



Q-ENTANGLE 2026

Quantum-Safe Cryptography: Navigating the Migration to Post- Quantum Security

Divya K Konoor,
Senior Technical Staff Member & Chief Architect,
Confidential Computing, Hardware Security Modules, Digital Asset & Security
IBM Z Systems (Mainframe) ,



Our modern digital
world depends on
cryptography

Cryptography impacts everything

Cryptography touches every corner of the digital world

Internet Protocols



Domain Name Service(DNS),
Hyper-text Transfer Protocol
(HTTP), Telnet, SFTP

Critical Infrastructure



Code updates; Control
systems- Oil pipelines, Electric
grids; Car systems,...

Blockchain Applications



Coin wallets, Transactions,
Authentication

Digital Signature Laws



EiDAS - PDF Advanced
Electronic Signature – (PAdES),
Advanced Electronic
Signatures (AES), ...

Financial Systems



Payment Systems: (EMV,
SWIFT, Settlement Systems,
FinTech, ...)

Enterprise Applications



EMAIL – PGP, Identity
Management PKI/LDAP/..., Virus
scanning patterns, PKI Services

Quantum Computing

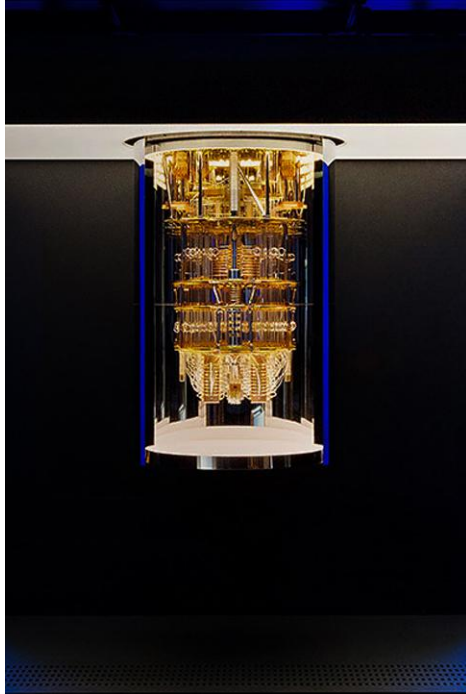


Quantum computers can,
in principle, perform certain mathematical
algorithms exponentially faster than a
classical computer

Quantum computers vs. classical computer

Quantum Computer

- Harnesses the laws of quantum mechanics to solve problems too complex for classical computers.
- Calculates using qubits which can represent 0 and 1 at the same time.
- Well suited for tasks like optimization problems, data analysis and simulations.
- Superposition / Entanglement / Interference



Classical Computer

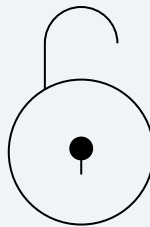
- Uses transistors and circuits whose behavior is defined by classical physics.
- Calculates using bits which can represent either 0 or 1.
- Everyday processing tasks are best handled by classical computers.



What will an adversary be able to do?



Manipulate updates, forge transactions, spoof communications through fraudulent authentication



Decrypt lost or harvested confidential historical data through cracking encryption keys



Manipulate legal history by forging digital signatures

Find your keys or derive your keys !

Adversaries can:

- Create fake identities for websites
- Create fake software downloads and software updates
- Launch extortion attacks by threatening to disclose harvested data
- Create indistinguishable fraudulent land records, lease documents, or command records

The Problem

Public key algorithms:

Completely broken by large scale quantum computer

Example Mitigations:

New algorithms and schemes needed

Symmetric key and hashing algorithms:

Impacted by quantum computing – algorithm strengths are reduced

Example Mitigations:

Increase the key or digest sizes (i.e., AES-256, SHA-512)

The Impact

- **Shor's algorithm** for factoring and discrete logarithms can completely break the RSA and Diffie-Hellman cryptosystems, and their elliptic-curve-based variants
- **Grover's algorithm** could be used to speed up an exhaustive search for symmetric keys or reverse engineer a cryptographic hash

