Post-Quantum Cryptography (PQC) Working Group

Infrastructure Inventory Technical Paper

FS-ISAC

PQC Working Group

Infrastructure Inventory

Contents

Executive Summary

The ability for an organization to understand its ability to react to changes in the cryptographic landscape requires knowledge of all uses of cryptography across its businesses. It is broader than just the encryption keys and algorithms and must also include the underlying technology and business processes that are being supported.

Introduction and Scope

The scope of this paper is limited to Post-Quantum Cryptography (PQC) and advances in PQC that may negatively impact the security of algorithms currently used to protect confidentiality and integrity of data in the financial industry. The larger encompassing field of Quantum Technology (including, quantum computing, sensing, navigation, quantum networking, Quantum Key Distribution, etc.) is not in scope.

In Scope:

- Keys created and managed by an organization
- Choices of cryptographic algorithms and their usage
- Configurations available to an organization
- Hardware devices, models, and manufacturers
- Operating Systems, vendor, product, and version
- An organization's standards and minimum cipher/key length, etc.
- Asset scanning and Asset Inventories
- Data being encrypted or cryptographically transformed
- Business processes that are supported by the various cryptographic methods
- Third-party SaaS providers or other service offerings where an organization has no direct responsibility to that third-party's cryptography. Here we are still responsible to reach out to third-parties to understand their use or cryptography and potential post-quantum risks to their handling of our data and further to ask for a plan to mitigate the risk and track their remediation efforts.

Please be aware, this paper is not a prescriptive method to build an inventory, but rather a collection of techniques and options to help you, the reader, create a process which suits your organization.

Objective and Principles

By building a clear inventory of assets and uses of cryptography, an organization can proactively identify risks and challenges being introduced by advances in PQC and allow the organization to be crypto agile in planning for future changes in cryptographic requirements.

Principles:

- Determine - What (have we got?), Where (is it?), When (was it created?), Who (owns it?), Why (are we doing it?) How (is it being used?)
- Determine what is in an organization's control and should be inventoried and reported against.
- Determine what is outside an organization's control and should be documented where vendors are asked to provide a risk statement, an approach for risk remediation, and roadmaps as to when we can expect changes to happen that will increase our agility and mitigate risk.
- Standardize business process and deployment methods to simplify both the ability to create inventories and to support cryptographic agility.
- The inventory shall be comprehensive and reflect all components for where PQC advance and associated risks may negatively impact the organization.

Creating a Full Inventory

Inventories can exist in many places and for different reasons. Usually, inventories catalogue discreet items like hardware devices, collections of data, service offerings and so on. There will often be multiple inventories with each inventory designed and built to meet a specific requirement at a point in time. However, the challenge of cryptography is that it applies across multiple different aspects of the enterprise, e.g., in applications or technology platforms. It is very much a cross-disciplinary effort; therefore, a single inventory is unlikely to give a full picture.

To determine the use of cryptography requires a level of knowledge of the different components involved to find and capture the right information. Below is an example of what a holistic inventory may look like.

