Post-Quantum Cryptography Alliance Overview for CCC

July 25, 2024

Hart Montgomery, Linux Foundation

THE LINUX FOUNDATION

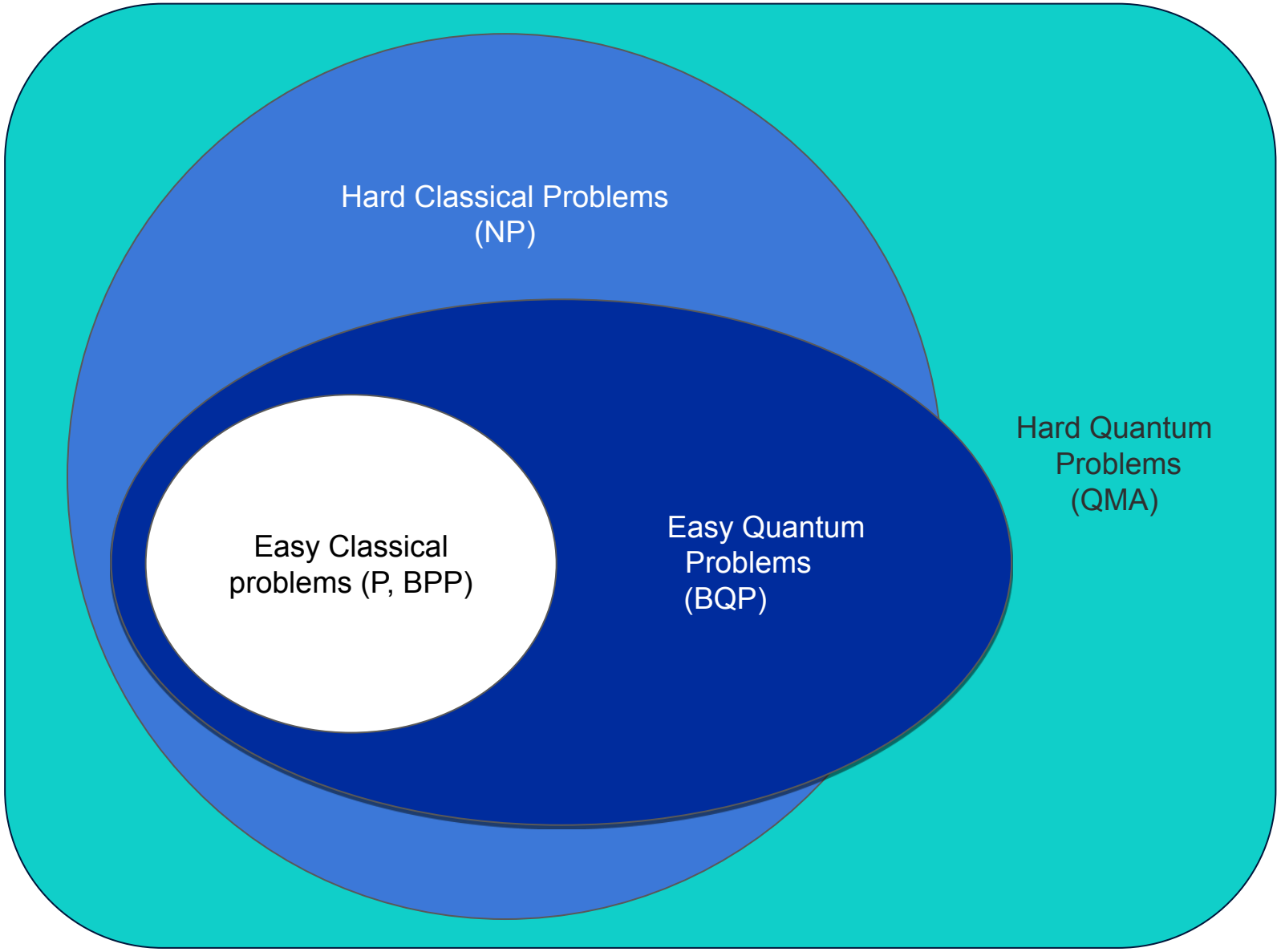
Talk Outline

1. The Quantum Cryptography threat
2. What is the Post-Quantum Cryptography Alliance?
3. Project details and how to get involved

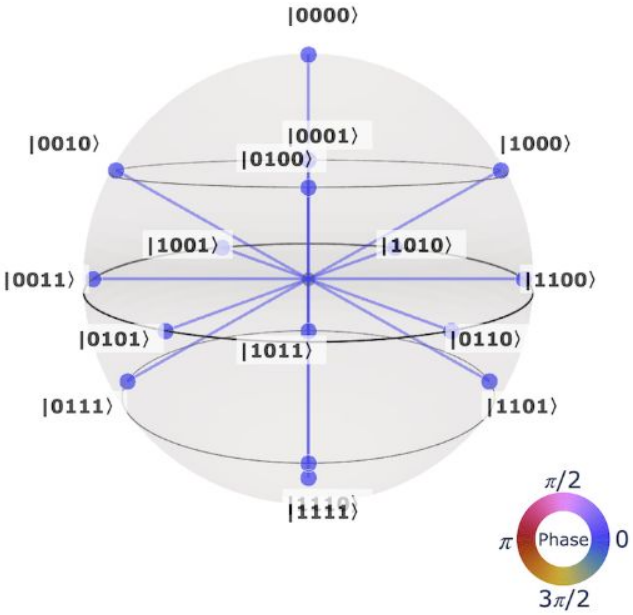
The Quantum Cryptography Threat

Sufficiently large enough quantum computers can break all “traditional” public-key cryptography!

