

Quantum Computing Risk & Post-Quantum Cryptography

A Technical and Strategic Roadmap

JP Aumasson

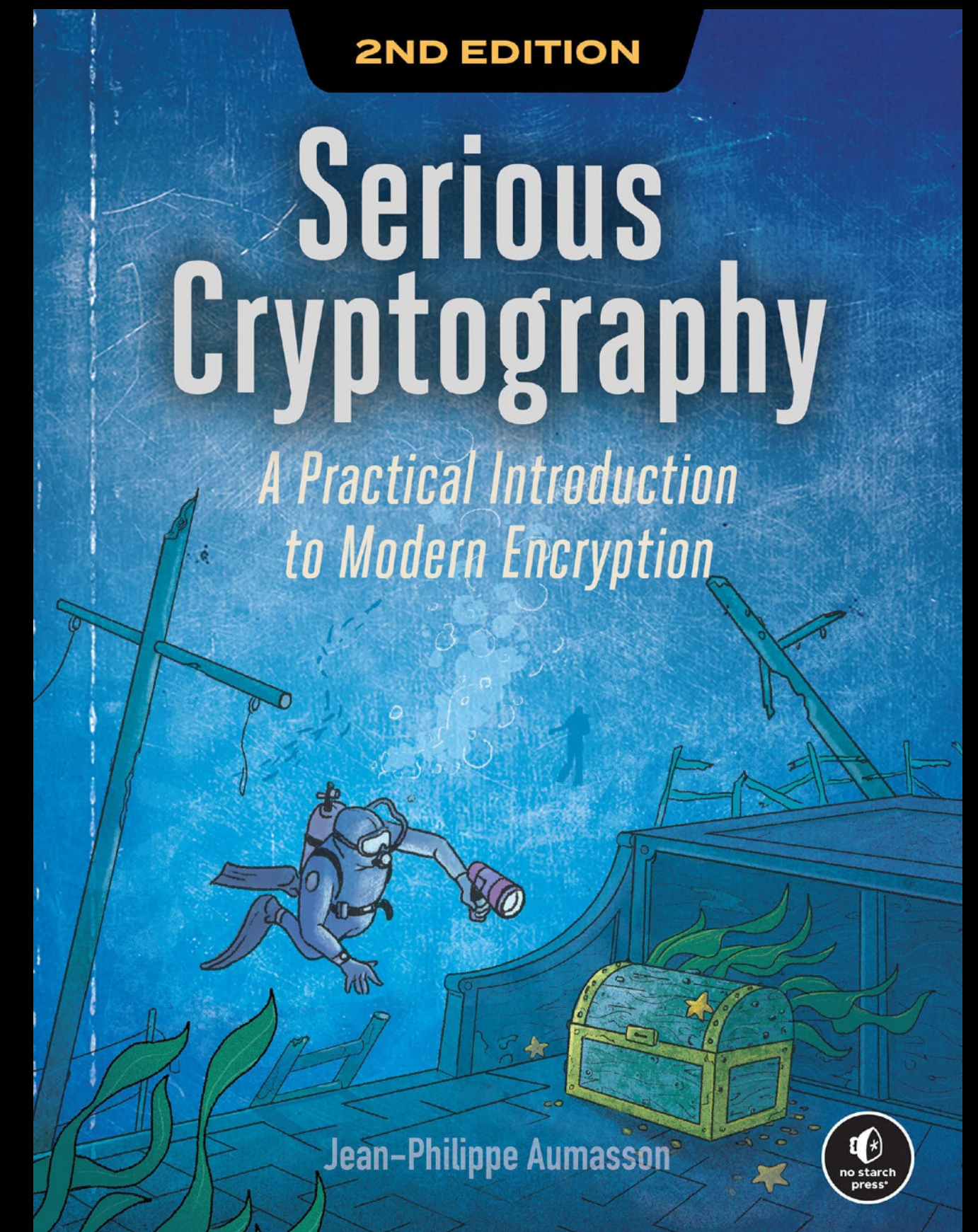


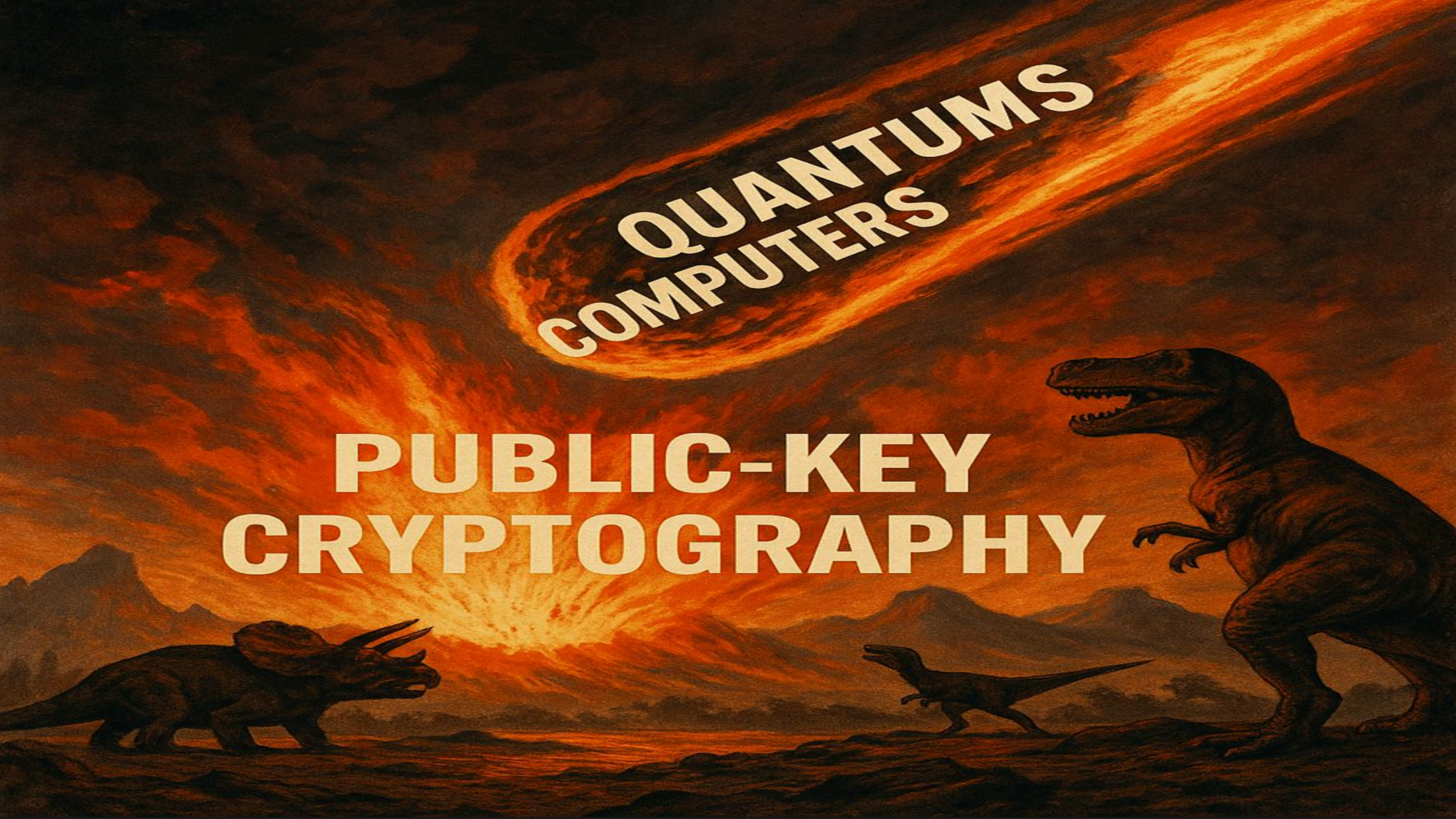
About me

aumasson.jp | bfswa.substack.com | taurushq.com

- 20 years in cryptography & security
- Co-author **SLH-DSA**, BLAKE2/3, SipHash
- **Chief Security Officer** at Taurus SA
 - Crypto asset custody tech for banks
 - Managing **quantum risk** since 2018