Algorithm*	Purpose	Impact from quantum computer
DES, TDES	Encryption	No longer secure
AES-256	Encryption	Secure
SHA-256, SHA-3	Hash Functions	Secure
RSA	Signatures, Key Establishment	No longer secure
ECDSA, ECDH (Elliptic Curve Cryptography)	Signatures, Key Exchange	No longer secure
DSA (Finite Field Cryptography)	Signatures, Key Exchange	No longer secure

NIST Approved Post-Quantum Cryptography Standards

Aug 2024

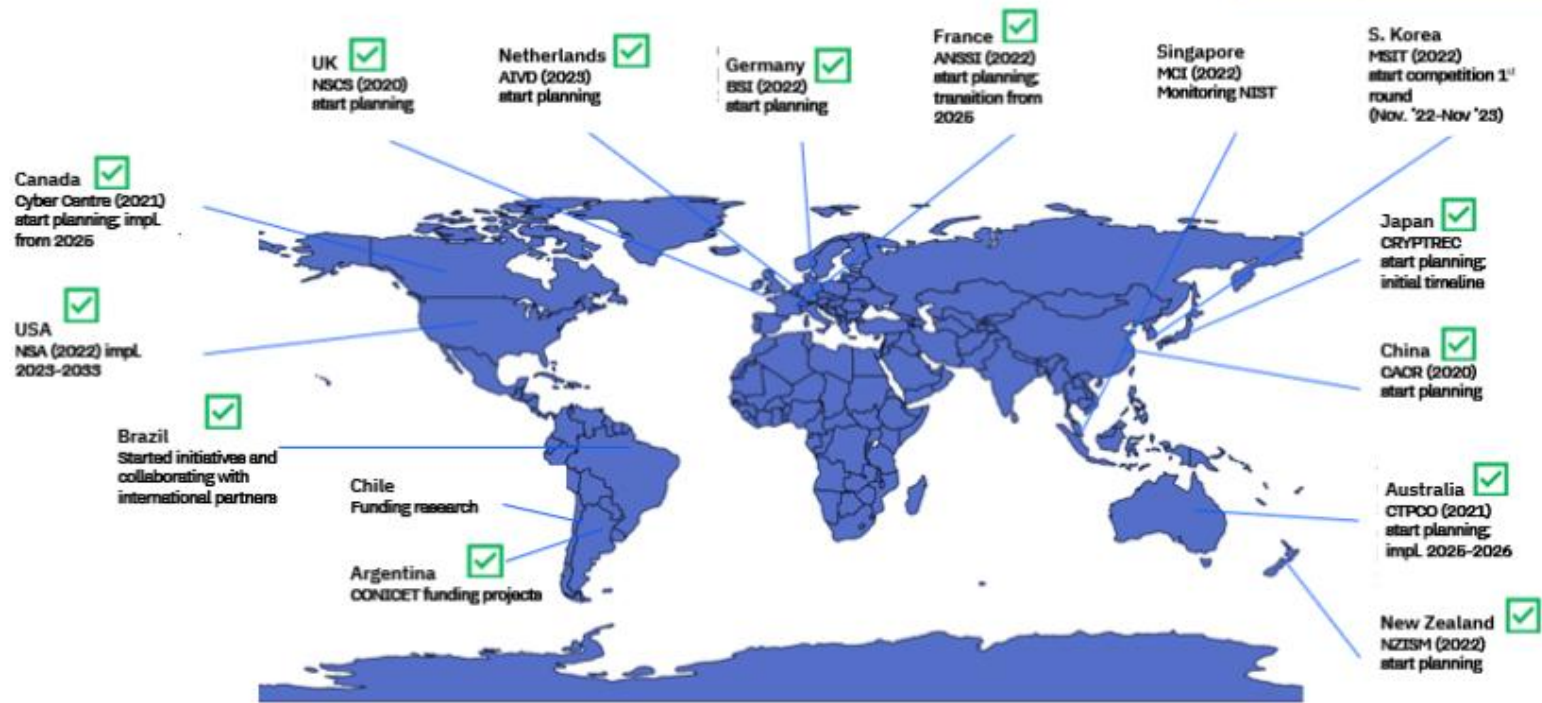
FIPS 203 - ML-KEM	FIPS 204 - ML-DSA	FIPS 205 - SLH-DSA	FIPS 206 - FN-DSA (Upcoming)
• Module-Lattice-Based Key-Encapsulation Mechanism Standard • A general encryption(public key cryptography) algorithm for key establishment • Formerly known as CRYSTALS-Kyber • ML-KEM-512, ML-KEM-768, and ML-KEM-1024	• Module-Lattice-Based Digital Signature Standard • A digital signature standard for authentication • Formerly known as CRYSTALS-Dilithium • Fast signing / verification • 2-4KB signature size • ML-DSA-44(2x2), • ML-DSA-65(3x3), • ML-DSA-87(4x4)	• Stateless Hash-Based Digital Signature Standard • A digital signature standard for authentication • Formerly known as SPHINCS+ • Hash-based backup signature scheme • Much larger signature (7-8KB) size compared to ML-DSA; slower processing speed	• FFT over NTRU-Lattice-Based Digital Signature Algorithm) • Designed for digital signatures and moving toward final publication as FIPS 206 • Formerly known as FALCON • Under 1KB signature size • Catered to address the bloated file size problem • IoT / Digital Asset smart contract etc