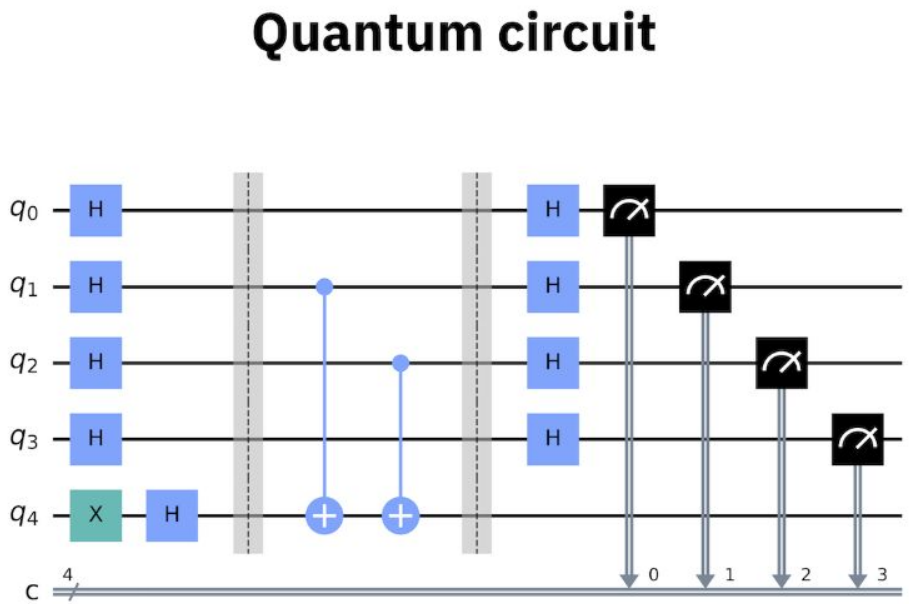
Why quantum?


Hard Classical Problems
(NP)
Easy Classical
problems (P, BPP)
Easy Quantum
Problems
(BQP)
Hard Quantum
Problems
(QMA)