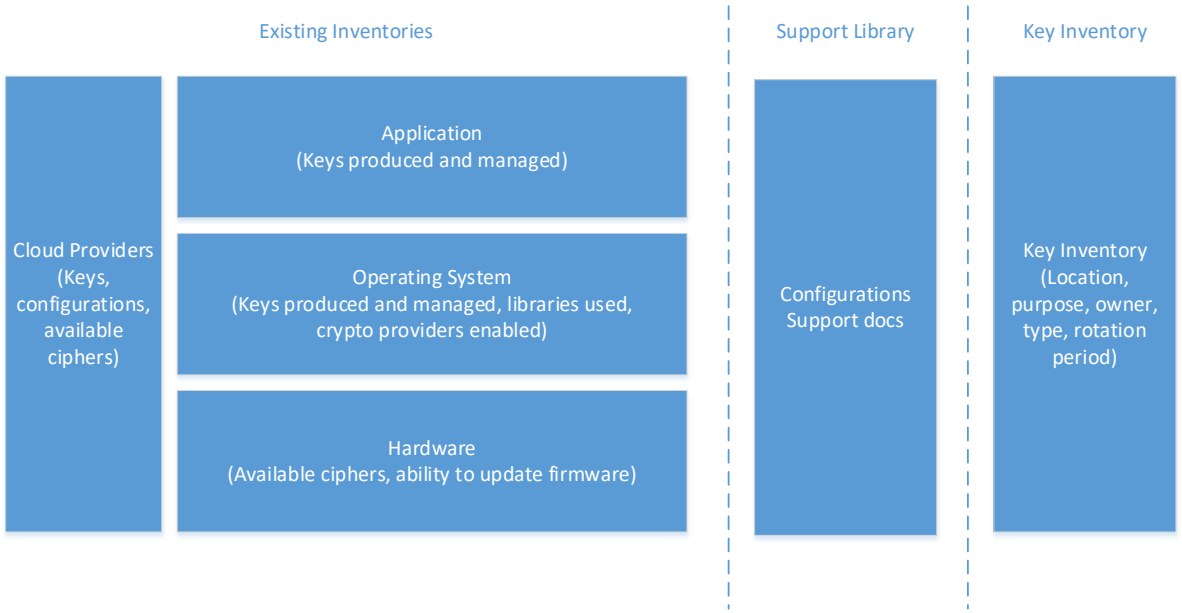


Figure 1: An example holistic inventory

How Do We Get There?

- It is not a simple prescriptive set of steps that will get us there. Rather it's a journey of continuous crypto agility improvement and optimization that needs to be weaved into the organization's strategic planning. Here are some tips to help you embark on your journey:
- Examine existing inventories and the data elements captured. Can they be expanded to better identify cryptographic usage and support cryptographic agility?
- Examine tooling used to capture data. Do the tools provide the necessary visibility into cryptography to understand its runtime usage? Scanning tools are not a panacea in building an inventory using a scanner is only as good as the access it has and the specifics as to what data it is collecting.
- Look for blind spots - It is crucial to understand the potential blind spots scanning tools may have. There may be offline keys, keys in file structures inaccessible to network scanners or keys of unknown format. Where such blind spots exist, it may be required to find alternative methods to scan or work on the assumption of keys being present but accept the inability to validate those keys.

- Determine the **frequency of scans** - Frequency should be relevant to the environment being scanned, with more frequent scans where there is more change activity and for higher risk areas, and lower frequency for lower risk or areas using keys with longer life.
- Develop or refine an existing process for **exception handling** and for handling **alerts triggered** through monitoring (e.g., algorithm deprecated, key expiring).
- Ensure the organization has a well-managed and up-to-date **inventory of software libraries** used across all development projects.
- Develop and deliver **awareness and training** to the organization's development teams that addresses the need for and approaches to better ensure cryptographic agility in the development process and decision making by the project.
- Ensure the organization has a well-managed and up-to-date **inventory of vendors** and a view into their cryptographic usage, agility, and current posture. Include steps in the procurement process to capture this type of data being handled along with clauses in the legal agreements (where possible) to better ensure cryptographic agility.
- Use the inventories to understand the environment and cryptographic usage. Consider developing a **cryptographic agility index** (CAI) that can be used to understand your organization level of preparedness for PQC threat, and support plans to transition to quantum safe algorithm and prioritize work. Some thoughts for creating a CAI are presented in Appendix A.

The following sections provide details on a number of methods that can be used to discover, create, and maintain inventories to accurately reflect cryptographic usage across the enterprise and business functions.

Scanning to Populate an Inventory

Scanning can be used to discover, create, and maintain an inventory. Scanning however is not a panacea in that any scanner is only as good as the access it has and the specifics of what it is looking for.

Any attempt to find encryption keys must have the following items:

- Knowledge of where encryption keys will be located in an operating system, logically or available to a TCP port
- The format of a potential encryption key
- Network access to the devices to be scanned

Multiple scanners may be required to cover different technologies, all feeding into a central inventory.

It is crucial in this process to also understand the blind spots scanning tools may have. There may be offline keys, keys in file structures inaccessible to network scanners or keys of unknown format. Where such blind spots exist, it may be required to find alternative methods to scan or work on the assumption of keys being present but accept the inability to validate those keys.