Things to consider switching over to PQC

- **Larger Key and Signature Sizes**
 - *Classical ECC: Public keys are tiny, typically around 32 bytes (256 bits)*
 - *Classical RSA: Standard keys range from 256 bytes to 512 bytes (2048 to 4096 bits)*
 - *PQC ML-KEM (Kyber): Standard quantum-resistant public keys for key encapsulation range from 800 to 1,184 bytes depending on the security level*
 - *PQC Digital Signatures (e.g., ML-DSA/Dilithium or SPHINCS+): Signature and public key sizes jump drastically from a few hundred bytes to anywhere from 2 KB to over 30 KB*
- **Higher memory consumption**
- **Network Bandwidth and Network Fragmentation**
 - *Classical keys fit inside one packet, larger PQC keys and certificate chains exceed this limit, causing packet fragmentation, higher latency, and potential dropouts on constrained networks*
- **Storage and Hardware Constraints**

What is happening around the world?

Governments are aware of the risks and recommend that organizations begin planning and/or transitioning.



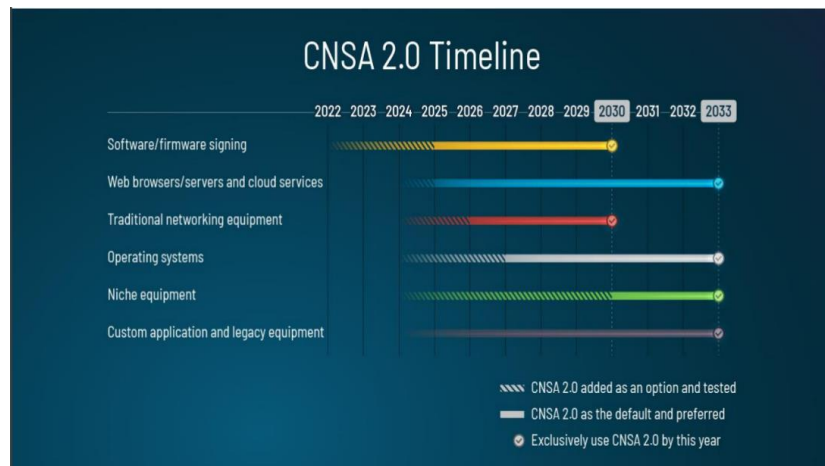
Other countries moving – Spain, Czech Republic, G7

NSA - Commercial National Security Algorithm Suite 2.0

Implementing CNSA 2.0 requirements for NSS	Preferred adoption by	Exclusive use by
Software and firmware signing	2025	2030
Web browsers/servers and cloud services	2025	2033
Traditional networking equipment (for example, virtual private networks, routers)	2026	2030
Operating systems	2027	2033
Niche equipment (such as constrained devices, large public -key infrastructure systems, and so on)	2030	2033
Custom applications and legacy equipment	-	2033

NSA sets 2035 deadline for adoption of post-quantum cryptography across national security systems

