

Isogeny-Based Cryptography Made Easy

Sikhar Patranabis, PhD

Staff Research Scientist, IBM Research India

sikhar.patranabis@ibm.com

<https://research.ibm.com/people/sikhar-patranabis>

Need for Cryptography

Our digital infrastructure depends on cryptography. It is the building block for creating secure systems.

Confidentiality

Enigma, Signal, and WhatsApp messages are **encrypted** for message [confidentiality](#).

Personal data in banking or healthcare is **encrypted** for [privacy](#).

Nonrepudiation

Card payments are cryptographically **signed** to [prevent fraud](#) by customers, merchants, or third parties.

Digital ledgers (blockchains, CBDCs) are cryptographically **signed** to [prevent fraud](#) (double spending).

Integrity

Software/firmware updates for your electric vehicle, mobile phone, and server are cryptographically **signed** to [prevent malware](#).

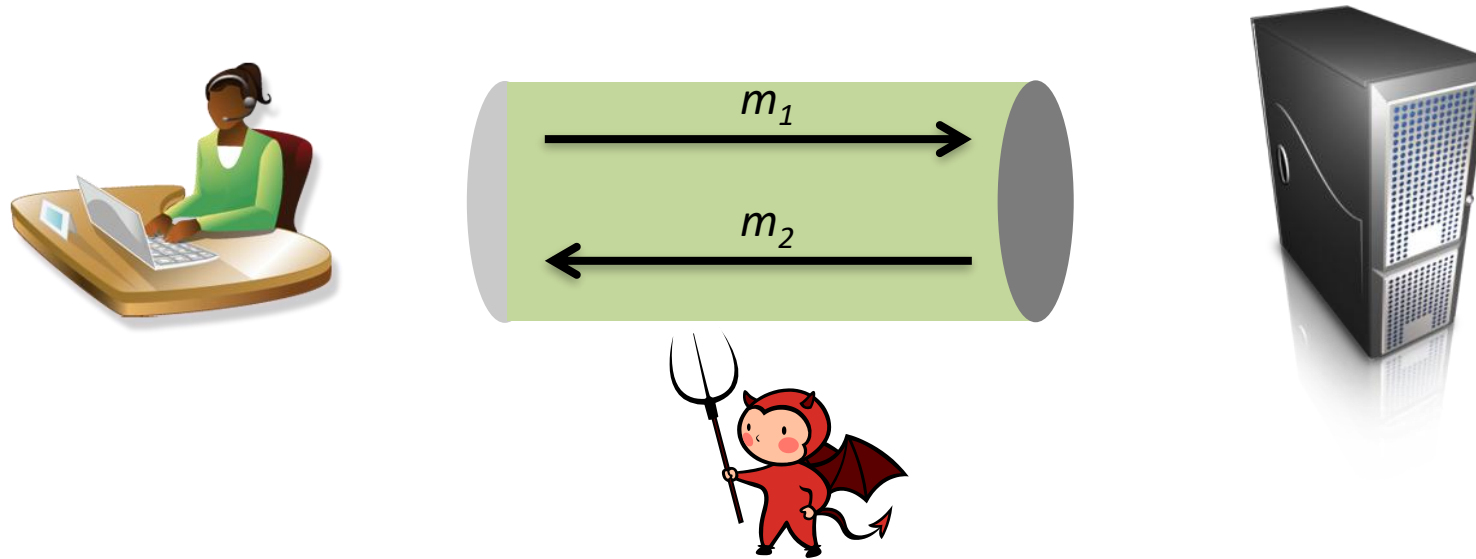
Documents are cryptographically **signed** to protect their [integrity](#) and prove the identity of the signees.

Authentication

Digital certificates are used to **authenticate** to systems.

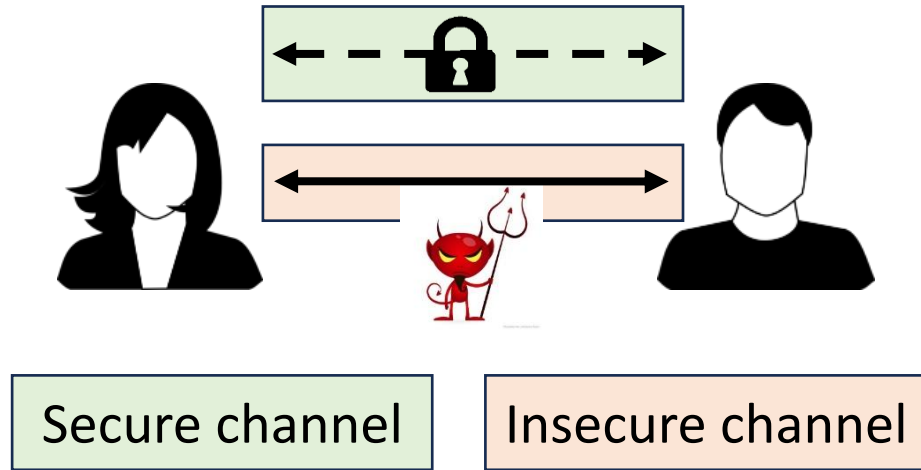
Passports are cryptographically **signed** to authenticate the issuer and [prevent forgery](#).

Basic Premise: Confidential and Authenticated Communication



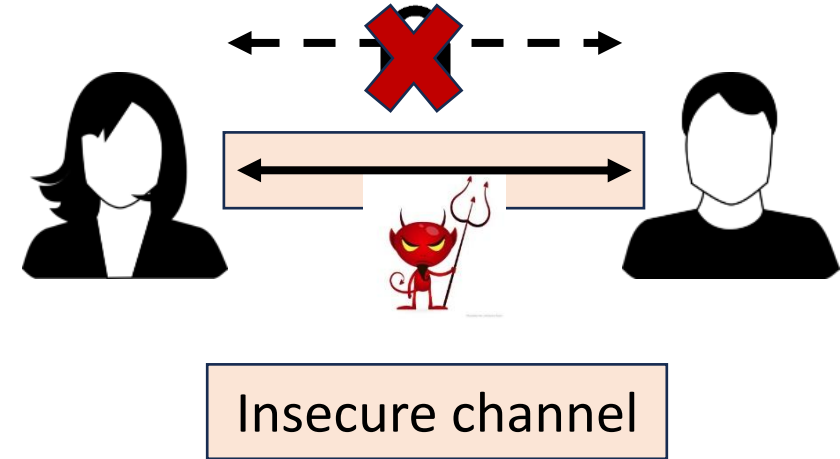
- Two entities exchange messages over an **insecure channel**.
- The insecure channel will be provided by a communications network.
 - Examples: wireless LAN, telecom network, “the Internet”
- **Exchanged messages should be confidential and authenticated**
 - **How? Use cryptography to build a secure channel on top of the insecure channel**

Symmetric-key vs Public-key Cryptography



Symmetric-Key Cryptography

Assume a “costly” secure channel



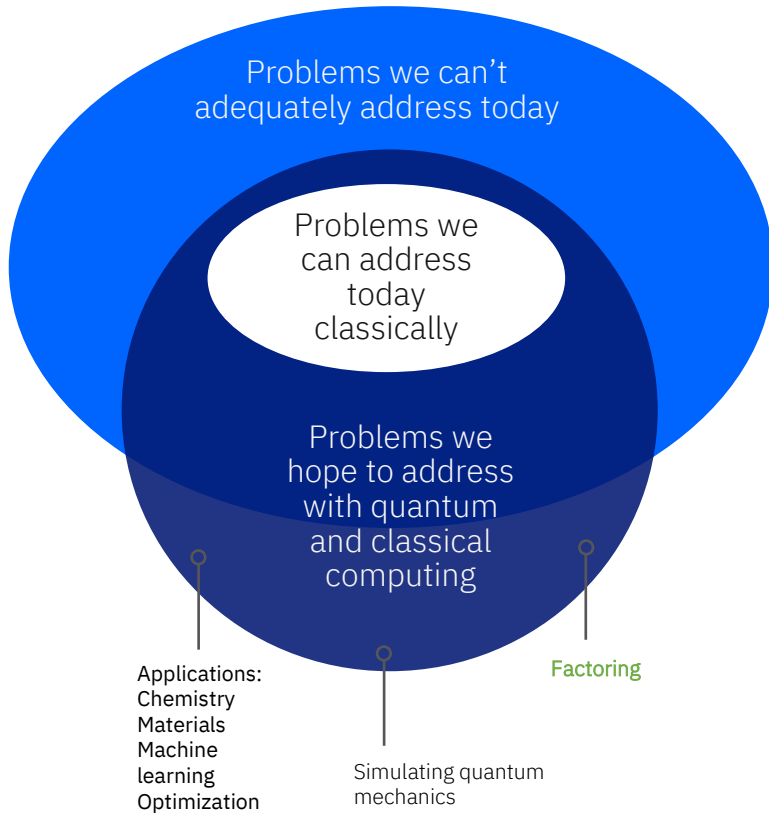
Public-Key Cryptography

Do not assume a secure channel

- **Public-Key Cryptography: helps to realize the costly secure channel**
- Symmetric-Key Cryptography: use secure channel for communication over insecure channel

Quantum Threats to Public-key Cryptography

- Power of Quantum Computers



- Public-key cryptography relies on hard mathematical problems.