TAURUS





**QUANTUMS
COMPUTERS**

**PUBLIC-KEY
CRYPTOGRAPHY**

Quantum computers

A different way to compute, using quantum mechanics

- NOT **faster** computers, in terms of clock cycle

Quantum computers

A different way to compute, using quantum mechanics

- NOT **faster** computers, in terms of clock cycle
- NOT **trying-all-answers-simultaneously**

Quantum computers

A different way to compute, using quantum mechanics

- NOT **faster** computers, in terms of clock cycle
- NOT **trying-all-answers-simultaneously**
- Could only solve VERY FEW math problems, including
 - **Factoring**: find p and q given $n = p \times q$
 - **Discrete logarithm**: find d given x, p , and $y = x^d \bmod p$

Quantum computers

A different way to compute, using quantum mechanics

- NOT **faster** computers, in terms of clock cycle
- NOT **trying-all-answers-simultaneously**
- Could only solve VERY FEW math problems, including
 - **Factoring**: find p and q given $n = p \times q$
 - **Discrete logarithm**: find d given x, p , and $y = x^d \bmod p$
- 🤯 These are the hard problems in RSA and elliptic curve schemes

This talk

1. How bad is it?
2. Post-quantum cryptography
3. Tech you can use today
4. Migration plan guidelines



1. How bad is it?

- 🍌 Impact on cryptography
- 🔬 Quantum computers today
- 📢 Recent announcements
- 💰 The curious cryptocurrency case



Impact on cryptography (1/3)

Signature and authentication

Anything using RSA or elliptic curve crypto is broken:

- **Public-key infrastructure** (PKI): certificates, OCSP responses, CT logs
- **General infrastructure**: DNSSEC, BGP/RPKI, EMV payment
- **Code signing**: OS applications, firmware, APT/RPM packages, dependencies
- **Authentication** protocols: SSH host keys, FIDO2, client certs
- **Blockchain**: transaction signatures

Impact on cryptography (2/3)

Encryption and key agreement

Anything using RSA or elliptic curve crypto is broken:

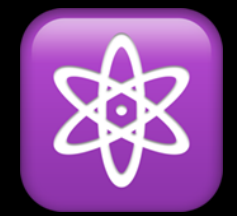
- **Key establishment:** TLS, SSH, IPsec, WireGuard, messaging protocols
- **Encryption key wrapping:** PGP, S/MIME, KEK, ECIES
- **Protecting data confidentiality**, e.g. some privacy-oriented blockchains

“Store now, decrypt later” applies, unlike signature/authentication

Impact on cryptography (3/3)

Other cryptography

- “**MPC**” versions of ECDSA, Ed25519
- Many **ZK proof** systems: blockchains' ZK rollups, zkVMs, privacy coins
- Most deployed **PAKEs** (password-based auth protocols), incl. OPAQUE
- Private set intersection, as used in iCloud
- Anonymous credential protocols, as Privacy Pass



Quantum computers today: ~100 qubits

Research prototypes of no practical interest yet

Most visible companies are **Google** “Quantum AI” and **IBM**, many others



Challenges:

- **Scale** to a large number of connected, entangled qubits
- Improve **coherence** time and the number of **quantum gates**
- Maintain low **error rate** and support quantum **error correction**

Recent announcements (Google)

Reduced the number of qubits and gates needed to break crypto

Actual algorithm **undisclosed** (evidence as a ZK proof)

*"[could] execute in minutes using fewer than **half a million** physical qubits"*

Builds upon the results from eprint.iacr.org/2026/280

Safeguarding cryptocurrency
by disclosing quantum
vulnerabilities responsibly



Google
Quantum AI

March 31, 2026 ·

Ryan Babbush, Director of Research, Quantum Algorithms, and Hartmut Neven, VP of Engineering, Google Quantum AI, Google Research