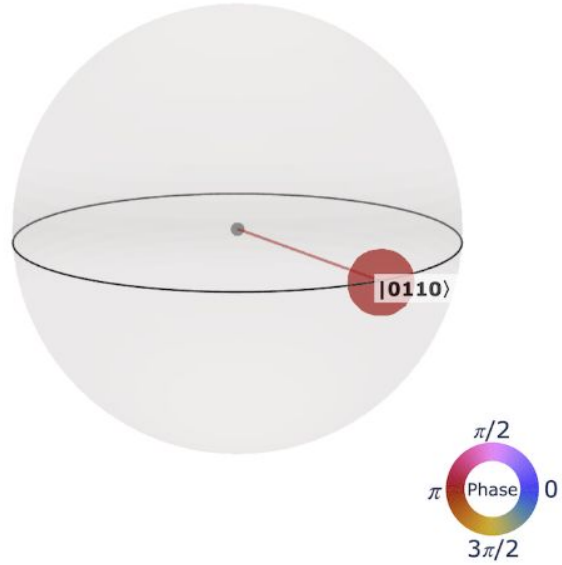
Quantum computers use qubits



Superposition of all possibilities

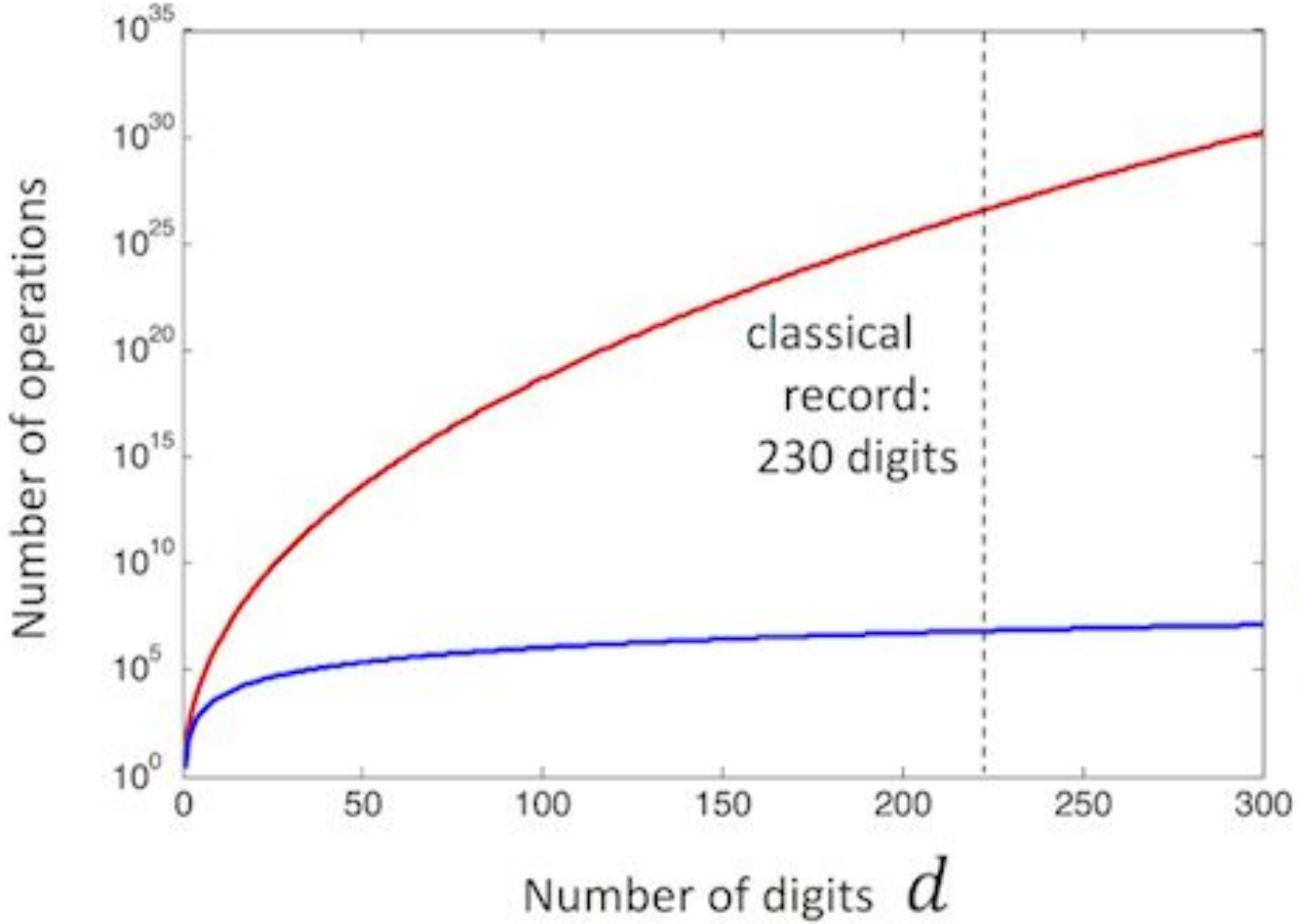


Computation driven interference



Solution

Ex: Shor's algorithm for factoring



$\exp(\text{const} \times d^{1/3})$

best classical
algorithm
(number field sieve)

$\text{const} \times d^3$

Shor's algorithm

Current cryptography is at risk

Prime factors

= p x q

2048-bit composite integer

```
25195908475657893494027183240048398571429282126204032027777137836043662020707595556264018525880784406918290641249515082189298559149176184502808489120072844992687392807287776735971418347270261896375014971824691165077613379859095700097330459748808428401797429100642458691817195118746121515172654632282216869987549182422433637259085141865462043576798423387184774447920739934236584823824281198163815010674810451660377306056201619676256133844143603833904414952634432190114657544454178424020924616515723350778707749817125772467962926386356373289912154831438167899885040445364023527381951378636564392120103971228221207203 57
```

Expected computation time

The most powerful computer **today:**

Millions of years

Shor's quantum algorithm:

Hours

Per Shor's algorithm, all public key crypto standards are vulnerable to attacks from large scale quantum computers

Public Key Encryption	RSA
Digital Signatures	DSA, ECDSA
Key Exchange Algorithms	Diffie-Hellman, ECDH