Factorization: Find prime factors

25195908475657893494027183240048398571429282126204032027771378360436
62020707595556264018528807844069182906412495150821892985591491761845
0280848912007284499268793280728776735971418347703618963750149718246
91165077613379859095700097330459748808428401974291006424586918171951
187461215151726546322822168699875491824224336372590851418654620435767
98423387184774479207399342365848238242811961538150106748104516603773
060562016196762561338441436038339044149526344321901146575444541784240
209246165157233507787077498171257724679629263863563732899121548314381
6789988504044536402352738195137863656439212010397122822120720357

= ? · ?

Expected computation time

Most powerful computers today: millions of years

Shor's quantum algorithm: few hours

Decade-old cryptography broken

Public key encryption • Digital signatures
Key exchange algorithms

RSA • DSA • ECC • ECDSA • DH

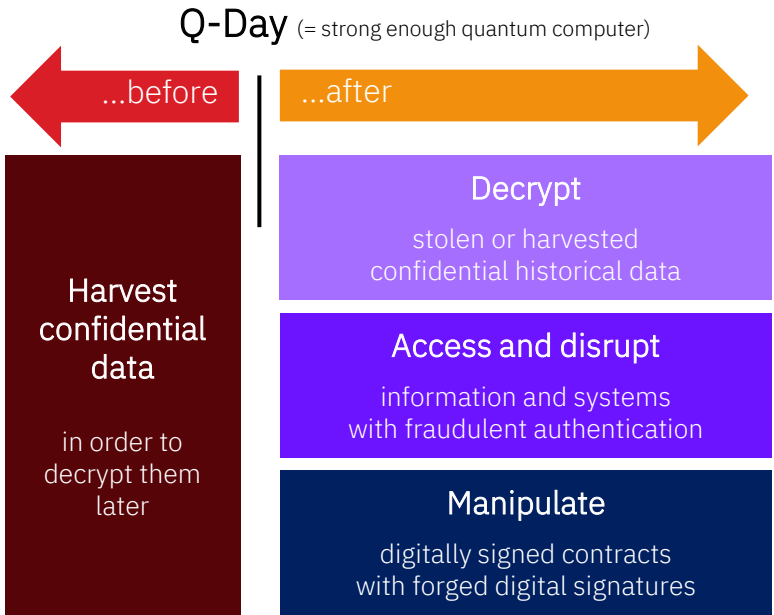
- Till recently, **Shor's algorithm** was the best-known quantum algorithm for factoring
- But recently, **breakthrough improvements** to Shor's algorithms have been made, which reduce the number of qubits significantly
- Confluence of improved quantum algorithms and quantum hardware/computing platforms **might shorten the timeline** for cryptographically relevant quantum computers

Why Quantum Threats to Public-Key Cryptography Matter

“The National Institute of Standards and Technology predicts it may be possible to break 2048-bit RSA by 2030”

- NIST report on Post Quantum Cryptography

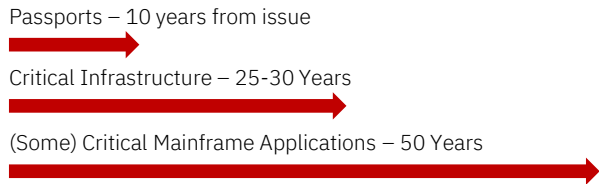
What will attackers do?



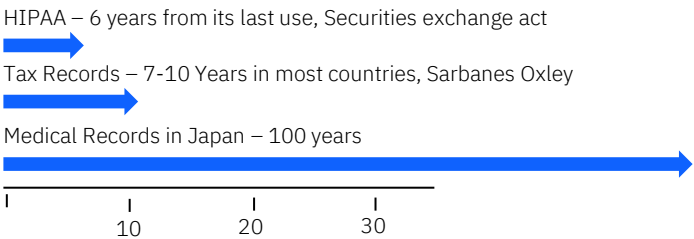
What is at risk?

Internet	HTTP/TLS, SSH, DNS, VPNs, ...
Digital Signatures	eIDAS, PAdES (PDF), AES ...
Critical Infrastructure	Control systems, car systems, ...
Financial Systems	Payment & Settlement Systems, ...
Blockchain	Wallets, Transactions, Authentication, ...
Enterprise	PGP E-mail, IAM, PKI, App signatures, ...

Infrastructure replacement cycles



Security Time Value of Data



Quantum-Safe Cryptography (QSC)

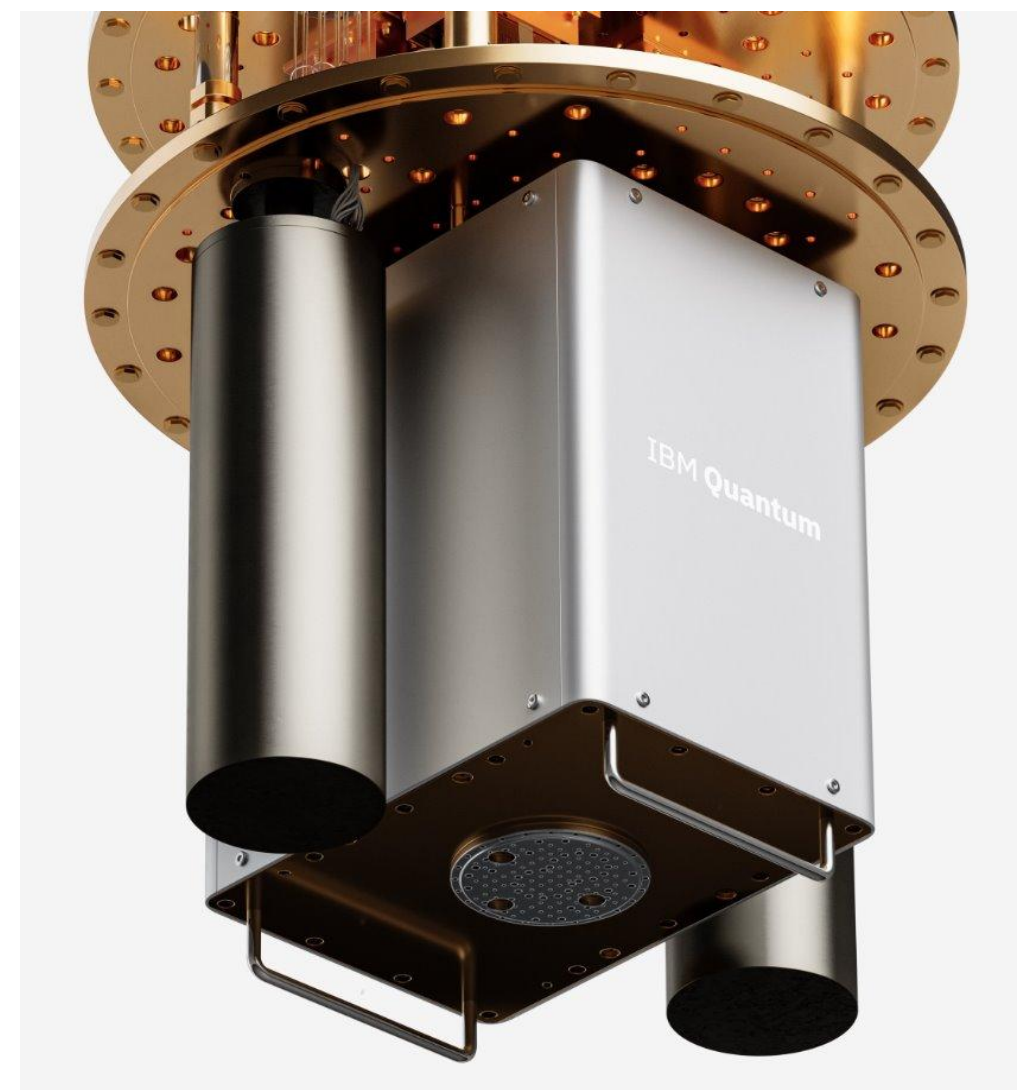
Or Post-Quantum Cryptography (PQC)

QSC refers to **public-key cryptography** that is:

- resistant to attacks by **BOTH** classical and quantum computers
- but runs on classical computers

IBM's QSC Mission

Securing the world's digital infrastructure for the era of quantum computing.



Solution Components for QSC

Rebuild the cryptographic foundations of our digital world for the quantum era

PQC algorithms

New cryptography based on a variety of new mathematical techniques

Resistant to classical and quantum attacks

Standardization

Standardization of PQC for key encapsulation and digital signature started in 2016

NIST Standards (FIPS 203, 204, 205, 206, 207) announced in 2024-2026

Crypto discovery

Where do you use cryptography?