Algorithm	Purpose	Relevant NIST standards
General purpose algorithms		
AES-256	Block Cipher	FIPS 197
SHA-384 or SHA-512	Cryptographic Hash	FIPS 180-4
ML-KEM-1024	Public Key Establishment	FIPS 203
ML-DSA-87	Digital Signature	FIPS 204
Specific use case algorithms		
LMS or XMSS (LMS 256-192 recommended)	Software/Firmware Signature	SP 800-208
SHA3-384 or SHA3-512	Hashing in internal hardware (e.g. secure boot)	FIPS 202



India: sovereign assurance built around the National Quantum Mission

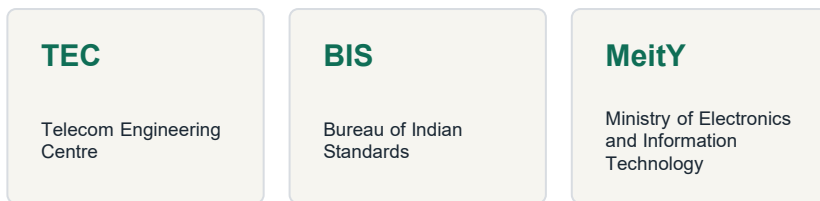
Department of Science and Technology • National Quantum Mission • “Implementation of Quantum Safe Ecosystem in India,” February 2026

A tiered migration timeline



A sovereign three-tier laboratory system

For PQC product testing and certification — aligned with ISO/IEC 17025 and the Digital Personal Data Protection Act



Designation authority shared across all three bodies

Distinctive feature

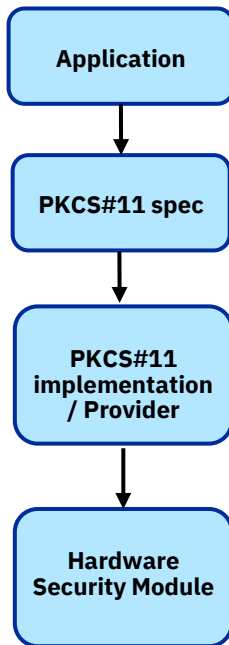
Domestic assurance, reduced dependence on foreign validation

The Department of Telecommunications is expected to mandate PQC readiness for new telecom equipment procured from 2026 onward — a significant signal given the scale of India's 5G rollout and the breadth of public digital infrastructure depending on it.

PQC in Open-Source and Standards

Hardware Security Modules, PKCS#11 and PQC

- **PKCS#11 Specification Version 3.2** introduces official standardized support for post-quantum cryptographic (PQC) mechanisms, approved as an OASIS Standard in July 2026
- PKCS#11 v3.2 does not introduce new structural functions. Instead, applications leverage the **traditional signing API but pass new PQC key types (CKK_ML_DSA) and mechanisms (CKM_ML_DSA)**
- **Standardized Algorithms:** Includes support for NIST-approved post-quantum algorithms like ML-KEM (key encapsulation), ML-DSA, and SLH-DSA (digital signatures)



HSM	PQC Support
IBM Crypto Express 8S	Full Native Support
Thales Luna	Full Native Support
Entrust nShield	Full Native Support
Utimaco cryptoserver	Full Native Support



Post-Quantum Cryptography Alliance



CBOMkit

The project aims to develop and advance tools for creating and analyzing Cryptographic Bills of Materials (CBOMs)—standardized, machine-readable catalogs of cryptographic assets and dependencies—to enable automated compliance, assess quantum vulnerability, and support post-quantum cryptography adoption.

[Learn More](#)



Open Quantum Safe

Open Quantum Safe (OQS) project aims to support the development and prototyping of quantum-resistant cryptography.

[Learn More](#)



PQ Code Package

Creating and maintaining high-assurance implementations of standards-track post-quantum cryptography algorithms.