Our modern digital world depends on cryptography

It is the ultimate line of defense

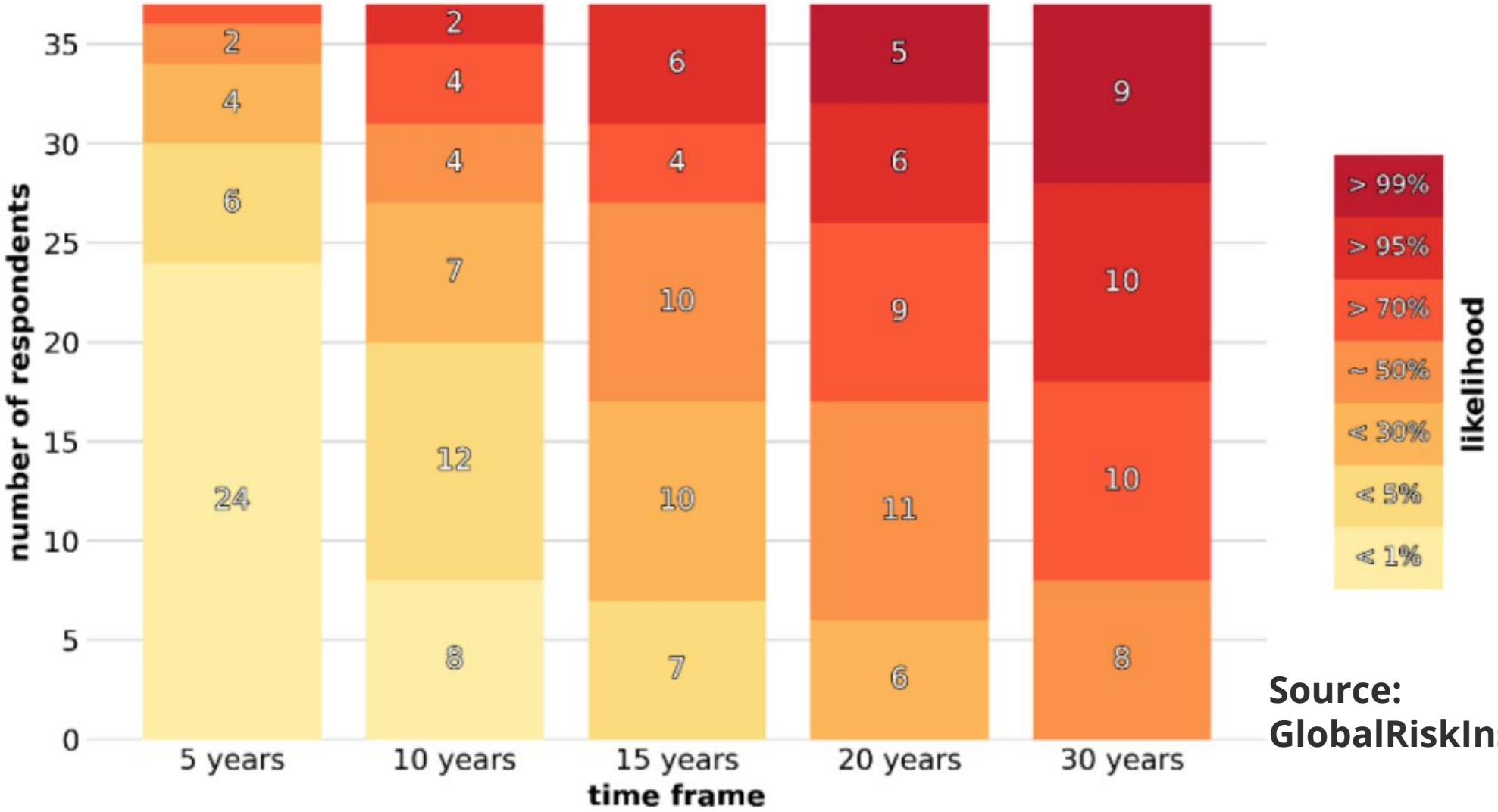
Trillions of Transactions on Billions of Devices use cryptography - including cellphones, laptops, desktops, services, ATMs, Internet Routers, VPN Servers, Smart IoT

What will a cybercriminal be able to do?

- Fraudulent authentication
- Forge digital signatures
- Harvest now, decrypt later

2023 EXPERTS' ESTIMATES OF LIKELIHOOD OF A QUANTUM COMPUTER ABLE TO BREAK RSA-2048 IN 24 HOURS

Number of experts who indicated a certain likelihood in each indicated timeframe



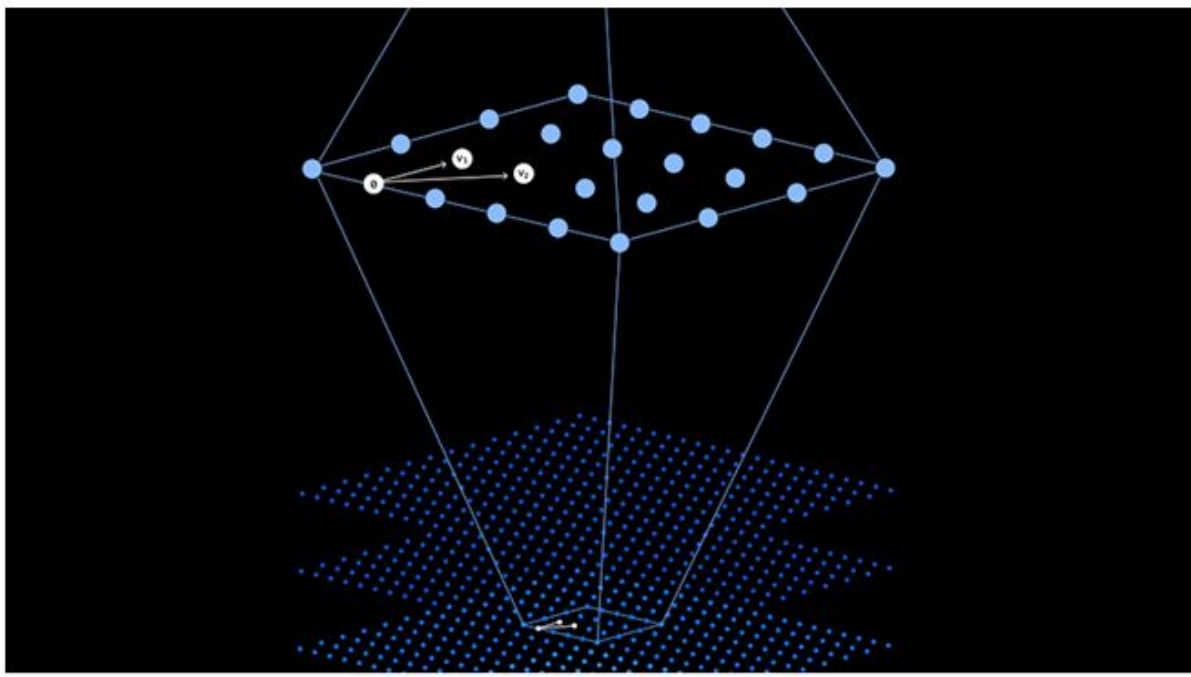
Source: GlobalRiskInsitute.org

Quantum Safe Cryptography

a.k.a. Post Quantum Cryptography or Quantum Resistant Cryptography

Traditional public-key cryptography relies upon mathematical problems that are difficult to solve on classical computers.