Cryptographic inventory, or cryptography bill of materials (CBOM)

Migration

Software development

Vendor products

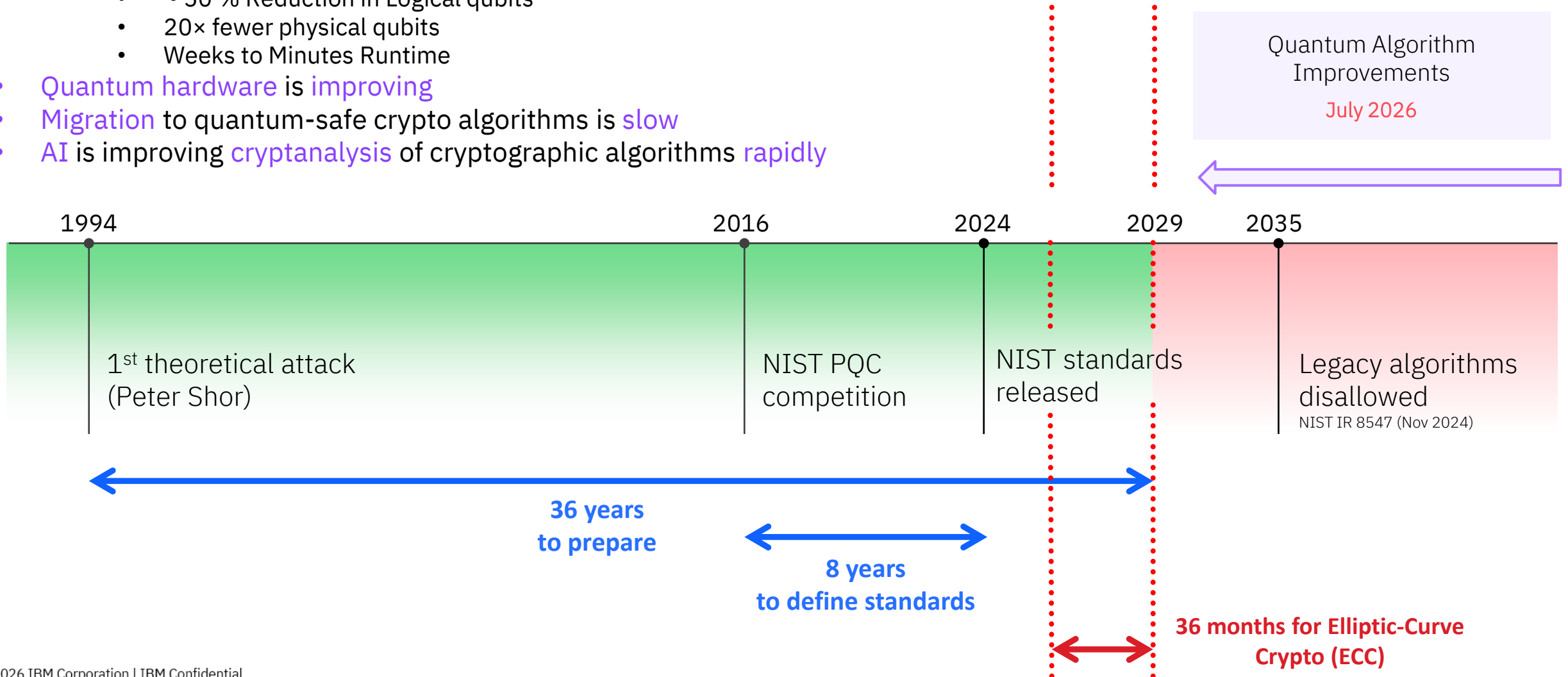
Software as a service

Infrastructure, network, devices

Cryptographic agility

QSC Migration and the Shrinking Timeline

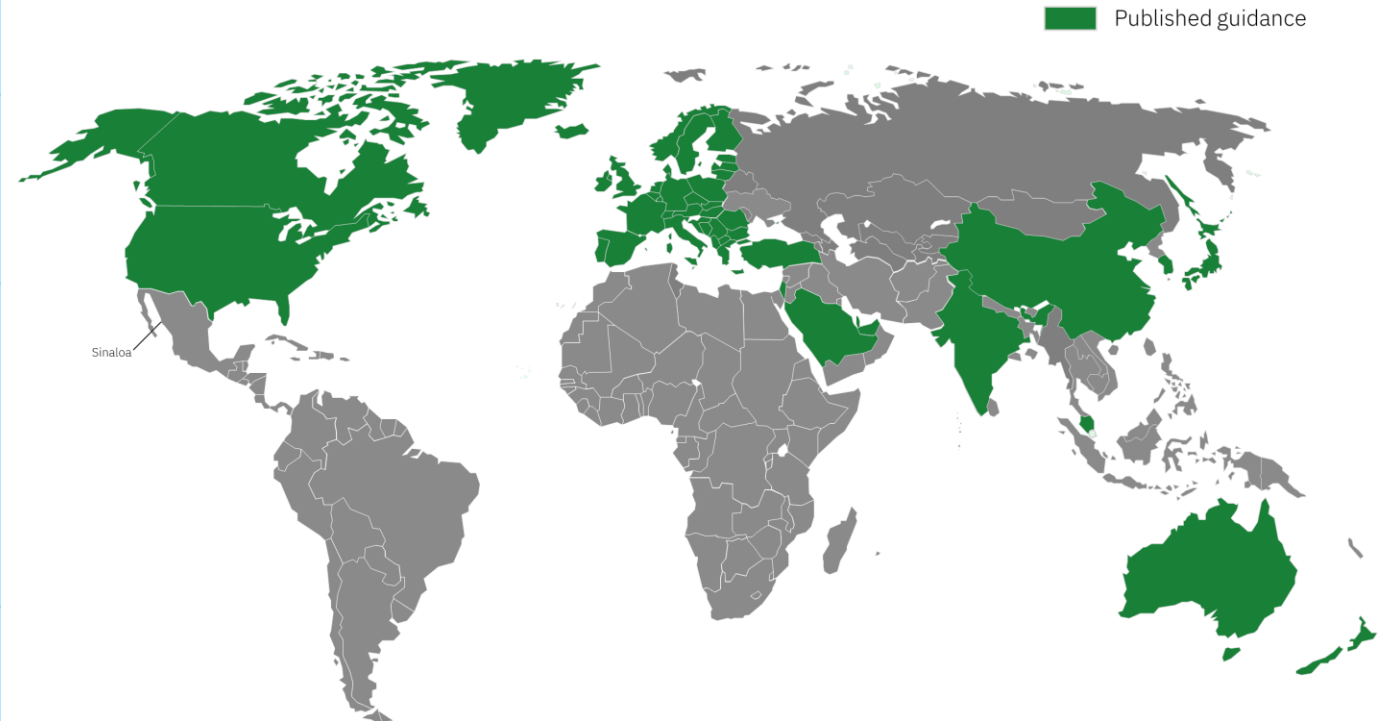
- Quantum algorithms are improving
 - Improvements to Shor's algorithm (2017-2026):
 - ~1,500× Gate Reduction
 - ~ 50 % Reduction in Logical qubits
 - 20× fewer physical qubits
 - Weeks to Minutes Runtime
- Quantum hardware is improving
- Migration to quantum-safe crypto algorithms is slow
- AI is improving cryptanalysis of cryptographic algorithms rapidly



QSC Global Government Initiatives Tracker (March 2026)



	Action
Updated Guidance in 2025 (12)	Australia, Canada, China, Czechia, Israel, Japan, Netherlands, New Zealand, Singapore, South Korea, United Kingdom, United States
New Guidance (9 published + unpublished)	Belgium, Estonia, India, Ireland, Hong Kong, Lithuania, Malaysia, Saudi Arabia, United Arab Emirates
Sectoral	3GPP, Bank for International Settlements, Cross Markets Operational Resilience Group, CFDIR, Europol Quantum-Safe Financial Forum, FS-ISAC, G7 Cyber Experts Group, Mastercard,
Conferences	ETSI Sec (Oct), EU PQC (Dec), FCA (Oct), NCF Convening to Act (Mar), NCSC (Dec), NIST-5 (Apr), NIST-6 (Sept), PKI-C PQC (Jan), PKI-C PQC (Oct), PQCrypto (Apr)



Current timelines are being updated in many countries because of Google paper on improved quantum attacks on ECC