Scanning can also be used to validate inventories and highlight changes for review or error. The frequency of scanning should be relevant to the environment being scanned, so more frequent scans for higher change activity areas, lower frequency scans for lower risk, more long-lived areas, and targeted frequent scans on higher risk areas. This allows for prioritized response to changes and alerts.

Documenting the Unknown

Where a device, piece of software or application uses cryptographic methods that are not visible or configurable by an administrator, those must be treated as black boxes. A simple documented model of an inventory item would allow detail to be captured without trying to understand every library, key, and internal process.

A library of assets relating to each of these items would provide a reference to understand the wider cryptographic landscapes.

In the event of changes in the Cryptographic landscape or regulation, it would be up to the vendors supporting these items to make the relevant changes and for those changes to be tracked across an organization.

Where a third-party is providing a service, an organization's inventory should reflect only that which they are responsible for or can administer. Other inaccessible keys or cryptographic capabilities should be treated as a black box.

Based on the analysis, which existing cryptographic algorithms are at risk, the next important step is to understand which quantum computer safe alternatives will be available.

Public Key Infrastructure (PKI)

As mentioned above, it is necessary for organizations to establish visibility into their crypto systems and resources to know everywhere cryptography is used and how it's being used to identify PQC risks and be able to address them. PKI (Public Key Infrastructure) is an area we recommend inventorying.

A PKI is used by an organization to protect its critical infrastructure and secure data and communication exchange through authentication and encryption algorithms. The technology is critical because it enables a wide range of initiatives from cloud modernization to IoT, to DevOps pipelines, etc.

PKI can be complicated in terms of how it's being used by systems. In its simplest form, it manages security through encryption it assigns identities to keys so that the recipients can verify the owners. This process involves public and private keys to encrypt and decrypt messages, those keys can be used by people, applications, and devices. Most public-key algorithms in use today will leverage asymmetric encryption, which itself will be vulnerable to quantum computing attack. Any attack on compromised PKI puts public and private keys at risk as they can become useless or be used for malicious acts. For example, if a hacker can listen in on a key exchange and discover the symmetric key established to encrypt data on that communication, channel, the confidentiality of the communication has been breached. Note that this type of attack could be used today to harvest communications and uncover the data in the future once a post-quantum attack becomes plausible. This is referred to as harvesting. A second risk lies within the use of private keys and certificates to bind an identity to an entity such as a server or person, if compromised, certificates used to establish identity or for non-repudiation can no longer be trusted.

Here are some tips to help you achieve crypto agility with your PKI infrastructure and protect asymmetric algorithms which may become breakable in the event of advances in PQC:

- Create an inventory that lists all applications and communications channels that use asymmetric cryptography
- Verify key management for your PKI (including key management) and document the PKI process
- Store encryption keys in a trusted hardware security module (HSM) where practical
- Know what's at risk, protect data that needs to be protected in the long term
- Review and re-issue certificates to ensure their integrity on a regular basis and to prevent risks of compromise specially for the ones with extended validity period, such as 25 years or longer

Application Development (AppSec)

In this section, we examine the Application Development Process and aspects that can be leveraged to assist in building out a comprehensive cryptographic inventory. Applications often use cryptography to accomplish specific tasks including the protection of sensitive data, signing and/or verifying the authenticity of transactions or documents, etc. These tasks rely on specific algorithms, cryptographic libraries, keys, modes of operation, protocol versions, certificates, etc. It is this information that we are interested in collecting and adding to the firm's inventory.

A number of methods can be used to collect and maintain this type of data. These methods include self-identification by the application owner where the use of cryptography is explicitly recorded in the firm's application inventory, manual code review, static code scanning, dynamic execution, and file system discovery. In addition, the introduction of Software Bill of Materials (SBOM), although in its early days, may provide a view into the use of cryptography in third-party products. Each of these methods is discussed in the subsections below.

Self-Identification

This is the simplest method, and it can provide a starting point. Organizations likely have an application inventory today where the inventory identifies characteristics of the

application such as data classification, whether the application handles customer personal identifiable information (PII) data, whether it supports financial transactions, etc. Some of the existing fields may provide insight into whether the application may use encryption based on the firm's data protection policies. Here, you may want to add a field and require the application owner to explicitly record whether the application uses encryption, the type of encryption and a brief description as to its usage – e.g., to protect personal data stored in the application database using AES-128, or to digitally sign transactions. This data can then be correlated with data collected via other methods and used to identify key risks to the firm posed by weak or deprecated cryptographic algorithms and to prioritize remediation efforts.