Quantum-safe cryptography includes **a suite of algorithms and systems that are resistant to attacks by both classical and quantum computers.**

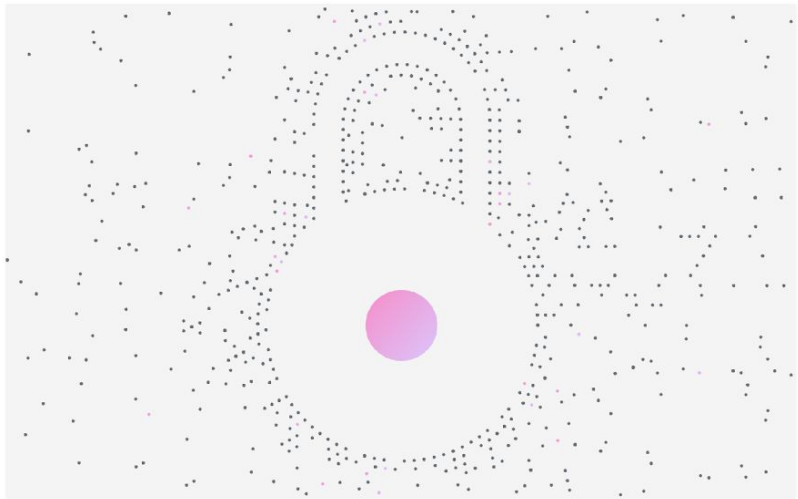


IBM Quantum **Learning** | Home | Catalog | Network | Composer | Lab

Practical introduction to quantum-safe cryptography

Review the basics of cryptography, and understand the challenges posed by new quantum algorithms, as well as how to mitigate the impact of that challenge through use of new quantum-safe encryption algorithms.

Start from beginning →



Lessons (0/7 complete) Expand all lessons

Introduction to this course

- Course introduction
- Key takeaways for this course
- Tips for navigating this course
- Before you begin
- Lesson structure
- Running the Python Examples
- Next steps after this course

Go to lesson →

Cryptographic hash functions

Symmetric key cryptography

Asymmetric key cryptography

Helpful materials

The following may be useful to you when reading through the course:

Pre-reading

This course contains a lot of detail on cryptography, and numerous maths examples which may use terminology and symbols you are not familiar with.

This Presentation offers:

- A primer on cryptography
- An introduction to some of the maths concepts used in this course

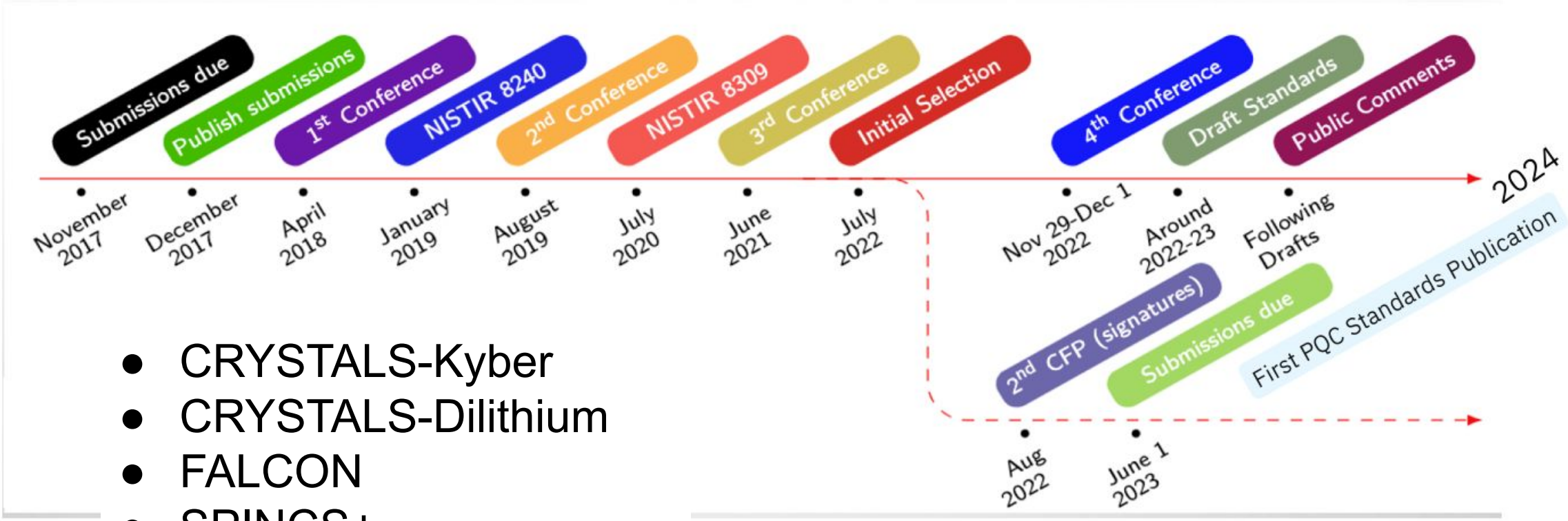
Course review

After completing the course you may find it useful to have a summary of what's been learnt as a take-away.