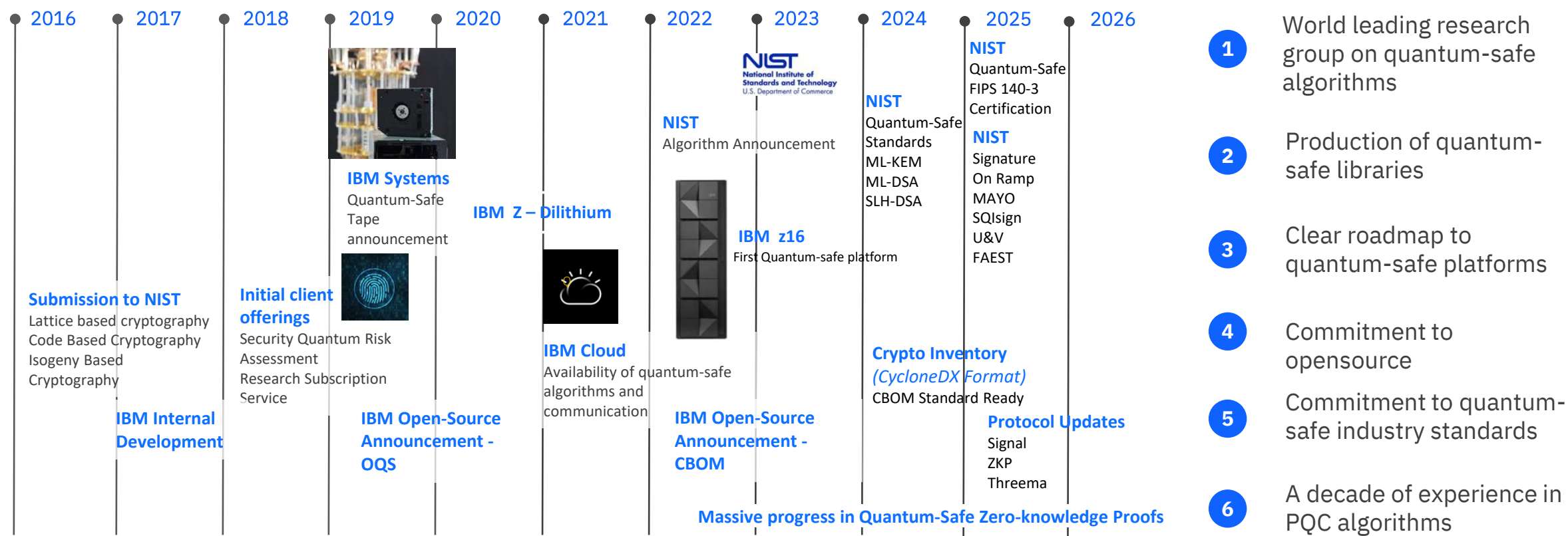
Rest of the Talk

- IBM's perspective on QSC
- Isogeny-based QSC: an overview
- Accessible abstractions for Isogeny-based QSC

Rest of the Talk

- IBM's perspective on QSC
- Isogeny-based QSC: an overview
- Accessible abstractions for Isogeny-based QSC

IBM has led the development of QSC globally



NIST QSC Standardization (2016-2026)

Main project: 2016 – 2026

82 submissions, 4 rounds

Result: Standardization of 2 key-encapsulation mechanisms, and 3 digital signature algorithms

Key Encapsulation Mechanisms (KEMs):

- CRYSTALS – KYBER (ML-KEM, lattice-based)
- Hamming Quasi-Cyclic (HQC-KEM, code-based)

Digital Signatures:

- CRYSTALS-Dilithium (ML-DSA, lattice-based)
- FALCON (FN-DSA, lattice-based)
- SPHINCS+ (SLH-DSA, based on MPC + hashing)

4 out of the 5 standards have been
co-designed by IBM researchers

**Digital signatures:
New call for Submissions**

Goal: to identify compact quantum-safe digital signatures based on non-lattice assumptions

Round 1 Submissions in June 2023.

Round 3 announced May 2026.

4 IBM Research submissions currently in round 2:

- SQIsign (based on isogenies)
- MAYO (based on multivariate quadratics)
- UOV (based on multivariate quadratics)
- FAEST (based on MPC + hashing)

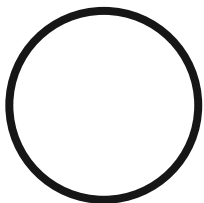
4 out of the 9 round-3 candidates have been
co-designed by IBM researchers

Our technology-driven approach to become quantum-safe and achieve crypto-agility

Protecting today. Preparing for tomorrow.

IBM Software Products

- IBM Guardium Cryptography Manager
- IBM Quantum-Safe Explorer
- IBM Quantum-Safe Remediator
- IBM Vault (HashiCorp)



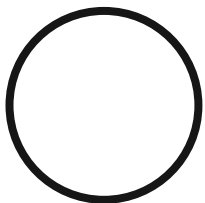
Discover and
Inventory



Identify cryptographic objects
and IT assets

Detect unknown or shadow
cryptography

Map dependencies,
ownership, and usage



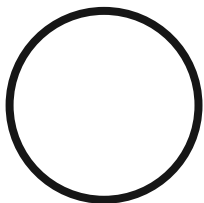
Assess and
Comply



Assess post-quantum
cryptography risk

Flag outdated or vulnerable
algorithms

Generate audit-ready reports



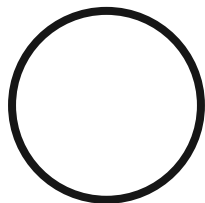
Manage
Lifecycle



Centralize crypto lifecycle
operations in one tool

Automate key generation and
certificate renewal

Enforce consistent policies
across the organization



Protect and
Adapt



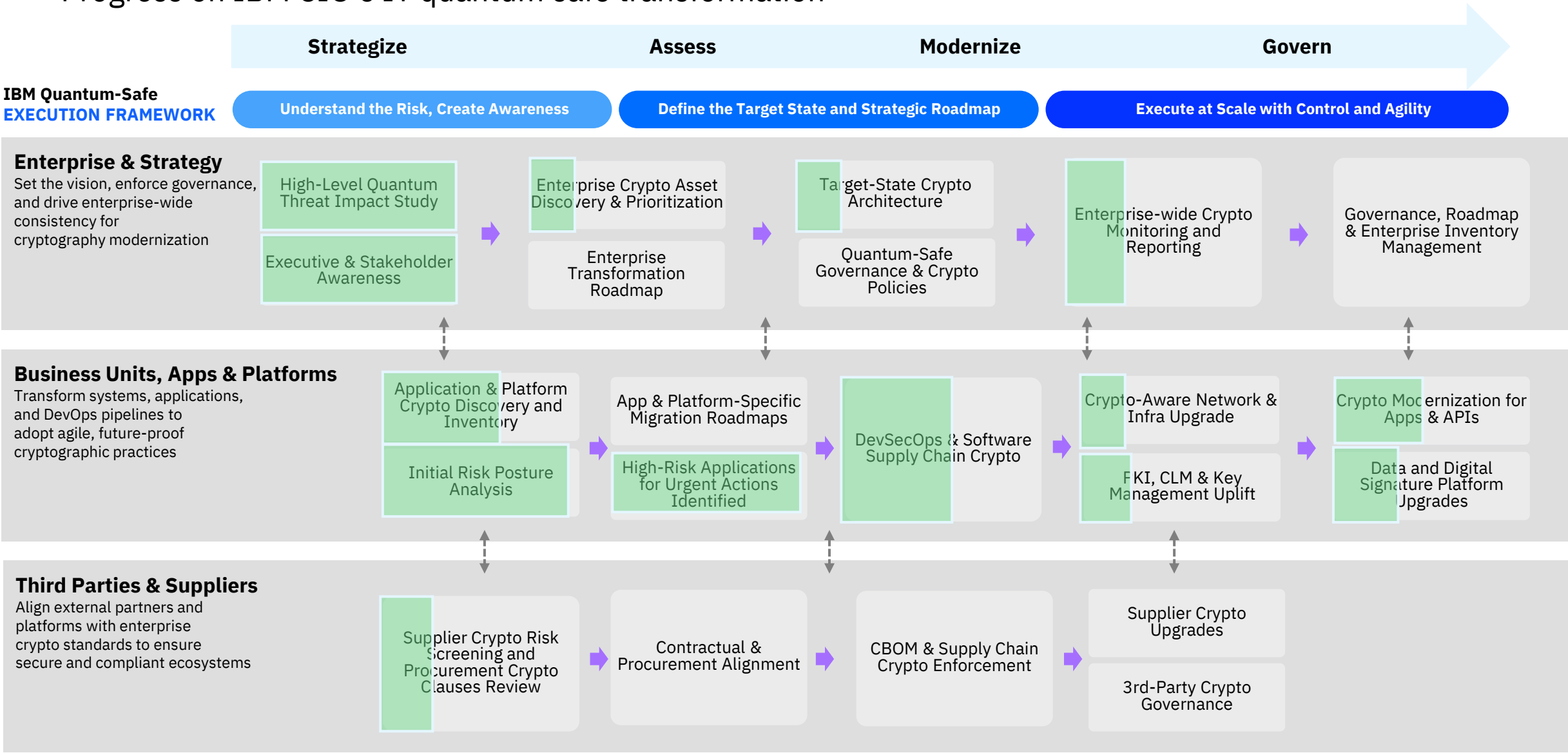
Utilize encryption key mgmt. to
protect sensitive data

Assess the efficiency and
feasibility of PQC algorithms

Protect against “harvest now,
decrypt later” attacks

A Cohesive, Enterprise-to-Ecosystem Plan is needed for Quantum-Safe Success

Progress on IBM CIO's IT quantum safe transformation



QSC Migration

Insights

QSC migration is not about changing algorithms, it's mostly about:

- The right risk management and governance
 - Updating or replacing components, products, and infrastructure
 - Reconfiguring them to use new algorithms
-

The availability of updates or replacements will depend on :

- Protocols and exchange format standards being ready
- Suppliers being ready
- Budget, skills and other dependencies being in place

What needs changing is governed by [dependency chains](#) – understand these first.

