FIPS-203, Draft FIPS-204, and Draft FIPS-205
A Notice by NIST on 08/24/2023, published in the U.S. Federal Register

[79] NIST Post-Quantum Standardization project homepage: https://csrc.nist.gov/pqc-standardization

APPENDIX C: U.S. NCCOE PROJECT ON MIGRATION TO PQC

On August 4, 2021, the U.S. National Cybersecurity Center of Excellence (NCCoE) within NIST announced the start of a new project on *Migration to Post-Quantum Cryptography*. [80]

The outputs of this project could input to the development of best practice recommendations for Section 3.4 - Migration to PQC (Phase 4).

> ***The NIST National Cybersecurity Center of Excellence (NCCoE) is initiating the development of practices to ease the migration from the current set of public-key cryptographic algorithms to replacement algorithms that are resistant to quantum computer-based attacks.***
>
> ***The project will provide systematic approaches for migrating from vulnerable algorithms to quantum-resistant algorithms across the different types of assets and supporting underlying technology.***

The NCCoE's scope for this project includes investigating five demonstration scenarios that would be applicable to a broad range of organizations globally (including organizations in Canada). The scenarios are:

> ***Scenario 1: FIPS-140 validated hardware and software modules that employ quantum-vulnerable public-key cryptography;***
>
> ***Scenario 2: Cryptographic libraries that include quantum-vulnerable public-key cryptography;***
>
> ***Scenario 3: Cryptographic applications and cryptographic support applications that include or are focused on quantum-vulnerable public-key cryptography;***
>
> ***Scenario 4: Embedded quantum-vulnerable cryptographic code in computing platforms;*** *and*
>
> ***Scenario 5: Communication protocols widely deployed in different industry sectors that leverage quantum-vulnerable cryptographic algorithms.***

In December 2023, NIST published for comment the Preliminary Draft of Volumes B and C for their new Special Publication (SP) 1800-38A, *Migration to Post-Quantum Cryptography*. [81]

---

[80] Migration to Post-Quantum Cryptography - Project Description NIST, August 2021, 16 pages

[81] Migration to Post-Quantum Cryptography NIST SP-1800-38 (B and C) NIST, December 2023, 155 pages

APPENDIX D: PQC CONSIDERATIONS FOR BLOCKCHAIN / DLT

This Appendix provides a brief introduction to the topic of blockchain and distributed ledger technology (DLT), in the context of Post-Quantum Cryptography (PQC) considerations.

The content provided herein assumes that the reader has some basic knowledge of the architectures and main features of blockchain and distributed ledger technologies.

Problem statement

The current implementations of blockchain and distributed ledger technology applications and solutions will be subject to increased risk with the appearance of quantum computers and quantum algorithms that are able to break the current suite of classical (non post-quantum cryptography) algorithms and implementations, especially asymmetric cryptography algorithms.

Cryptography is one of the key characteristics of the blockchain architecture.

Hashing, public-private key pairs, and the digital signatures together constitute the cryptographic foundations for the blockchain.

In a post-quantum environment, all of the cryptographic foundations are at increased risk, especially the digital signatures and public-private key pairs, which are based on asymmetric cryptographic algorithms.

Cryptocurrencies are one of the major applications of blockchain, where public-private key pairs are used to maintain addresses, and digital signatures are used to digitally sign transactions. Cryptocurrencies are therefore are at risk when cryptographically relevant quantum computers are available in the future.

The time to act is now

The standardization of post-quantum cryptographic algorithms is underway.

For applications of blockchain and distributed ledger technologies, such as cryptocurrencies and smart contracts, now is the time to review quantum related threats, vulnerabilities, impacts, and risks, and start researching, planning, and preparing for the mitigation and migration of those applications to post-quantum cryptographic based solutions.

References / resources

There is a growing set of publicly available resources focused on the topic of how quantum computers and post-quantum cryptography will impact blockchain and distributed ledger technologies, and the applications and solutions that use them, including for example digital currencies and smart contracts.

Below is a small non-exhaustive sample of some references on this topic.