Manual Code Review

Manual code review is a technique that may be useful for top-tier applications that are handling highly sensitive data. Note however that it is unlikely that this method will scale and additional methods that leverage automated tools to address discovery should be adopted in order to establish and maintain a comprehensive and useful inventory.[1]

Static Scanning

Firms may already have static scanning tools built into their CI/CD pipelines. While these tools are typically used to check for vulnerabilities in code, they may be useful for identifying the use of cryptography or, more specifically, the use of cryptographic functions that we believe will not be PQC safe – see table below.

1 Studies show that developers should review no more than 400 lines of code per hour to keep the accuracy above 90%. As such, manual review will be slow and, once applications change, your inventory will be out of date and new code and code changes will need to be reassessed.

Cryptographic Algorithm	Type	Purpose	PQC Approach
AES-256	Symmetric	Encryption / Confidentially	Larger Key Sizes
SHA-256, SHA-3	Hash	Hash Functions / Integrity	Larger Key Sizes
RSA	Asymmetric [2]	Signatures / Key Establishment	NO LONGER SECURE
ECDSA, ECDH (Elliptic Curve Encryption)	Asymmetric	Signatures / Key Establishment	NO LONGER SECURE
DAS (Finite Field Encryption)	Asymmetric	Signatures / Key Establishment	NO LONGER SECURE

Table 1: PQC Risks - Asymmetric cryptography is primarily at risk
Source: US National Institute of Standards and Technology (NIST) 8105[1]

This method will provide visibility into the algorithms being called by an application which can then be cross referenced against the set of algorithms that have been deemed as not being post-quantum safe which will help to define scope.

In preparation for the future, once specific cryptographic function calls have been identified, in anticipation of a future need to replace an algorithm with a PQC-safe alternative, you may want to consider introducing a shim. A shim will provide an abstraction

2 Asymmetric encryption uses two separate keys for encryption and decryption - a public key and a private key. It is a foundational component of a public key infrastructure (PKI) which makes secure communications over the internet possible.

layer between the function call and the specific algorithm being used by the application – see Appendix C.

Potential limitations with static scanning include:

- Lack of precision in that each function call within an application may depend on parameters that are supplied in configuration files that are accessed at runtime.
- The scan results may provide too much information in that they may provide detail on all the cryptographic algorithms that are detected and available to an application whether an algorithm is used or not. For example, many applications include a cryptography library that includes insecure algorithms such as MD5 or DES for legacy reasons, but the application may never call them.
- If cryptographic libraries are loaded dynamically and algorithm identifiers are created by reflection from strings in a run-time configuration file, the static scanner may not have sufficient visibility as to what is actually used.

To address these limitations and further refine the scope, dynamic analysis (described below) can be used.

Dynamic Analysis

While knowing what algorithms are inside an application can be useful as a very first step, it does not produce a view as to what algorithms are being used and may need to be replaced. Dynamic analysis at run-time or Interactive Application Security Testing (IAST) tools may help in this regard. The advantage of this approach is that these tools have visibility into the cryptographic functions that are used, including calls from third-party libraries and framework components. As you might expect, this approach does have its own disadvantages in that it will only work on running applications, and those applications must be exercised to use all code paths in order to identify the comprehensive view into the use of cryptography, either by unit or integration tests, or by tracing a production server.

An approach that best fits into your firm's tooling, methodologies, and capabilities and that leverages multiple methods is recommended.

Software Bill of Materials (SBOM)

The concept of a Software Bill of Materials (SBOM) was introduced in 2018 and is an emerging field. An SBOM can be viewed as a list of ingredients that make up software components. It outlines both what software packages and libraries went into an application and the relationship between those packages and libraries and other upstream projects. In the past year, industry efforts to embrace SBOMs has gain some momentum driven primarily by President Biden's 2021 Executive Order on Cybersecurity[2] and the industry is making progress on standards and methods to generate and share SBOMs.