- Start with critical business assets and how they are composed
 - Understand which external data/services they rely on and how they are secured
-

The most critical steps :

- Updates to procurement procedures
- Supplier roadmaps in a normalized form (CBOM)
- Clear definitions and position on Hybrid & Agility
- Plan for evolving availability of supply chain components

QSC Migration

Technical Debt

Hardware & HSMs

Firmware, TPM, secure enclaves

OS & Runtime

OpenSSL, BoringSSL, NSS — distro-pinned versions

Language Runtime

Java JCA/JCE, .NET CNG, Python cryptography lib

Framework / Platform

Spring, Quarkus, .NET, Node.js — version-gated PQC

Application Layer

Custom crypto integrations, legacy auth flows

QS Algorithms

ML-KEM / ML-DSA / SLH-DSA (FIPS 203/204/205)



PQC migration is not about swapping algorithms — we need to migrate each of the layers described above

 **Java / JVM**

Java 8–11 (70%+ enterprise)

→ Java 27+ required for Hybrid ML-KEM ([JEP 527](#))

 **.NET**

Framework 4.x / .NET 5–6

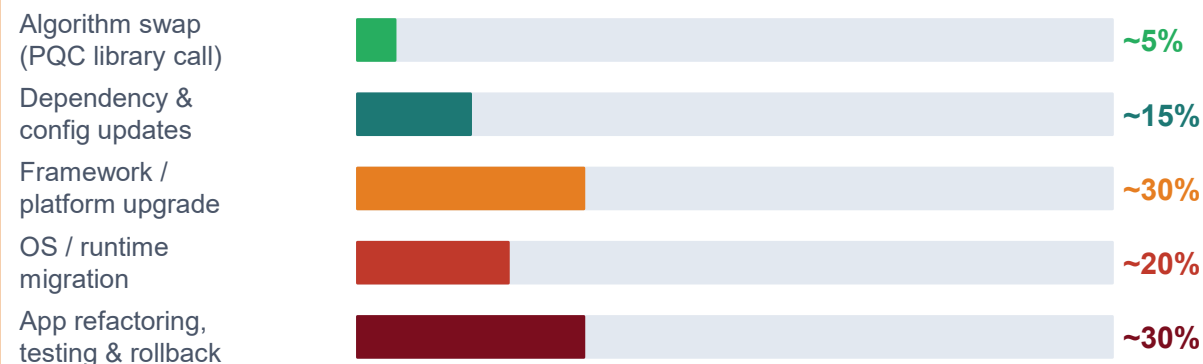
→ .NET 8+ required for PQC support (Microsoft PQC preview)

 **OpenSSL / TLS**

OpenSSL 1.1.x (EOL, pervasive)

→ OpenSSL 3.4+ for ML-KEM/DSA; distros lag 12–36 months

WHERE THE EFFORT ACTUALLY GOES



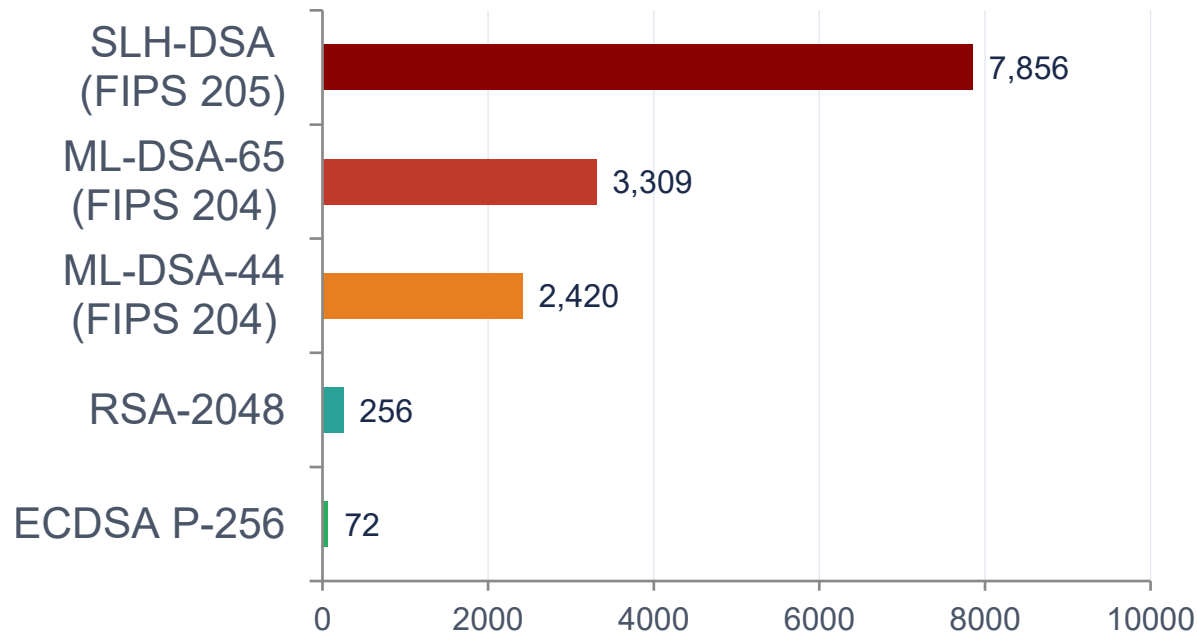
For a system on Java 8 or .NET 4.x, the PQC migration project is primarily a platform modernisation project — the algorithm change is the smallest part of the work.

Effort estimates are illustrative. Actual distribution varies by system age, architecture, and existing test coverage.

QSC Migration

Performance Impact

Signature sizes: Classical vs PQC
(in bytes)



The Problem

- ⚠️ QS signatures are 30–100× larger than classical equivalents
- ⚠️ X.509 certificate chains multiply this overhead at every hop
- ⚠️ TLS handshakes balloon — impacting latency and bandwidth
- ⚠️ High-frequency reporting connections in financial infrastructure are especially exposed

⚠️ **Planning implication: Simply swapping algorithm libraries is not sufficient in many protocol contexts — the underlying certificate architecture must also evolve.**

PQC Migration

Mitigating Performance Impact through Research and Standardization

- NIST has initiated a new standardization process for “on-ramp” quantum-safe digital signatures
- IBM continues to lead the cutting-edge global research efforts driving this process

Goals of the new NIST Process:

- **Diversification.** Most of the current NIST PQ standards are based on lattices. NIST wants some algorithms based on different assumptions as a back-up
- **Improve signature size and verification speed.** There is a significant gap in key sizes and signature sizes between quantum-safe and classical crypto for digital signatures
 - Reducing this gap could make the transition to quantum-safe easier for some applications

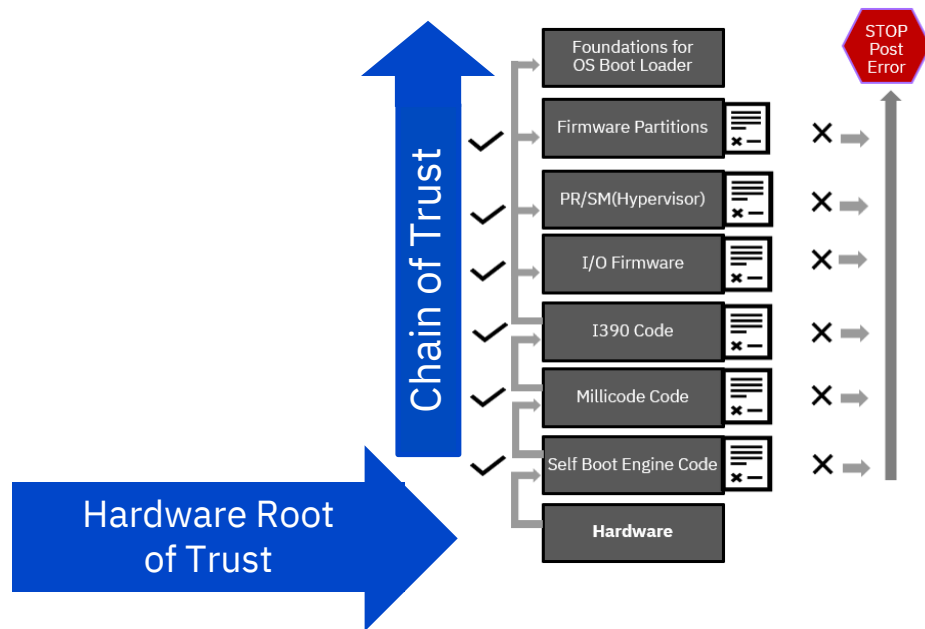
Timeline of the new NIST Process:

- 41 Round-1 submissions in June 2023; 9 Round-3 candidates announced in May 2026
- 4 out of 4 Round-1 IBM submissions have progressed to Round 3:
 - SQIsign (based on isogenies)
 - MAYO (based on multivariate quadratics)
 - UOV (based on multivariate quadratics)
 - FAEST (based on MPC + hashing)

Infrastructure for Quantum-Safe

Secure boot enhancements

- Secure boot protects system integrity by verifying firmware components during Initial Machine Load (IML)
- On z16/z17, Secure boot takes advantage of a dual signature scheme using ECC and CRYSTALS-Dilithium
- Provides two layers of protection which future proofs against attacks from unauthorized firmware (malware)



4770 HSM / CEX8S

Built for the Quantum-Safe future

- IBM Crypto Express8S (CEX8S) IBM z17 and IBM z16 includes implementation of CRYSTALS-Dilithium and CRYSTALS-Kyber quantum-safe algorithms
- (CEX7S) for IBM z16 and IBM z15™ includes CRYSTALS-Dilithium support
- Quantum-Safe protected boot loader using parallel signatures (ECDSA + CRYSTALS-Dilithium)



IBM's Impact on Standards and Community Updates



MITRE Organization
PQCA

Draft Cryptographic Inventory Paper



Linux Foundation
PQCA

Post-Quantum
Cryptography
Alliance Launched
(Feb 2024)



NIST NCCoE

NIST SP 1800-38B: Approach, Architecture, and Security Characteristics of Public Key Application Discovery Tools (Preliminary Draft)

NIST SP 1800-38C: Quantum-Resistant Cryptography Technology Interoperability and Performance Report (Preliminary Draft)



Cryptography Bill
of Material (CBOM)

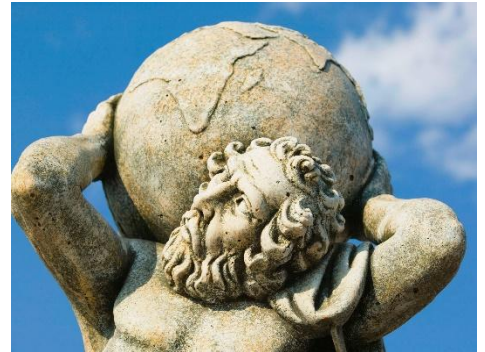
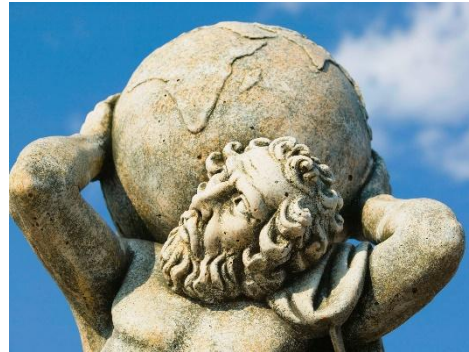
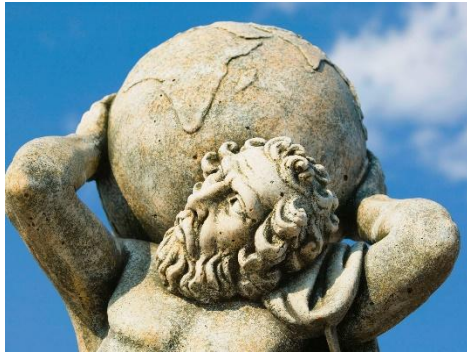
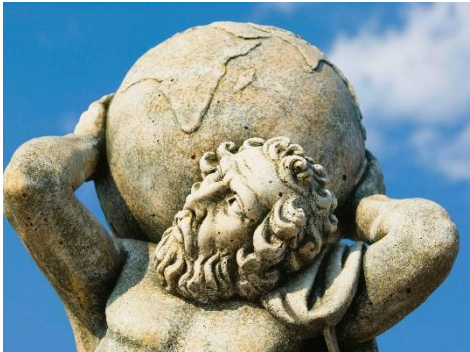
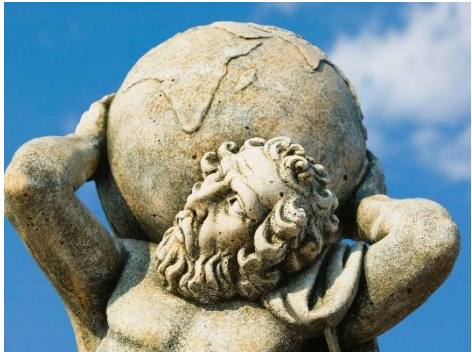
- **CycloneDX** xBOM version 1.6 (April 2024)
- **ECMA** xBOM Standard (July 2024)
- IBM QS Explorer, Microsoft/Santander, etc.
- SonarQube CBOM generator (June 2024)
- IBM CBOM Viewer (July 2024)

Rest of the Talk

- IBM's perspective on QSC
- Isogeny-based QSC: an overview
- Accessible abstractions for Isogeny-based QSC



Public-Key Cryptography



Discrete Log

Factorization

Lattices

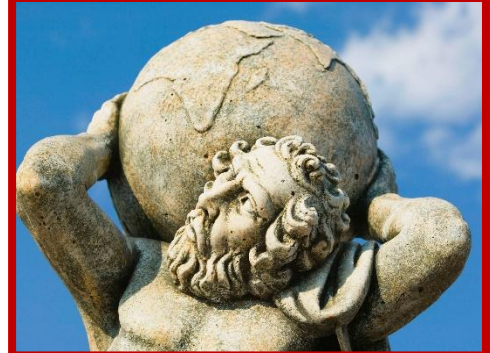
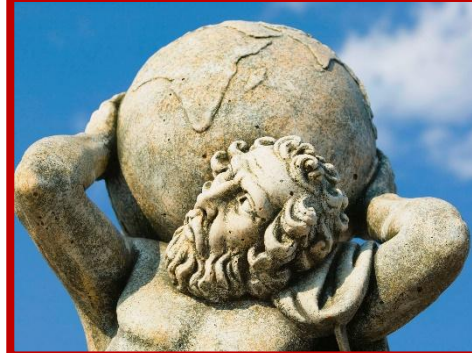
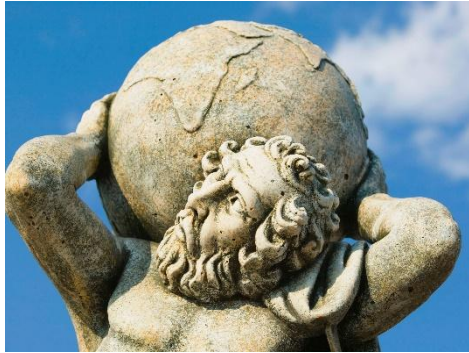
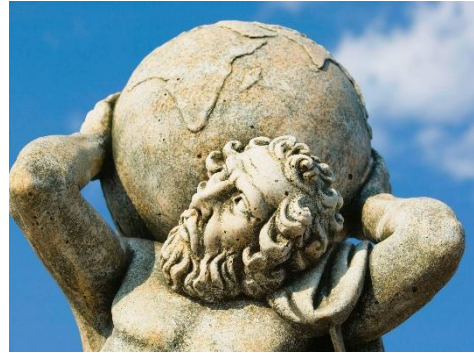
Codes

Isogenies

Shor's Algorithm



Public-Key Cryptography (in a quantum world)



Discrete Log

Factorization

Lattices

Codes

Isogenies

Isogeny-Based Cryptography: An Overview

KEM	Assumption	Key Size	Ciphertext Size
ML-KEM-512	Lattice-based	800 bytes	816 bytes
HQC-128	Code-based	2249 bytes	4545 bytes
dCTIDH-2048	Isogeny-based	256 bytes	304 bytes

Signature	Assumption	Key Size	Signature Size
ML-DSA-44	Lattice-based	1312 bytes	2420 bytes
FN-DSA-512	Lattice-based	897 bytes	666 bytes
SLH-DSA-128	Hash-based	32 bytes	7888 bytes
SQIsign-I	Isogeny-based	83 bytes	200 bytes

- **Isogenies are a promising quantum-safe/post-quantum alternative to lattices:**
 - Highly compact key sizes
 - Highly compact ciphertext sizes
 - Highly compact signature sizes
- **Relatively less understood:**
 - Attacks on SIDH and the SIDH-based NIST PQC finalist SIKE [CD23,MMPPW23,Rob23]
- **Nonetheless, a large portion of isogeny-based cryptography continue to survive (and even thrive):**
 - **CSIDH** and its advancements (such as dCTIDH) for public-key encryption/key-encapsulation (KEM)
 - **SQIsign** for digital signatures