https://learning.quantum-computing.ibm.com/course/practical-introduction-to-quantum-safe-cryptography

National Institute of Standards and Technology (NIST)

Post Quantum Cryptography (PQC) Standardization Progress



- CRYSTALS-Kyber
- CRYSTALS-Dilithium
- FALCON
- SPINCS+

Source: NIST

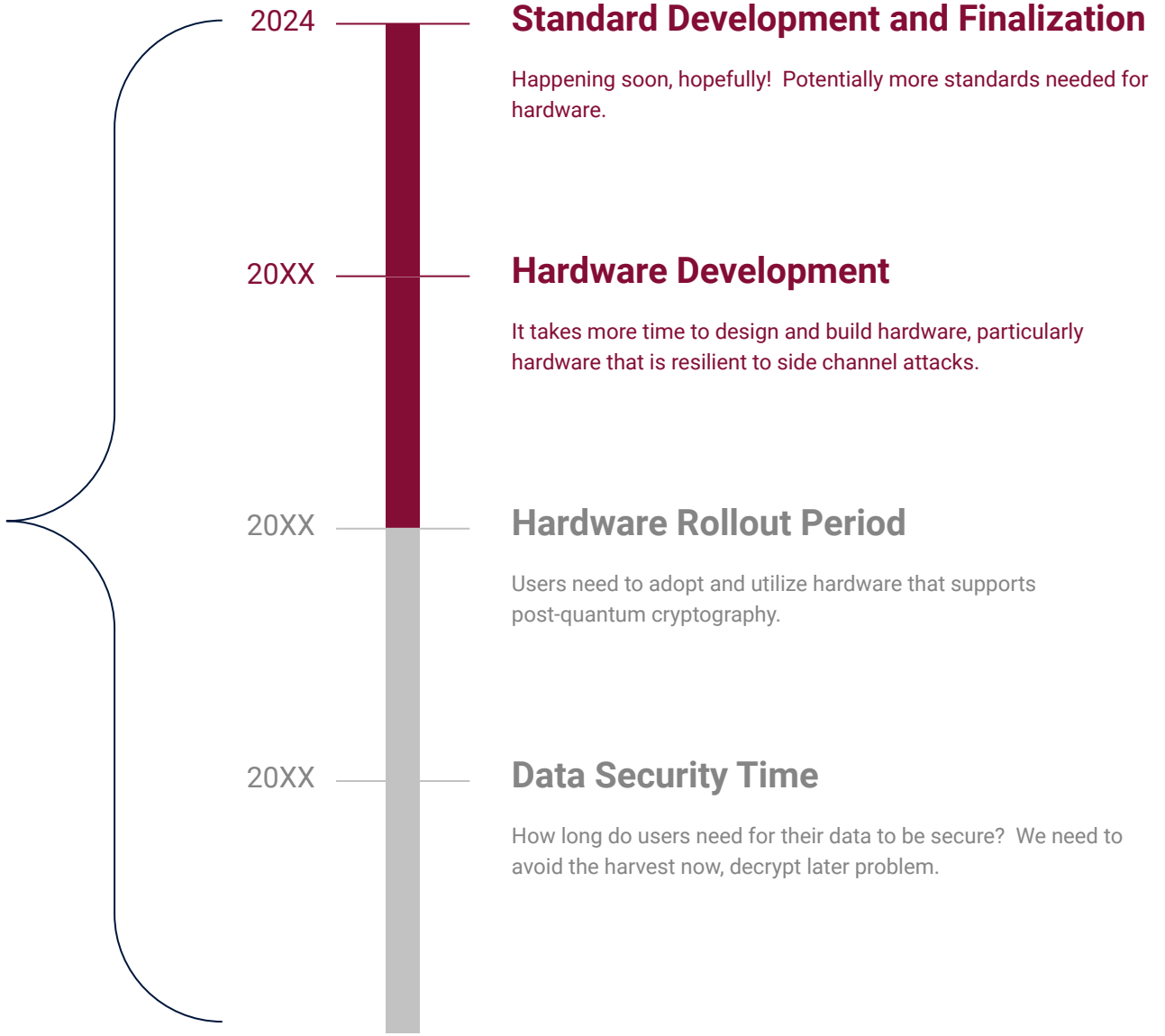
What Are the Issues?

Computation is not much more expensive with PQC than with traditional cryptography for basic cryptography.

Keys, ciphertexts, and signatures are much larger in size.

Hardware Timeline

Hopefully this is faster than powerful quantum computers arriving!