Quantum-Proofing the Blockchain	Blockchain Research Institute, November 2017 Vlad Gheorghiu, Sergey Gorbunov, Michele Mosca, and Bill Munson University of Waterloo https://www.blockchainresearchinstitute.org/project/quantum-proofing-the-blockchain
Quantum-Resistance in Blockchain Networks	Inter-American Development Bank, June 2021 ITE Department & IDB Lab DISCUSSION PAPER No IDB-DP-00866 https://publications.iadb.org/publications/english/document/Quantum-Resistance-in-Blockchain-Networks.pdf
Vulnerability of blockchain technologies to quantum attacks	Joseph J. Kearney, Carlos A. Perez-Delgado, 23 April 2021 University of Kent, School of Computing Canterbury, Kent, UK https://www.sciencedirect.com/science/article/pii/S2590005621000138

APPENDIX E: QUESTIONS TO ASSESS THE PQC POSTURE OF A 3rd PARTY

This Appendix contains a series of questions to help an organization to begin assessing the PQC maturity or 'posture' of a 3rd Party organization that it may do business with. A 3rd Party in this context may be a technology partner or vendor, or a supplier of other products, goods, or services.

The intent/focus is to evaluate a 3rd Party's cryptography and PQC posture, to assist the organization that asks these questions, to determine the risk of doing business with the 3rd Party. This risk determination can and will vary in different organizations based on their risk tolerance associated to this topic.

The questions in this Appendix can be used, wholly or partially, to generate insight into 3rd Party risk associated with the likelihood that the Quantum threat will affect business continuity. The responses by a 3rd Party to these questions may be used by the organization asking these questions, to evaluate risk to their organization, by defining a risk rating that is aligned to their organization's risk tolerance.

Different Questions for Different Time Periods

Three sets of assessment questions are provided below, to assist in determining a 3rd Party's maturity in cryptography and posture with respect to Post-Quantum cryptography migration. Each set of questions is designed for a different time period associated with the following stages of the Post-Quantum Cryptography migration:

A. Pre-Standardization (Today)

B. Post-Standardization (Starting 2025 or 2026)

C. Post-Quantum (Starting 2030 or later)

A) Questions for the Pre-Standardization Period (Today)

The following questions are for the period of time before quantum-safe algorithms and PQC standards are finalized, and before government agencies decide on the set of standardized quantum-safe algorithms they will recommend be used.

This period is best characterized with planning for PQC migration.

3rd Party PQC Posture Assessment Questions (Pre-Standardization)

1. Have you (viz., the 3rd Party being asked) considered the future impacts of quantum computing in the cryptography used to deliver your services?
   Response Selection: Yes, No

2. Do you have a well-defined and up-to-date cryptographic management practice within your organization which includes:
   a. An approved cryptographic Policy and/or Standard?
      Response Selection: Yes, No, In progress
   b. An up-to-date inventory of cryptography usage (at rest and in transit)?
      Response Selection: Yes, No, In progress
   c. An up-to-date inventory of cryptographic artifacts, components, modules, and systems?
      Response Selection: Yes, No, In progress
   d. Do you have up-to-date operational processes and procedures for managing cryptographic technology?
      Response Selection: Yes, No, In progress
   e. Do you have a documented, up-to-date, and approved process for the upgrade and replacement of obsolete and deprecated cryptography?
      Response Selection: Yes, No, In progress
   f. Do you have cryptographic agility capabilities?
      Response Selection: Yes, No, In progress

B) Questions for the Post-Standardization Period (Starting 2025 or 2026)

In the face of shifting market demands, technological advances, and customer expectations, industry standards may be revised and enhanced. The questions proposed in this section will concentrate on the early stages of established standards.

These questions are for the period after quantum-safe algorithms and PQC standards have been fully defined. This period is best characterized as the time for organizations to start migrating their IM, IT and OT products and systems to PQC, and to complete their migration as soon as practical.

3rd Party PQC Posture Assessment Questions (Post-Standardization)

1. Do you (viz., the 3rd Party being asked) have a well-defined and up to date cryptographic management practice within your organization which includes:
    a. An approved cryptographic Policy and/or Standard?
        Response Selection: Yes, No, In progress
    b. An up to date inventory of cryptography usage (at rest and in transit)?
        Response Selection: Yes, No, In progress
    c. An up to date inventory of cryptographic artifacts, components, modules, and systems?
        Response Selection: Yes, No, In progress
    d. Do you have up to date operational processes and procedures for managing cryptographic technology?
        Response Selection: Yes, No, In progress
    e. Do you have a documented, up to date, and approved process for the upgrade and replacement of obsolete and deprecated cryptography?
        Response Selection: Yes, No, In progress
    f. Do you have cryptographic agility capabilities?
        Response Selection: Yes, No, In progress
2. Does your organization have an approved PQC migration strategy/plan?
    Response Selection: Yes, No, In progress
3. Do you have funding allocated for the PQC strategy/plan?
    Response Selection: Yes, No, In progress
4. Have you begun the migration?
    Response Selection: Yes, No, In progress
5. When do you expect your PQC migration to the completed?
    Response Selection: < 2 years, 2-5 years, more than 5 years