Reducing the Number of Qubits in Quantum
Discrete Logarithms on Elliptic Curves *

Clémence Chevignard, Pierre-Alain Fouque^{id}, and André Schrottenloher^{id}

Univ Rennes, Inria, CNRS, IRISA, Rennes, France
`firstname.lastname@inria.fr`

Recent announcements (Oratomic)

“Neutral atoms” approach may require even fewer qubits

Neutral atoms technology architecture

No hardware, only theoretical description

2025 experiment with 6100 qubits, but:

- No entanglement, 13 seconds coherence
- No error correction implemented
- Not a quantum computer

Quantum Physics

[Submitted on 30 Mar 2026]

Shor's algorithm is possible with as few as 10,000 reconfigurable atomic qubits

[Madelyn Cain](#), [Qian Xu](#), [Robbie King](#), [Lewis R. B. Picard](#), [Harry Levine](#), [Manuel Endres](#), [John Preskill](#), [Hsin-Yuan Huang](#), [Dolev Bluvstein](#)

Article | [Open access](#) | Published: 24 September 2025

A tweezer array with 6,100 highly coherent atomic qubits

[Hannah J. Manetsch](#), [Gyohei Nomura](#), [Elie Bataille](#), [Xudong Lv](#), [Kon H. Leung](#)  & [Manuel Endres](#) 

[Nature](#) **647**, 60–67 (2025) | [Cite this article](#)

When...

will a quantum computer break cryptography?

Nobody knows. Could be 2030, 2060, 2300, or never.

IMHO no "Q-Day" before 2036.



💰 The curious cryptocurrency case

Or "the quantum gravity principle"

"No quantum computer will ever be used to steal bitcoins"

- Price will **crash** if Bitcoin vulnerable AND a QC is **known/suspected** to exist
- A QC kept secret has much more valuable use than stealing cryptocurrency

Quantum computers will not steal your
bitcoins, even if they can

The quantum gravity principle

NOV 13, 2025 • JP AUMASSON



bfswa.substack.com/p/quantum-computers-will-not-steal

2. Post-quantum cryptography

- 💡 What is PQC?
- 🦅 NIST standards and deadlines
- 🚀 Performance trade-offs
- 🦁 Regional PQC initiatives
- 🔒 PQ HTTPS adoption



What is PQC?

a.k.a. quantum-safe or quantum-secure

Public-key crypto **not vulnerable** to quantum algorithms

- Must not rely on the hardness of factoring or discrete logarithm

💡 What is PQC?

a.k.a. quantum-safe or quantum-secure

Public-key crypto **not vulnerable** to quantum algorithms

- Must not rely on the hardness of factoring or discrete logarithm
- Instead, PQC relies on
 - Problems related to **NP-hard** problems (lattice, error correction)
 - Breaking symmetric cryptography (like finding **hash function preimages**)
 - Problems that we don't know how to solve classically :)



De facto PQC standards

Developed via a public competition (like AES and SHA-3) since 2016

Signature and **KEM** (key encapsulation mechanism, encryption)

FIPS 203

Federal Information Processing Standards Publication

Module-Lattice-Based
Key-Encapsulation Mechanism Standard

Category: Computer SecuritySubcategory: Cryptography

Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8900

This publication is available free of charge from:
<https://doi.org/10.6028/NIST.FIPS.203>

Published August 13, 2024

ML-KEM

FIPS 204

Federal Information Processing Standards Publication

Module-Lattice-Based Digital
Signature Standard

Category: Computer SecuritySubcategory: Cryptography

Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8900

This publication is available free of charge from:
<https://doi.org/10.6028/NIST.FIPS.204>

Published August 13, 2024

ML-DSA

FIPS 205

Federal Information Processing Standards Publication

Stateless Hash-Based Digital Signature
Standard

Category: Computer SecuritySubcategory: Cryptography

Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8900

This publication is available free of charge from:
<https://doi.org/10.6028/NIST.FIPS.205>