[Learn More](#)

CycloneDX and CBOM

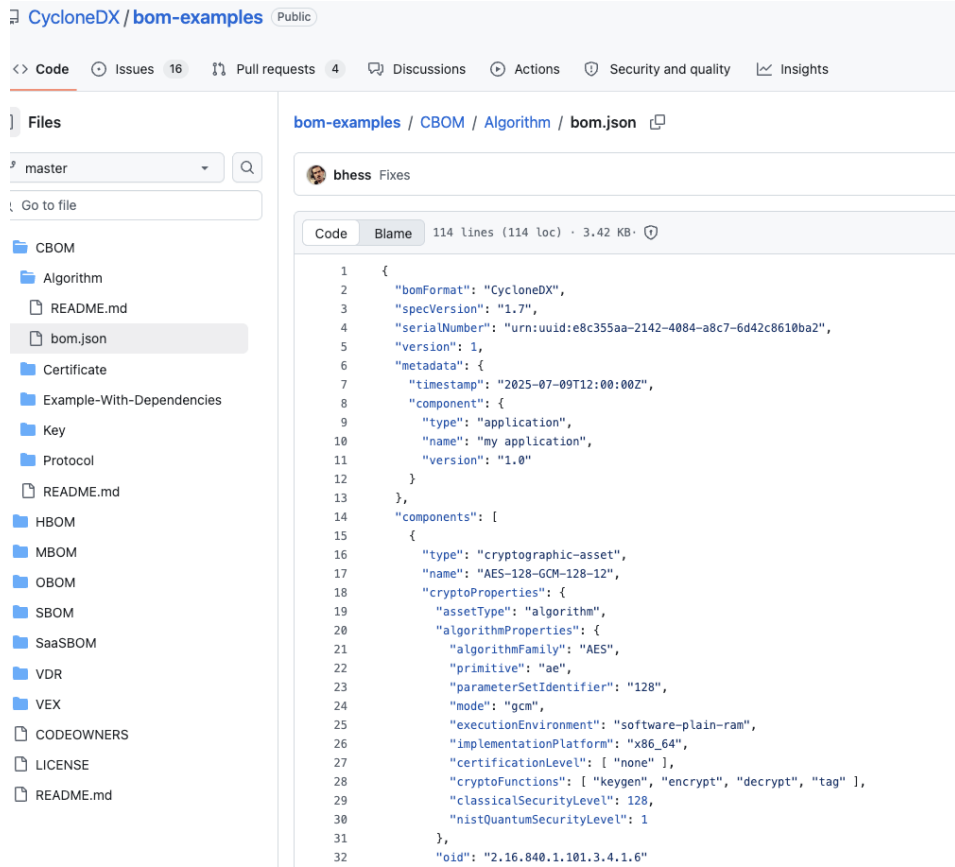
- The CycloneDX CBOM standard, provides a machine-readable way to document and exchange information about the presence of cryptographic assets in applications, enabling automated security analysis, compliance checking, and risk management.

CBOMkit - the essentials for CBOMs

license Apache-2.0 release v2.3.0

CBOMkit is a toolset for dealing with Cryptography Bill of Materials (CBOM). CBOMkit includes a

- CBOM Generation** ([CBOMkit-hyperion](#), [CBOMkit-theia](#)): Generate CBOMs from source code by scanning private and public git repositories to find the used cryptography.
- CBOM Viewer** ([CBOMkit-coeus](#)): Visualize a generated or uploaded CBOM and access comprehensive statistics.
- CBOM Compliance Check**: Evaluate CBOMs created or uploaded against specified compliance policies and receive detailed compliance status reports.
- CBOM Database**: Collect and store CBOMs into the database and expose this data through a RESTful API.



The screenshot shows a GitHub repository for `CycloneDX / bom-examples`. The file `bom.json` is selected, showing its content. The file is 114 lines long, 3.42 KB, and was last fixed by user `bhess`. The JSON content defines a CycloneDX component for a cryptographic asset.

```
1 {
2   "bomFormat": "CycloneDX",
3   "specVersion": "1.7",
4   "serialNumber": "urn:uuid:e8c355aa-2142-4084-a8c7-6d42c8610ba2",
5   "version": 1,
6   "metadata": {
7     "timestamp": "2025-07-09T12:00:00Z",
8     "component": {
9       "type": "application",
10      "name": "my application",
11      "version": "1.0"
12    }
13  },
14  "components": [
15    {
16      "type": "cryptographic-asset",
17      "name": "AES-128-GCM-128-12",
18      "cryptoProperties": {
19        "assetType": "algorithm",
20        "algorithmProperties": {
21          "algorithmFamily": "AES",
22          "primitive": "ae",
23          "parameterSetIdentifier": "128",
24          "mode": "gcm",
25          "executionEnvironment": "software-plain-ram",
26          "implementationPlatform": "x86_64",
27          "certificationLevel": [ "none" ],
28          "cryptoFunctions": [ "keygen", "encrypt", "decrypt", "tag" ],
29          "classicalSecurityLevel": 128,
30          "nistQuantumSecurityLevel": 1
31        }
32      },
33      "oid": "2.16.840.1.101.3.4.1.6"
```