C) Questions for the Post-Quantum Period (Starting 2030 or later)

Whereas the focus of the questions up to this point has been on the risk posed by third parties, depending on their quantum posture. The questions in this section, however, can also be seen as guidance for third parties, which will need to be quantum-ready for their own purposes (notably business continuity) even if the haven't been pressed to do so by their customers or partners.

These questions are for the period of time after a quantum computer has successfully proven classical cryptography to be vulnerable. This period is best characterized with realized risk to classical cryptography.

3rd Party PQC Posture Assessment Questions (Post-Quantum)

1. Do you (viz., the 3rd Party being asked) have a well-defined and up to date cryptographic management practice within your organization which includes:
    a. An approved cryptographic Policy and/or Standard?
        Response Selection: Yes, No, In progress
    b. An up to date inventory of cryptography usage (at rest and in transit)?
        Response Selection: Yes, No, In progress
    c. An up to date inventory of cryptographic artifacts, components, modules, and systems?
        Response Selection: Yes, No, In progress
    d. Are you aware of any cryptography within your organization which should not be used in light of the quantum computing threat?
        Response Selection: Yes, No, In progress
    e. Do you have up to date operational processes and procedures for managing cryptographic technology?
        Response Selection: Yes, No, In progress
    f. Do you have a documented, up to date, and approved process for the upgrade and replacement of obsolete and deprecated cryptography?
        Response Selection: Yes, No, In progress
    g. Do you have cryptographic agility capabilities?
        Response Selection: Yes, No, In progress
    h. Is your organization fully migrated to Post-Quantum Cryptography?
        Response Selection: Yes, No, In progress
2. If the answer to 1h is “No” or “In Progress” :
    a. Does your organization have an approved PQC migration strategy/plan?
        Response Selection: Yes, No, In progress

b. Do you have funding allocated for your PQC strategy/plan?
Response Selection: Yes, No, In progress

c. When do you expect your PQC migration to the completed?
Response Selection: < 2 years, 2-5 years, more than 5 years

3. If the answer to 2c is “2-5 years” or “more than 5 years” :

a. Does your organization (viz., the 3rd Party being asked this question) have crisis-management capacity, as it may be necessary given your circumstances?
Response Selection: Yes, No, In progress

APPENDIX F: TEMPLATE TO CATALOG TECHNOLOGY VENDOR/ SUPPLIER PQC CAPABILITIES

This Appendix contains a template that an organization could use to begin compiling a view of the PQC roadmaps (e.g., PQC features, capabilities, compliance to standards, and anticipated timelines for commercial availability) for each of the technology vendors and/or suppliers it deals with.

This template, or a customized version of it, can be used to canvas a technology vendor or supplier to gather information needed to inform your PQC migration planning, by gathering relevant information about that vendor or supplier's products manufactured by that vendor as used within your organization. Note that the development of an organization's timeline (and project plan) for migrating to PQC may gated by the PQC implementation timelines of its technology vendors and suppliers.

The column headings shown below can be used as a starting point to canvas suppliers for information needed to develop an organization's PQC migration project plan and schedule. This template can also be used as part of the RFP process during acquisition of new products.

When using this template for an existing vendor/supplier, prior to sending, insert a description of the vendor's products and versions (of those products) currently used within your organization.

Technology Vendor/Supplier: *Add Vendor Name Here*

To assist in Post-Quantum Cryptography migration planning, complete the following table for all technology products/services, including current targets on release of a Quantum-Safe version and the supported algorithms.