Introducing the PQCA

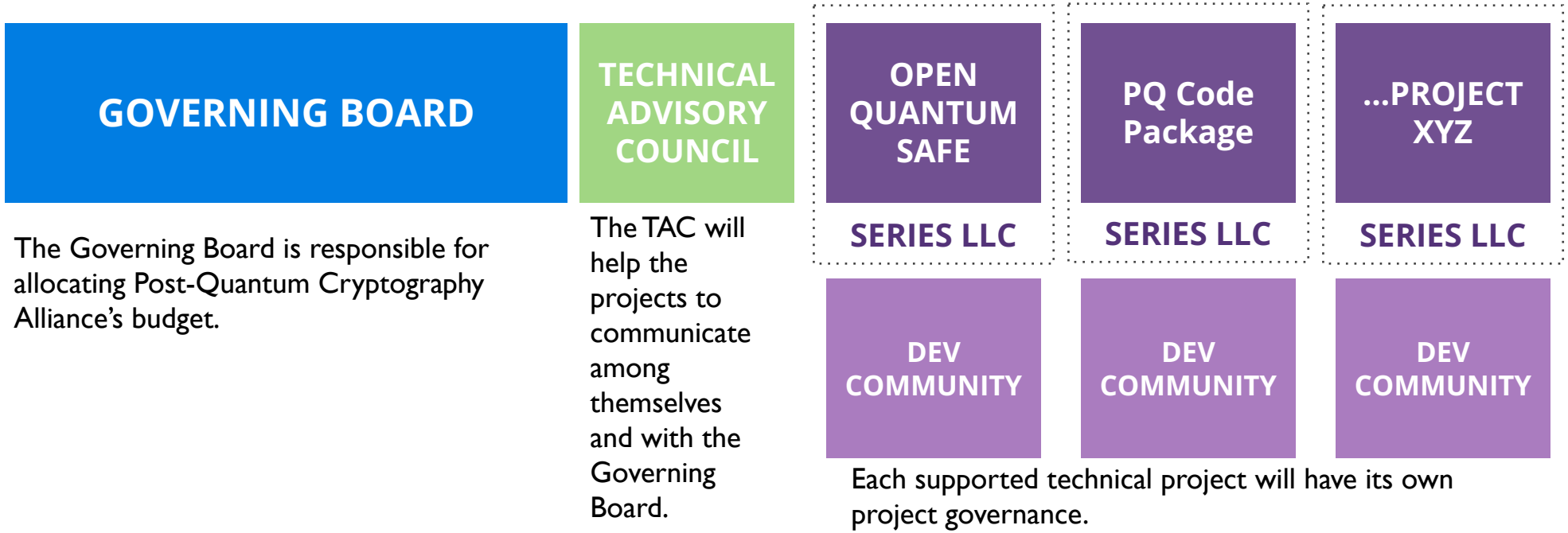
The Linux Foundation’s open source umbrella for post-quantum cryptography.

THE LINUX FOUNDATION

Post-Quantum Cryptography Alliance

To advance the adoption of post-quantum cryptography, by producing high-assurance software implementations of standardized algorithms, and supporting the continued development and standardization of new post-quantum algorithms with software for evaluation and prototyping.

Post-Quantum Cryptography Alliance Structure



The Governing Board is responsible for allocating Post-Quantum Cryptography Alliance's budget.

The TAC will help the projects to communicate among themselves and with the Governing Board.

Each supported technical project will have its own project governance.

Initial Projects Overview

Open Quantum Safe project

liboqs

Library of many PQ algorithms

- Main profile: standards-track algorithms
- Experimental profile: new algorithms, NIST signatures on-ramp etc.

OQS demos

Prototype integrations of PQ into protocols and applications to support experiments, standardization, interoperability

OQS OpenSSL 3 Provider

Integration of PQ + hybrid algorithms from liboqs into OpenSSL 3 via OpenSSL provider interface

- TLS key exchange, authentication
- X.509
- S/MIME, CMS, CMP

PQ Code Package

"Kyber" code package

High-assurance production source-code implementations of Kyber

- C, x86_64, ARMv8, …
- Rust, Go, …
- audited/certified/formally verified

Plus appropriate wrappers / providers, e.g. Kyber OpenSSL 3 provider

Potential Phase 2 projects

- Dilithium
- XMSS, LMS
- SPHINCS+
- Falcon (-> Phase 3?)

THE LINUX FOUNDATION

Production track: safe for use in production environments, with external audits or certification

Experimental track: primarily for prototyping and experiments, mindful of potential production use

Project Tracks

Projects within the Post-Quantum Cryptography Alliance will be labelled as either “production track” or “experimental track”, which will inform goals, requirements, and process.

Production track: These projects intend to release software that is **safe for use in production environments**. These projects should establish and use best practice development and testing processes, and have clearly stated threat models and security processes. Formal verification is highly desirable but not necessarily required. Resources will be allocated for these projects to receive external audits. Depending on demand from project members and availability of resources, it may be a goal for these projects to be submitted for certification (e.g., FIPS).

Experimental track: These projects intend to release software that is **primarily for research, experimentation, and prototype**, and is not intended for use in production environments. Mindful that some users may use still use them in production environments, these projects should make a reasonable effort to achieve good quality, including best effort development processes, automated testing, and clearly stated threat models.

Open Quantum Safe project

Goal: Provide an open-source software for evaluating and using post-quantum cryptography.

Contributed by the University of Waterloo–you will find **many familiar faces** from the cryptographic community!