Post Quantum Cryptographic Alliance



The [Open Quantum Safe \(OQS\) project](https://openquantumsafe.org/) is an open-source project that aims to support the transition to [quantum-resistant cryptography](#). OQS is part of the [Linux Foundation's Post-Quantum Cryptography Alliance](#).

OQS consists of two main lines of work: [liboqs](#), an open source C library for quantum-resistant cryptographic algorithms, and prototype integrations into [protocols and applications](#), including the widely used OpenSSL library. These tools support [research](#) by ourselves and others.

All of our development takes place on our [GitHub](#) repositories. We welcome new contributors interested in joining our [team](#). Feel free to begin participating on GitHub or say hello on our [Discord server](#) on the [#oqs-general channel](#). We are grateful to the organizations that have supported and continue to support our work, and the academic, industry, public sector, and individual contributors who participate in the project.

<https://openquantumsafe.org/>
<https://openquantumsafe.org/liboqs/>

OpenSSL and PQC

- **OpenSSL 3.5** integrated the first official (NIST) finalized post-quantum standards directly into the primary libcrypto core – without liboqs
- **Pre-OpenSSL 3.5**, no native support to PQC. OpenSSL 3.0 -3.4 uses **oqs-provider** plugin (wrapper around the **liboqs** C lib)
- **"Hybrid" by Default in TLS 1.3**
 - Connections now establish a **X25519MLKEM768** key share out of the box.
 - Combines the classic, ECC (X25519) with ML-KEM-768. If either algorithm holds up, your data remains secure



September 10, 2026

OpenSSL's new alpha build speeds up post-quantum crypto

The OpenSSL project released the first alpha of OpenSSL 4.1.0, giving developers an early look at a version built for encrypted communication over unreliable connections and faster **post-quantum cryptography**. This marks the opening test build for a version still months from general availability.



April 9, 2025

OpenSSL prepares for a quantum future with 3.5.0 release

The OpenSSL Project has released version 3.5.0 of its widely used open-source cryptographic library, introducing new features and notable changes that signal its evolution toward future-ready cryptography. This feature release includes support for post-quantum cryptography (PQC), server-side QUIC, and tighter control over TLS behavior.

\$

\$

\$ un

OpenSSH and PQC

The Shift to ML-KEM: In **OpenSSH 10.0**, the project designated `mlkem768x25519-sha256` as the default key exchange scheme. It pairs the NIST-standardized **ML-KEM-768** lattice-based algorithm with the traditional X25519 elliptic curve

OpenSSH Post-Quantum Cryptography

OpenSSH supports a number of cryptographic key agreement algorithms considered to be safe against attacks from quantum computers. We recommend that all SSH connections use these algorithms.

OpenSSH has offered post-quantum key agreement (*KexAlgorithms*) by default since release 9.0 (April 2022), initially via the `sntrup761x25519-sha512` algorithm. More recently, in OpenSSH 9.9, we have added a second post-quantum key agreement `mlkem768x25519-sha256` and it was made the new default scheme in OpenSSH 10.0 (April 2025).

To encourage migration to these stronger algorithms, OpenSSH 10.1 will warn the user when a non post-quantum key agreement scheme is selected, with the following message:

```
** WARNING: connection is not using a post-quantum key exchange algorithm.  
** This session may be vulnerable to "store now, decrypt later" attacks.  
** The server may need to be upgraded. See https://openssh.com/pq.html
```

Linux and PQC

- Distros have started pulling in OpenSSL 3.5
- **RHEL 10 / 9.6+:** RHEL offers **integrated, out-of-the-box PQC capabilities**. RHEL 10 features a cryptographic policy that prefers hybrid ML-KEM key exchange for TLS and SSH
- **Amazon Linux 2027 (AL2027 Preview):** Enables post-quantum cryptography by default in its system-wide cryptographic policies, utilizing updated libraries like OpenSSL 3.5, GnuTLS 3.8.10, and NSS 3.124 for hybrid key