Note however the availability of an SBOM is only the first part of the puzzle. Once available, its components will need to be mapped to a list of known vulnerabilities for each component. Both pieces of information as well as a process to facilitate the mapping are needed in order to know what's in the software, its vulnerabilities, and whether anything is tied to a PQC risk and needs to be remediated.

The availability and effective use of SBOMs is a topic in itself. Once embraced by the industry and SBOMs become readily available, it will be worth looking to leverage the SBOM to gain visibility into cryptographic weaknesses/risks in your environment and to enrich your cryptographic inventory.

File System Discovery

File systems scans can be used to discover cryptographic components such as keys, key stores, configuration files, certificates, and cryptographic libraries. Note that a file system scan may be able to leverage existing tools such as Tanium or Varonis. A downside however may be the level of noise in that the scan will likely point components that are not actually being used. As such, this method should be used with care and as a method to identify key areas that may support the use of cryptography in your environment and used in conjunction with other methods to achieve better fidelity.

Other Considerations

1. Software as a Service (SaaS) –SaaS providers may offer encryption. However, for operational ease, the SaaS provider may own and manage the key and the choice of algorithm. The method for updating the key would be for your organization to contact the SaaS provider and request the update. You may be able to receive

evidence the keys were rotated, but you will most likely need to ask. In terms of the underlying algorithm, you should ask and record this as part of your due diligence, and if the algorithm(s) are not PQC safe, ask the SaaS provider for a plan and timeline for addressing PQC risk.

2. There are however SaaS providers that support the concept of Bring Your Own Key (BYOK) which will allow for separate encryption of sensitive data using a customer managed key, but the remainder of the data is protected using a platform-provided encryption key such as a "vault" for secrets which are needed within the platform. This "vault" storage is encrypted using a key provided and managed by your organization. While you absolutely can change that key, there is a defined procedure which requires down-time to un-encrypt and re-encrypt using the new key. Some of the SaaS providers handle this with versioning and are more graceful – but this varies by provider. Like above, you should understand the underlying algorithms in that it is the choice algorithm will determine whether your data is susceptible to a PQC attack.

3. A third approach is where the SaaS provider allows the customer to provide and manage the key with the key stored in the customer's own vault. Others may require the use of their vault where you simply are given access to the vault location where the key must be present. Accessibility to changing the keys is typically reasonable, but the impact of changing the key ranges from full outage while re-encrypting the data, to none while the SaaS provider uses versioning and seamlessly re-encrypts behind the scenes.

4. Hardware Security Modules (HSMs) - HSMs will typically contain an organization's most important keys such root signing keys for the PKI and domain specific keys for payment processing and other critical functions. These keys and the algorithms used by the HSMs are important components of our cryptographic inventory.

5. In addition, an HSM typically provides a service (encryption, signing, etc.). It may be useful to examine HSM logs to identify all the apps that are making calls to the HSMs to perform cryptographic functions. Both the app owners and the group that owns the HSM will need to play a role in addressing the PQC risk mitigation tasks.

6. Third-Party Application Programming Interfaces (APIs) - APIs have become ubiquitous. APIs should be included as an asset class and details collected on each API's use of encryption. Weak connection ciphers (as demanded by the third-party) may introduce a risk to the organization that will need to be mitigated.

7. Internet of Things (IoT) - IoT may not necessarily be tied to specific business lines, it however presents a risk. Like APIs, IoT devices should be included as an asset class and details collected on each device type's use of encryption for completeness.

8. Blockchain – Blockchain is an emerging technology and is being used for many applications such as Smart Contracts, Crypto Assets, etc. Blockchain uses cryptography to build and sign blocks as they are added to the chain, to ensure transactional integrity, and in some cases, for privacy. Understanding its usage is vital and should be included in the organization's asset inventory. Blockchain leverages public key cryptography which is vulnerable to PQC risks.

Third-Parties: Leverage (Embed in) Procurement Process

Many (if not all) firms rely on third-party vendors. These vendors provide various services which may include performing financial transactions and may be handling sensitive data. In performing these services, it is likely the third-party will use encryption for integrity and protection of data. Examine your existing third-party vendor management process and ensure that you have the ability to identify critical third-parties that are dealing with transactional data and/or other sensitive data such as personal information (PI) and you have a view into their susceptibility to risks introduced by PQC, how they plan to mitigate the risks, and a date as to when the appropriate mitigation will be in place.