Main components:

- **liboqs:** A cryptographic library in C providing implementations of standards-track post-quantum signature schemes and public key encryption / KEM schemes, plus support for testing and evaluating new experimental PQ algorithms.
- **OQS Provider:** An OpenSSL 3 provider adding post-quantum algorithms from liboqs into OpenSSL 3-based applications, providing support for post-quantum and hybrid TLS, X.509, and S/MIME/CMS.
- **OQS Demos:** Provide prototype integrations of post-quantum algorithms into protocols and applications, to support experiments and interoperability by early adopters and assist in protocol-level standardization.

Open Quantum Safe project: who's involved?

liboqs

- Project lead: Douglas Stebila
- liboqs core team
  - Amazon Web Services
  - Cisco
  - IBM Research
  - Microsoft Research
  - Sandbox AQ
  - University of Waterloo
  - NVIDIA
  - Individual contributors: Michael Baentsch, Thom Wiggers, Vlad Gheorgiu (softwareQ)
- Algorithm submission teams and upstream algorithm implementation maintainers

OQS demos

- Contributors:
  - Michael Baentsch
  - IBM Research
  - University of Waterloo

OQS OpenSSL 3 Provider

- Project lead: Michael Baentsch
- Contributors:
  - Cisco
  - IBM Research
  - University of Waterloo

PQ Code Package

Goal: Develop and maintain high-assurance, production implementations of Kyber for a variety of target architectures (ARMv7, ARMv8, x86_64, ...) and languages (C, Rust, Go, ...), distributed primarily as source code. Aim for implementations to be audited and/or formally verified.

Project track: production

Intended audience: Implementers of cryptographic libraries and tools that need to add Kyber to their software as source code. The Kyber code package should be organized in a way that allows for easy tracking of changes and integration into software development lifecycles.

Starting points:

- Portable C (PQCrystals, PQClean, eventually HACL*)
- x86_64 AVX2 (libjade / Jasmin)
- x86_64 non-AVX2 (libjade / Jasmin)
- ARM64 (neon-ntt?)
- ARM32 (pqm4, eventually libjade)
- Rust
- Go
- OpenSSL 3 provider for Kyber

Key participants at launch:

- Still in progress!

Phase 2 Projects

There are multiple standards-track post-quantum algorithms beyond Kyber which would benefit from having a home for high-assurance production quality implementations. The Post-Quantum Cryptography Alliance would be a natural place for those as well. But mindful of potentially overextending the scope during the initial launch, it may be preferable to view some of these as a "phase 2" project starting perhaps a year later, or as resources solidify.

- Dilithium
- XMSS
- LMS
- SPHINCS+
- Falcon

Your project here!

Goal: Complete PQC Stack

Cryptographic Agility Providers/APIs

Libraries

Base Level Implementations

Formal Verification Libraries

CBOM Generators

We need to make it easy to use and switch to PQC, or people won’t do it.

Technical Project Governance

Project Lifecycle Policy

We have structured Post-Quantum Cryptography Alliance as an **umbrella project, able to support numerous separately-organized technical projects**. The group within Post-Quantum Cryptography Alliance that will assist the coordination of and communication across the various technical projects is the Technical Advisory Council, or TAC. The TAC will own and maintain a published Project Lifecycle Policy that describes how new projects are onboarded to, and project within, Post-Quantum Cryptography Alliance.

Sample Technical Project Charter

Individual technical projects are organized with their own governance, and an umbrella project can support all forms of technical collaboration from open source, open data, open specifications and other types of open projects. We have included above a sample governance template for how we typically structure governance for open source projects within an umbrella.

How to Get Involved

Discord

Github

Mailing Lists

Membership

Legal Notices

- The Linux Foundation, The Linux Foundation logos, and other marks that may be used herein are owned by The Linux Foundation or its affiliated entities, and are subject to The Linux Foundation's Trademark Usage Policy at https://www.linuxfoundation.org/trademark-usage, as may be modified from time to time.
- Linux is a registered trademark of Linus Torvalds. Please see the Linux Mark Institute's trademark usage page at https://lmi.linuxfoundation.org for details regarding use of this trademark.
- Some marks that may be used herein are owned by projects operating as separately incorporated entities managed by The Linux Foundation, and have their own trademarks, policies and usage guidelines.
- TWITTER, TWEET, RETWEET and the Twitter logo are trademarks of Twitter, Inc. or its affiliates.
- Facebook and the "f" logo are trademarks of Facebook or its affiliates.
- LinkedIn, the LinkedIn logo, the IN logo and InMail are registered trademarks or trademarks of LinkedIn Corporation and its affiliates in the United States and/or other countries.
- YouTube and the YouTube icon are trademarks of YouTube or its affiliates.
- All other trademarks are the property of their respective owners. Use of such marks herein does not represent affiliation with or authorization, sponsorship or approval by such owners unless otherwise expressly specified.
- The Linux Foundation is subject to other policies, including without limitation its Privacy Policy at https://www.linuxfoundation.org/privacy and its Antitrust Policy at https://www.linuxfoundation.org/antitrust-policy. each as may be modified from time to time. More information about The Linux Foundation's policies is available at https://www.linuxfoundation.org.
- Please email legal@linuxfoundation.org with any questions about The Linux Foundation's policies or the notices set forth on this slide.

THE LINUX FOUNDATION