What's new in post-quantum cryptography in RHEL 10.1

February 4, 2026 | 4-minute read | [Security](#)

Post-quantum cryptography in TLS

TLS can use post-quantum cryptography in two places: Key exchange, where it protects against "harvest now, decrypt later" attacks, and signatures, preventing machine-in-the-middle attacks that use quantum computers. With the release of RHEL 10.1, applications that use OpenSSL, GnuTLS, NSS, or the Go programming language have support for post-quantum key exchange enabled by default.

The OpenSSL, GnuTLS, and NSS cryptography libraries additionally support signatures and TLS certificates with Module-Lattice-Based Digital Signature Algorithm (ML-DSA), a NIST-standardized PQC algorithm. Note that no public Certificate Authorities (CAs) currently offer post-quantum cryptography, but private CAs or self-signed certificates [can be created with ML-DSA](#) today.

Post-Quantum Cryptography (PQC) on AL2027

The system-wide cryptographic policies on AL2027 enable post-quantum cryptography (PQC) by default. Most users will not need to change this default configuration.

AL2027 includes the following updated libraries and applications with PQC support:

- **OpenSSH** enables Post-Quantum Cryptography (PQC) key exchange by default.
- **OpenSSL 3.5** added support for the ML-KEM hybrid key exchange algorithm, and ML-DSA (ML-DSA-44, ML-DSA-65, and ML-DSA-87) and SLH-DSA signature algorithms.
- **GnuTLS 3.8.10** supports ML-KEM hybrid key exchange algorithms and ML-DSA-44, ML-DSA-65, and ML-DSA-87 signature algorithms for TLS communications.
- **NSS 3.124** supports the ML-KEM hybrid key exchange algorithm and ML-DSA-44, ML-DSA-65, and ML-DSA-87 signature algorithms for TLS communications.

Journey to Quantum Safe

Starting the quantum-safe journey

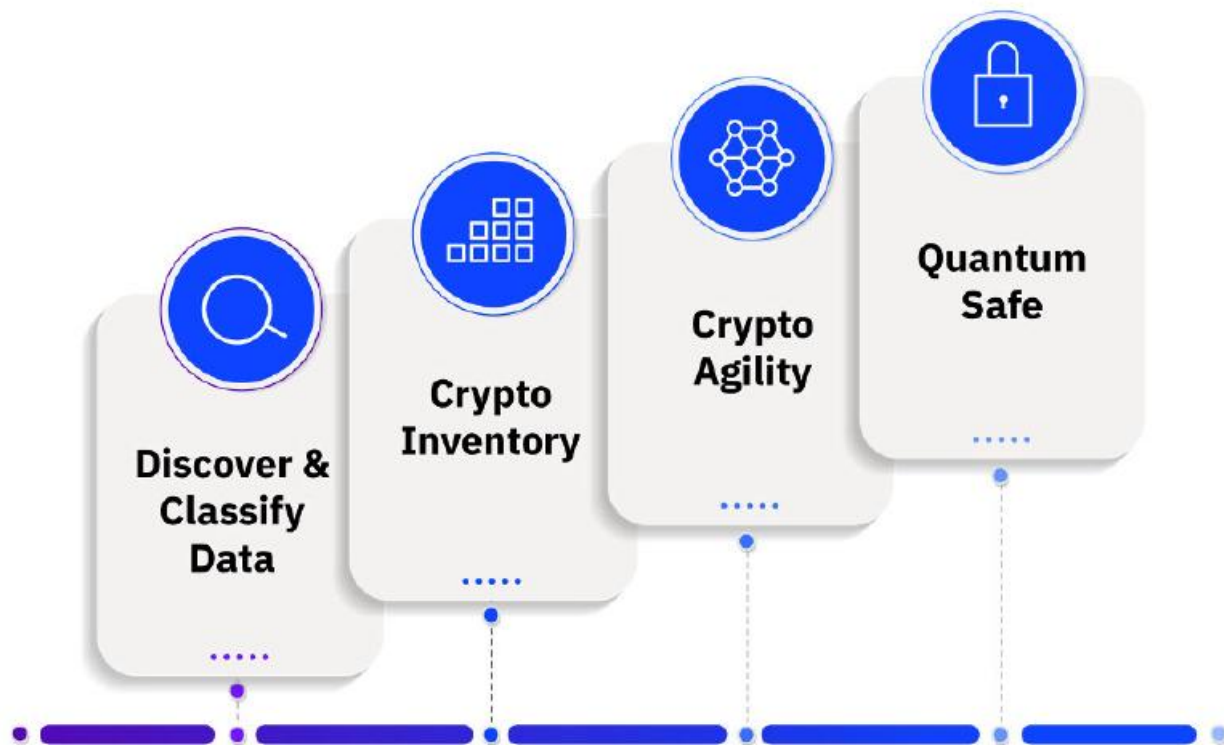
- Several steps must be taken along the journey
- Each organization has different cryptography use cases and usage constraints.
- Although no one-size-fits-all solution exists, general steps and guidance can be beneficial.
- Quantum-safety will become a compliance issue.