Appendix A - Creating a Cryptographic Agility Index (CAI)

What goes into a cryptographic agility index (CAI):

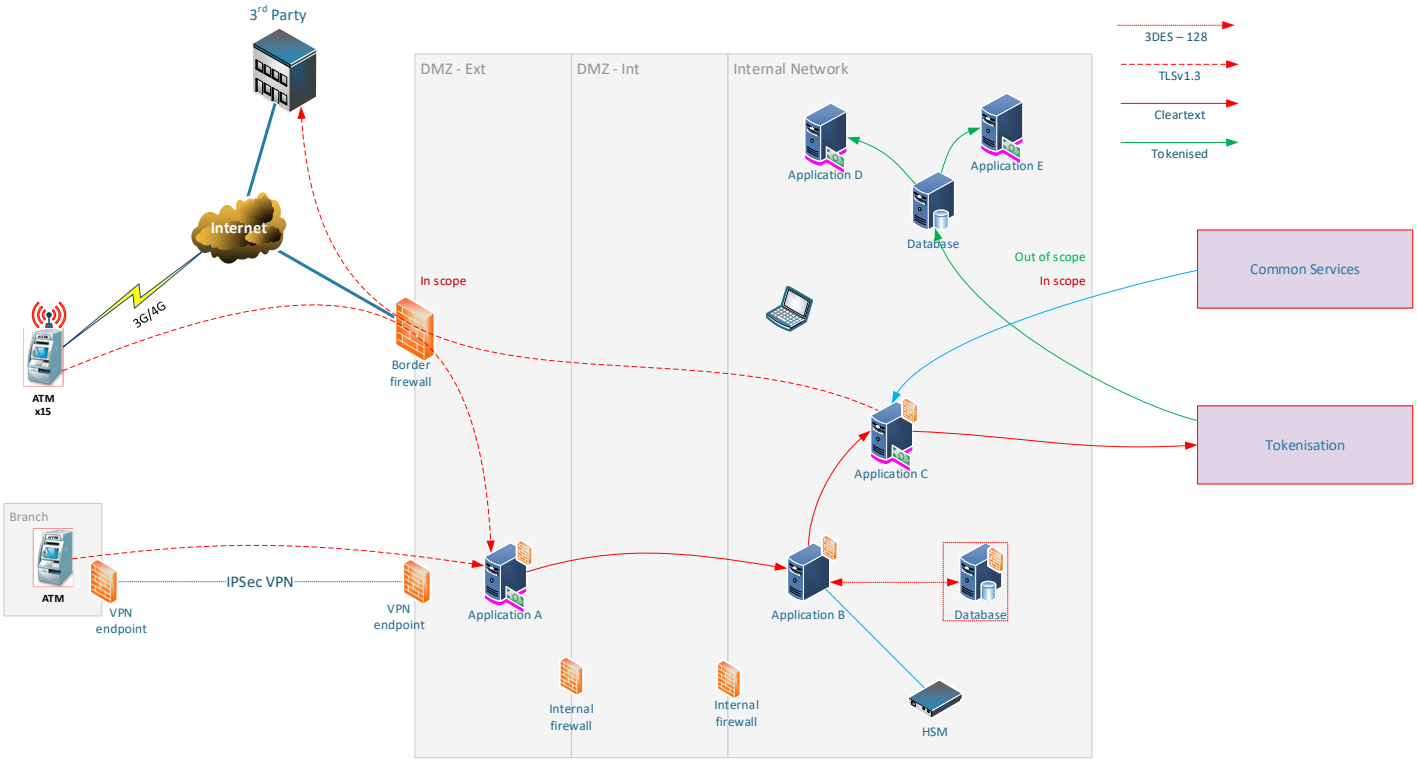
A CAI must take a holistic view that reflects several specific points around prioritization, controls, business capabilities, vendors, mitigation, and implementation plan:

- Start a cryptographic PQC program within your organization.