#	Product (Name, #, Identifier)	Current Version	Quantum-Safe Version	Release/Target Date	PQC Algorithms Supported
Ex1	*Product with no cryptography (example)*	*v1.2.0*	*NA*	*01 March 2015*	*NA - No cryptography present*
Ex2	*Product with cryptography - (future release example)*	*v1.4.0*	*Future - v2.0.1*	*Q1 2025*	*FIPS 203 (Kyber) FIPS 204 (Dilithium)*
Ex3	*Product with cryptography - (quantum safe example)*	*v2.5.1*	*v3.1.x*	*2026-02-01*	*FIPS 203 (Kyber) FIPS 204 (Dilithium)*

#	Product (Name, #, Identifier)	Current Version	Quantum-Safe Version	Release/Target Date	PQC Algorithms Supported

Note that the column headings in this template may and should be revised as appropriate to gather relevant information for your organization. For example, additional columns may be added to:

- denote your organization's use of the vendor's product (e.g. secure data transfer, file storage, user authentication, signing and digital signatures, key establishment, certificate management);
- ask for more information about the PQC algorithms supported (e.g., which standard(s) do the PQC algorithms comply with?);
- ask about plans for certification (e.g. FIPS 140), when such certification supports PQC;
- ask if products support cryptographic agility;
- ask about software and firmware upgrade policies and procedures for any necessary or large agility updates; and
- more . . .

APPENDIX G: PQC ROADMAP QUESTIONS TO ASK VENDORS

This Appendix contains eight "PQC Roadmap" questions that have been developed for owners and operators of critical infrastructure (CI) to send to their vendors of Information or Communications Technology (ICT) products or services.

Background / Overview

From January to March 2023, the CFDIR Quantum-Readiness Working Group (QRWG) drafted an initial set of "PQC Roadmap" questions to seek information from vendors that will be needed by CI owners and operators to inform their Post-Quantum Cryptography (PQC) adoption/migration planning.

The QRWG "alpha tested" the utility of the questions in this Appendix by asking several organizational members of the CFDIR to answer them during April and May 2023. That process led to the revision of some questions to clarify the information being sought. The revised questions were vetted through additional testing and are presented below.

"PQC Roadmap" questions for vendors of ICT products and/or services

Q1: What can you share about your roadmap for including post-quantum cryptography (PQC) in your [**Product / Service**], such as a timeline for when PQC support will be available to customers for all quantum-vulnerable public key cryptography usage by your [**Product / Service**] ?

Q2: Will support for PQC in your [**Product / Service**] be made available through patches or updates under existing contracts and purchases?

Q3: Will your [**Product / Service**] require customers to replace existing hardware or make system architecture changes to support the PQC migration?

Q4: How will your [**Product / Service**] support cryptographic agility to allow flexible administration of configurations for planned cryptographic migration, or an unplanned and immediate migration to remediate a weakness in an algorithm?

Q5: What operational/configuration guidance will you be providing customers on how to migrate your [**Product / Service**] to utilize PQC?

Q6: When your [**Product / Service**] is updated to support PQC, will you ensure the cryptography is independently validated for implementation assurance, for example FIPS 140-3 certification under the Cryptographic Module Validation Program (CMVP)?

Q7: Are your 3rd party suppliers aware of and addressing the quantum computing threat, and are you evaluating how their PQC posture may impact your business operations and your customers?

Note: Appendix E of the CFDIR *Quantum-Readiness Best Practices v.03* provides questions an organization may use to assess the PQC posture of a third-party : https://ised-isde.canada.ca/site/spectrum-management-telecommunications/sites/default/files/attachments/2023/cfdir-quantum-readiness-best-practices-v03.pdf.

Q8: Can you nominate a contact person for any follow-up questions on your answers to questions 1 to 7 ?

How to use the [Product / Service] field that appears on some of the questions

The first six questions include a field denoted by square brackets: [**Product / Service**].

This field is a placeholder to identify the spot, in each question, where the name of a vendor's product or service should be inserted before sending the questions to that vendor.

In situations where a CI owner or operator uses more than one product or service from the same vendor, it is important to consider that the PQC roadmaps for the different products and services may not be identical. As a result, we recommend CI owners or operators ask their vendors to answer all eight PQC Roadmap questions for each of the products and/or services of interest that are provided by those vendors.

One way to ask a vendor about their PQC Roadmaps for different products or services is to send multiple copies of the questions to the vendor, and to write the name of a different [**Product or Service**] into each set of questions.

The contents of this document were developed
during the course of CFDIR QRWG meetings and workshops
between July 2020 and June 2024.

This document will be updated annually,
to reflect industry feedback from implementing
the best practices described herein.

TLP : CLEAR

Version 04 - July 10, 2024

Prepared by the Quantum-Readiness Working Group
of the Canadian Forum for Digital Infrastructure Resilience