Consider your cryptography use in four broad areas:

- Infrastructure
- Applications
- Data protection
- Key Management



High Level Milestones



Milestones towards quantum-safety



Where are you
TODAY ?

*Are you involved in
discovering crypto or
mitigating issues in the
crypto environment?*

Discover & Classify Data

- Classify the value of your data and identify your “Crown Jewels”
- Identify locations of your data
- Understand your compliance requirements
- Create and manage your data inventory with defined ownership

Crypto Inventory

- Identify how your data is encrypted
- Create your cryptography inventory (containing certificates, encryption protocols, algorithms, key lengths, etc.)
- Manage your cryptography inventory and the lifecycle of certificates, encryption keys, etc.

Crypto Agility

- Define and implement processes to update / replace cryptography with well-defined lead-times
- Take all dimensions of crypto agility into account
- Test your crypto agility

Quantum-Safe

- Implement quantum-safe cryptography algorithms
- Understand the performance impact of quantum-safe crypto on the organization

Summary

Classical Cryptographic Algorithms are widely used to protect data and communications in computer systems and networks.

An adversary with access to a sufficiently strong quantum computer can break the classical algorithms we have used for many years.

Most vulnerable are Asymmetric Algorithms and Protocols.

Risks include theft of digital assets, forged documents, transactions, signatures, code and the like. **Secure communications are also in jeopardy.**

Researchers and standards bodies are moving to address the threats.

They are identifying new quantum-safe algorithms that can be used to protect classical and quantum computer workloads and data from the attacks that can be launched from quantum computers.

Organizations are providing migration guidance.

Industry migration guidance

IBM Redbook

Transitioning to Quantum-safe Crypto on IBM Z®

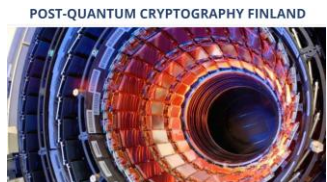
<https://www.redbooks.ibm.com/redpieces/abstracts/g248525.html>



PQC Finland

PQC Finland is national project aiming to develop, analyze and study Quantum-Safe algorithms and suitability to national security, products and practices.

<https://www.pqc.fi/index.php/home/>



National Cybersecurity Center of Excellence

Migration to post-quantum cryptography

<https://www.nccoe.nist.gov/projects/building-blocks/post-quantum-cryptography>



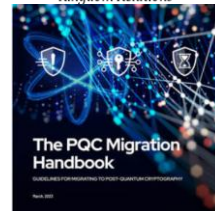
NETHERLANDS NATIONAL COMMUNICATIONS SECURITY AGENCY

PQC Migration Handbook

<https://english.aivd.nl/publications/publications/2023/04/04/the-pqc-migration-handbook>



General Intelligence and Security Service
Ministry of the Interior and Kingdom Relations



Electronic and Telecommunication Standards Institute

Migration strategies and recommendations to Quantum Safe schemes

https://www.etsi.org/deliver/etsi_tr/103600_103699/103619/01.01.01_60/tr_103619v010101p.pdf

ETSI TR 103 619 V1.1.1 (2020-07)



Thank You