SQIsign – The Most Compact Quantum-Safe Signature Today

Co-designers from IBM Research: Andrea Basso, Luca De Feo, Basil Hess, Sikhar Patranabis

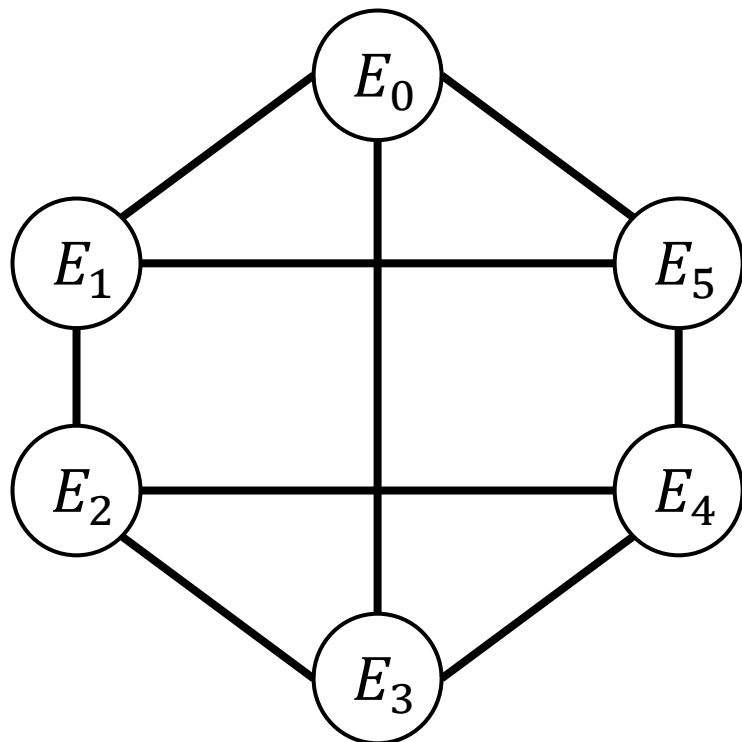
Digital Signatures		Type	Key size	Signature size
Quantum-safe (NIST standards)	Dilithium (ML-DSA-44)	Lattice-based	1312 bytes	2420 bytes
	Falcon (FN-DSA-512)		897 bytes	666 bytes
	Sphincs+ (SLH-DSA-128)	Hash + MPC-based	32 bytes	7888 bytes
Quantum-safe (NIST candidate)	SQIsign-I (Round-3)	Isogeny-based	82 bytes	200 bytes

- SQIsign is:
 - Nearly **3x smaller signature sizes** than **all** post-quantum NIST-standardized signatures
 - Nearly **10x smaller key sizes** than **all lattice-based** NIST-standardized signatures
- For more details, see:
 - Website: <https://sqisign.org/>
 - Latest specification: <https://sqisign.org/spec/sqisign-20250707.pdf>
 - Latest (stable) implementation: <https://github.com/SQIsign/the-sqisign/tree/nist-v2>
 - Latest security analysis: [ABDPW25, <https://eprint.iacr.org/2025/379>]

[ABDPW25] Marius A. Aardal, Andrea Basso, Luca De Feo, **Sikhar Patranabis**, Benjamin Wesolowski: A Complete Security Proof of SQIsign. **Crypto 2025** [eprint.iacr.org/2025/379]

Isogeny-Based Cryptography: An Example Embodiment

(p, ℓ) -Isogeny Graph

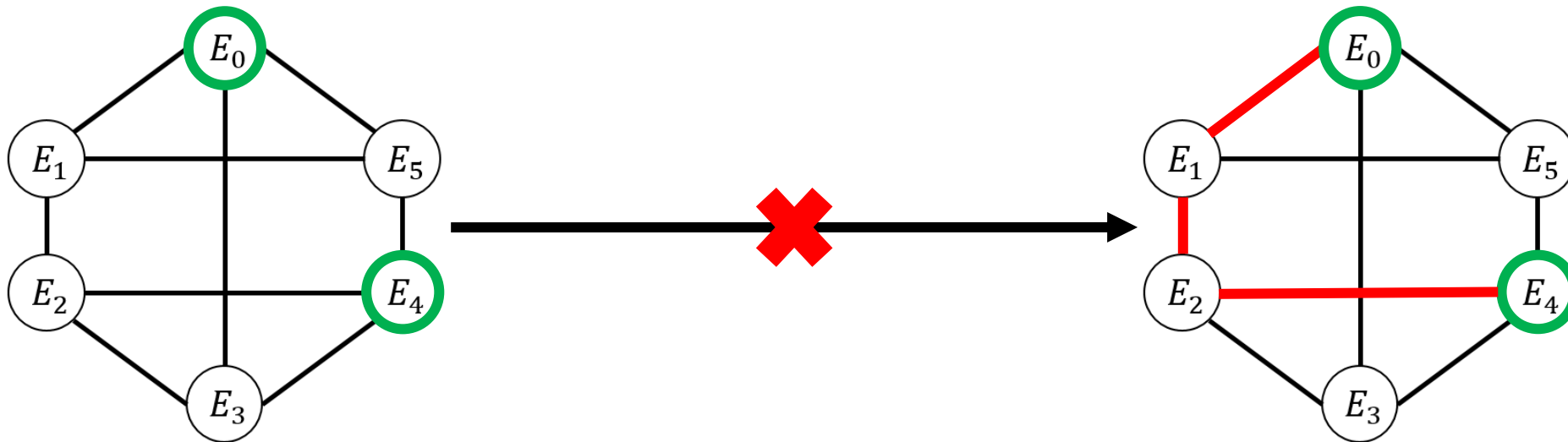


- A graph characterized by primes p and ℓ
- Vertices are (isomorphism classes of) supersingular elliptic curves over $\mathbb{F}_{p^2}$ ($\approx \frac{p+1}{12}$ vertices)
- Edges are isogeny maps between curves of degree ℓ
- Fully connected, $(\ell + 1)$ -regular, undirected
- Ramanujan (optimal expander graphs)
 - A random walk converges to a uniform vertex
 - Random walk of length $d \approx$ Uniform isogeny of degree ℓ^d

Isogeny-Based Cryptography: An Example Embodiment

Fundamental hard problem: Computing isogenies

Given uniformly randomly sampled supersingular E_0 and E_4 over $\mathbb{F}_{p^2}$, output $\varphi: E_0 \rightarrow E_4$



- **Equivalent formulation:** compute the endomorphism ring of a uniformly randomly sampled supersingular elliptic curve (say E_4)
- This formulation is called the **supersingular endomorphism ring problem**
- **Trapdoor information:** given the endomorphism rings of E_0 and E_4 , it is easy to compute $\varphi: E_0 \rightarrow E_4$

The landscape of isogeny-based cryptography remains limited due to the very high entry bar for non-experts



Rest of the Talk

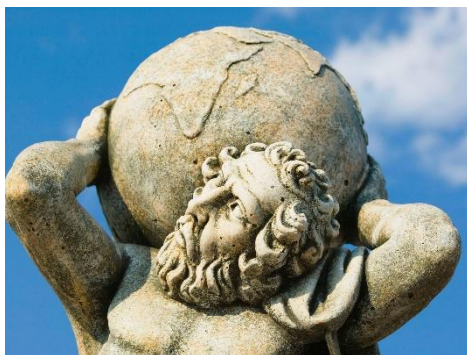
- IBM's perspective on QSC
- Isogeny-based QSC: an overview
- Accessible abstractions for Isogeny-based QSC

Isogenies for “Dummies”?

Can we build accessible, crypto-friendly
abstractions for isogenies?