- Define management ownership for cryptography.
- Stand up a Cryptographic Center of Excellence for an enterprise wide crypto planning and adoption of reliable practices.
- Build a transition plan and a roadmap to implement mitigation strategies.
- Identify the threat area and the assets related to it.
- Understand operational constraints.
- Evaluate the expected value of impacted assets being compromised.
- Identify assets that are going to be impacted by the threat, discount the ones that will not be impacted by the threat and take out of scope completely the ones to be phased out or deprecated in the near future. Consider interdependencies of the assets with potential risk, even if it's out of scope, as it may affect the one in scope.
- Understand the priority of each inventory item to the organization. Remember, not all assets have the same value. Consider prioritizing assets according to:
    - Their type: enterprise vs third-party vendor.
    - Risk tolerance and level: 1-low risk; 2-moderate risk; 3-high risk where the risk level can be tied to how an asset is used, e.g., an API to Swift vs an API for a threat intel data feed.
    - Intended scope of use and marked with high level of exposure to threat.
- Identify and fix vulnerabilities before data can be stolen and or compromised. Update with patches as often as possible to remain up to date on security breaches.
- When leveraging open-source components, it is bets practice to select projects with faster patching history and greater flexibility.
- Define security goals and requirements such as the level of security associated with personal data that must be encrypted at rest and in transit.
- Ensure vendors are using advanced encryption standards to protect against attacks. Evaluating the algorithms as part of the vendor evaluation process and understanding how the algorithms are being implemented.
- Understand regulatory mandates to use specific cryptography.
- Upskill and train employees to use new algorithms.
- Consider a single management platform to consolidate certificates and gain visibility into your organization's assets – refer to Public Key Infrastructure section of the main document.
- Last but not least, continuously evaluate the following:
    - Quantum computers timeline may change as more research is being done.
    - Existing functionality to change cryptographic settings.

- The ability to change ciphers: the ability to test and replace current algorithms with new ones.

Appendix B - Sample Cryptographic Diagram



Appendix C – Preparing Your Code for The Future

Once specific cryptographic function calls have been identified, in anticipation of a future need to replace an algorithm with a PQC-safe alternative, you may want to consider introducing a shim. A shim will provide an abstraction layer between the function call and the specific algorithm being used by the application – see Figure 2 below.

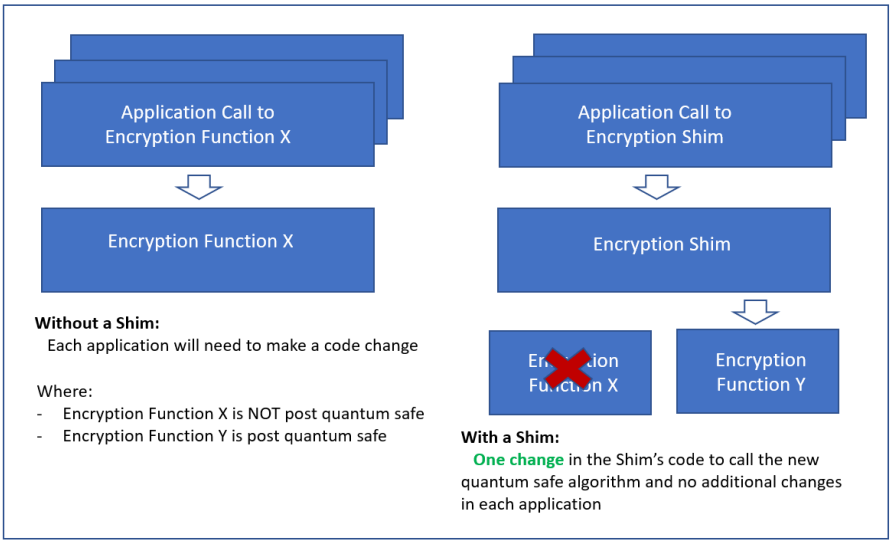


Figure 2: Use of a Shim today as preparation for a transition to a post-quantum safe algorithm tomorrow

This change can be made today in preparation for future usage. Down the road once the new NIST approved algorithms are in place, we can make changes to the shim to leverage the PQC-safe algorithm in one place and it will take effect across all applications that had previously anticipated the change and introduced the shim, leveraged an abstraction layer, in their code.

References

[1] US National Institute of Standards and Technology (NIST) 8105
[2] 2021 US Executive Order on Cybersecurity

If you are an FS-ISAC member and would like to join the PQC Working Group, please email us.