Cryptographic Group Actions

- Framework for isogeny-based crypto from **cryptographic “hard” group actions**
 - Abstract, crypto-friendly, particularly suitable for feasibility and impossibility results
 - Similar in flavor to using bilinear maps without knowing/thinking of Miller’s algorithm
- Can be instantiated from **CSIDH** and its derivatives (e.g., **CSI-FiSh**, **SCALLOP**, **Clapoti**,)
 - Captures **a subset of** isogeny-based hardness assumptions and cryptographic design techniques
 - Specifically, abstracts **isogenies of supersingular “oriented” curves**



Isogeny-based assumptions



Cryptographic group actions



Quantum-safe Cryptography

First constructions of several advanced cryptographic primitives from isogenies

- Advanced variants of public key encryption (CCA2-secure, circular/KDM secure, dual-mode)
- Trapdoor one-way functions
- Several (round-optimal) variants of oblivious transfer (OT) and secure multiparty computation (MPC)
- Several types of proof systems: Two-message witness indistinguishable proofs with private-coin verifier, designated-verifier non-interactive zero-knowledge proofs
- Cryptographic primitives with hinting property

[AMP19]: N. Alamati, H. Montgomery, **S. Patranabis**: Symmetric Primitives with Structured Secrets. **CRYPTO 2019**

[AMP23]: N. Alamati, H. Montgomery, **S. Patranabis**, A. Roy: Minicrypt Primitives with Algebraic Structure and Applications. **EUROCRYPT 2019**, **Journal of Cryptology 2023**

[AMP21]: N. Alamati, H. Montgomery, **S. Patranabis**, P. Sarkar: Adaptive MPC from Isogenies, CDH, or LPN. **ASIACRYPT 2021**

[AP22]: N. Alamati, **S. Patranabis**: Cryptographic Primitives with Hinting Property. **ASIACRYPT 2022** **[Best paper by early career authors]**, **Journal of Cryptology 2024**

[BS22]: S. Badrinarayanan, **S. Patranabis**, P. Sarkar: Statistical Security in Two-Party Computation Revisited. **TCC 2022**

[BCCDFLMPW23]: A. Basso, G. Codogni, D. Connolly, L. De Feo, T. Fouotsa, G. Lido, T. Morrison, L. Panny, **S. Patranabis**, B. Wesolowski: Supersingular Curves You Can Trust. **EUROCRYPT 2023**

[BMMPS23]: S. Badrinarayanan, D. Masny, P. Mukherjee, **S. Patranabis**, S. Raghuraman, P. Sarkar: Round-Optimal Oblivious Transfer and MPC from Computational CSIDH. **PKC 2023**

Group Actions Today

Popular applications (old and new)

- Public-key encryption (Couveignes–Rostovtsev–Stolbunov, CSIDH and its variants, SCALLOP)
- Digital signatures (SeaSign, CSI-FiSh, Clapoti, PEGASIS, LESS-FM)
- Threshold cryptography
 - DKG, threshold encryption, threshold signatures – Sashimi, CSI-SharK
 - **PQarrots: Macaw** (threshold signature) **Kea** (threshold public-key encryption), **Kakapo** (distributed key generation) – **a submission to the NIST Multi-Party Threshold Cryptography (MPTC) Project**

Several advanced feasibility/impossibility results

- Advanced variants of digital signatures (e.g., group, ring, blind, and adaptor signatures)
- Advanced variants of public key encryption (e.g., updatable encryption)
- Advanced variants of key-exchange (password-authenticated key exchange)
- Advanced proof systems: Logarithmic OR proofs
- Building blocks for quantum cryptographic primitives
 - Trapdoor claw-free functions (enable classical verification of quantum computation and proofs of quantumness)
 - Walkable invariants (imply quantum lightning and quantum money)
- Ideal models (GGAM/AGAM) for proving generic lower bounds (ala GGM/AGM)
 - Lower bounds on signature sizes

Where have Group Actions fallen short?

Isogenies of non-oriented supersingular elliptic curves

- Collision-resistant hash function (CRHF): the Charles-Goren-Lauter (CGL) CRHF
- New generation of public-key encryption schemes:
 - SIDH fixes (e.g., POKÉ)
 - FESTA (a novel trapdoor OWE based on a constructive application of the SIDH attacks)
- New and prominent digital signatures:
 - **SQIsign**: follows zero-knowledge proof of knowledge (ZKPoK)/Sigma-protocol + Fiat-Shamir paradigm
 - **PRISM**: follows the hash-then-sign paradigm

Advanced cryptographic primitives

- Functionally richer classes of public-key encryption: identity-based encryption (IBE), attribute-based encryption (ABE), functional encryption (FE)
 - Very recent results on isogeny-based IBE (eprint.iacr.org/2026/1457), but not group action-based
 - No known constructions for ABE or FE from any isogeny-based assumptions
- Sequential cryptoprimitives (Time Lock Puzzles aka TLPs, Verifiable Delay Functions aka VDFs)
 - Very recent results on isogeny-based post-quantum TLP, but not group action-based
 - No isogeny-based post-quantum VDF beyond the generic DF + SNARK route

Isogenies for Dummies: The Road Ahead

- Can we design simple, crypto-friendly abstractions for isogeny techniques not captured by group actions?
- Specifically, can we design a crypto-friendly abstraction for SQIsign?



Why SQIsign?

- SQIsign (**S**hort **Q**uaternion and **I**sojeny **s**ignature)
 - Most prominent isogeny-based digital signature today
 - Security is based on the most fundamental problem in isogeny-based cryptography
 - Microcosm of many novel isogeny techniques over non-oriented supersingular curves
 - Not very accessible; has a very complicated security proof [ABDPW25]



SQIsigning like it's the 80s

- Based on isogeny-ideal correspondence
- [KLPT14, DKLPW20, DKLPW22]



Modern SQIsigning (NIST submission)

- Higher dim. isogeny representations
- [DLRW23, BDDL+23, DF24, NOCC+24, ON24]

[Illustrations borrowed from Luca De Feo's talk at SQIParty 2025]

[ABDPW25] Marius A. Aardal, Andrea Basso, Luca De Feo, **Sikhar Patranabis**, Benjamin Wesolowski: A Complete Security Proof of SQIsign. **Crypto 2025** [eprint.iacr.org/2025/379]

SQLsign for “Dummies”: An Attempt

Forensic Categories [BDPRW25]

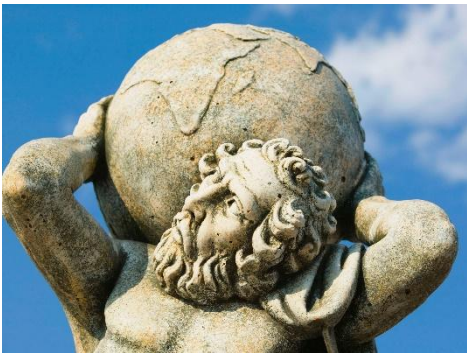
- A framework based on category theoretic concepts that abstracts the key features of SQLsign
- Applications of the framework:
 - A 3-round interactive zero-knowledge proof of knowledge (ZKPoK) for a (trapdoor) hard language (abstracts the Sigma-protocol underlying SQLsign)
 - New constructions of ring signature, trapdoor CRHF (simpler and more efficient than group action-based constructions)
- Two instantiations from isogenies of (non-oriented) supersingular curves:
 - Kernels of smooth-degree isogenies (yields original SQLsign)
 - Level structures (yields a novel variant of SQLsign, also introduced concurrently in [BDS25])

Based on:

- [BDPRW25] Andrea Basso, Luca De Feo, **Sikhar Patranabis**, Ilinca Radulescu, Benjamin Wesolowski: Forensic categories: a framework for SQLsign-like primitives. [eprint.iacr.org/2026/1171]
- [ABDPW25] Marius A. Aardal, Andrea Basso, Luca De Feo, **Sikhar Patranabis**, Benjamin Wesolowski: A Complete Security Proof of SQLsign. **Crypto 2025** [eprint.iacr.org/2025/379]
- SQLsign submission to the NIST standardization process [<https://sqisign.org/>]

Open questions

- Build advanced public-key primitives from isogenies?
 - Fun part: design crazily powerful abstractions that yield powerful cryptographic primitives
 - Reality check: can we instantiate them from (post-quantum) isogeny-based assumptions?
- Integrate a broader class of isogeny-based techniques into abstract frameworks?
 - SOTA public-key encryption schemes (POKÉ, FESTA)
 - Alternative (hash-then-sign) digital signature schemes (PRISM)
- Instantiate forensic categories from a wider class of assumptions?
 - Attempt to model several constructions from other assumptions using (forensic) categories



Isogeny-based assumptions

More assumptions?



Forensic Categories



- ZKPoK
- Ring signature
- Trapdoor CRHF



More primitives?



SPACE 2026: Sixteenth International Conference on Security, Privacy and Applied Cryptographic Engineering

December 16-19, 2026 — Bangalore, India

Co-located with Indocrypt 2026! Featuring multiple keynotes, industry talks, paper and poster presentations

Scope and Focal Areas:

- Cryptographic Engineering and Hardware Security
- Machine Learning Security and Privacy
- Applied Crypto, Blockchain and Distributed Systems
- Security, Privacy and Usability

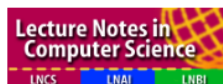
Upcoming Submission Deadline (Cycle 2):

- Abstract registration deadline: **September 18, 2026 (AoE)**
- Paper submission deadline: **September 25, 2026 (AoE)**
- Notification of acceptance: **November 2, 2026 (AoE)**
- **Submission site:** space2026-cycle-2.hotcrp.com/

Proceedings Information

All accepted conference papers will be published in Springer's LNCS series.

Check previous SPACE proceedings on [SpringerLink](https://www.springerlink.com/)



Organizing Committee

Program Chairs

Bimal Viswanath
Virginia Tech

Sujoy Sinha Roy
TU Graz

Sikhar Patranabis
IBM Research

General Co-Chairs

Debdeep Mukhopadhyay
IIT Kharagpur

S.D. Sudarsan
CDAC Bangalore

Organizing Chair

Raghvendra Patil
CDAC Bangalore

Web Chair

Selim Kirbiyik
TU Graz