Published: August 13, 2024

SLH-DSA

+ **FN-DSA**
(signature)

+ **"HQC"**
(KEM)

NIST deadlines

PQC must be deployed by 2035

In NIST IR 8547 "Transition to Post-Quantum Cryptography Standards"

Digital Signature Algorithm Family	Parameters	Transition
ECDSA [FIPS186]	112 bits of security strength	Deprecated after 2030 Disallowed after 2035
	≥ 128 bits of security strength	Disallowed after 2035
EdDSA [FIPS186]	≥ 128 bits of security strength	Disallowed after 2035
RSA [FIPS186]	112 bits of security strength	Deprecated after 2030 Disallowed after 2035
	≥ 128 bits of security strength	Disallowed after 2035

Key Establishment Scheme	Parameters	Transition
Finite Field DH and MQV [SP80056A]	112 bits of security strength	Deprecated after 2030 Disallowed after 2035
	≥ 128 bits of security strength	Disallowed after 2035
Elliptic Curve DH and MQC [SP80056A]	112 bits of security strength	Deprecated after 2030 Disallowed after 2035
	≥ 128 bits of security strength	Disallowed after 2035
RSA [SP80056B]	112 bits of security strength	Deprecated after 2030 Disallowed after 2035
	≥ 128 bits of security strength	Disallowed after 2035

Federal agencies: mandated by OMB M-23-02 & NSM-10, enforced by **CISA**

Hybrid schemes

Best of both worlds

Hybrid schemes combine post-quantum and “pre-quantum” schemes

3.2. PQC-Classical Hybrid Protocols

The migration to post-quantum cryptography may initially include hybrid solutions that incorporate the use of quantum-resistant and quantum-vulnerable algorithms when establishing cryptographic keys or generating digital signatures. These hybrid solutions are typically designed to remain secure if at least one of the component algorithms is secure.

Hybrid KEM = 2 ciphertexts: X25519 and ML-KEM, for example

Secure as long as one of the two is secure, adds **performance overhead**

Best choice depends on the threat model and data criticality

Performance trade-offs

KEMs

Value (bytes)	ECDH P-256	ML-KEM-512 post-quantum	HQC-128 post-quantum
Public key	64	800	2,249
Private key	32	1,632	2,289
Ciphertext	65	768	4,481

- ML-KEM's speed **comparable** to ECDH, HQC orders of magnitude **slower**
- An ML-KEM private key can be "compressed" to a 64-byte seed
- **Hybrid** cheap: ciphertext size dominated by the PQ part

Performance trade-offs

Signatures

Value (bytes)	ECDSA P-256	ML-DSA-44 post-quantum	SLH-DSA-128s post-quantum
Public key	64	1,312	32
Private key	32	2,528	64
Signature	64	2,420	7,856

- ML-DSA's speed **comparable** to ECDSA, SLH-DSA orders of magnitude **slower**
- ML-DSA's private key can be "compressed" to 32 bytes
- **Hybrid** cheap, but less valuable than for KEMs (no confidentiality breach)

Regional PQC initiatives

A short overview

-  **China:** Competition program to select standards (NGCC)
www.niccs.org.cn, submission deadline June 2026

关于开展新一代商用密码算法征集活动的公告

为应对量子计算威胁，推动新一代商用密码算法标准制定，按照密码行业标准化技术委员会工作安排，我院将面向全球陆续开展新一代公钥密码算法、密码杂凑算法、分组密码算法征集活动，从安全性、性能、特点等方面组织评估，遴选出优胜算法开展标准化工作。欢迎各界积极参与算法提交与公开评议，鼓励在算法设计工作中加强国际合作。

征集活动安排具体事宜后续将在www.niccs.org.cn相继发布，敬请关注。

商用密码标准研究院
2025年2月5日

Announcement on Launching the Next-generation Commercial Cryptographic Algorithms Program (NGCC)

In response to the threat of quantum computing and to promote the standardization of the next-generation commercial cryptographic algorithms, the Institute of Commercial Cryptography Standards (ICCS) is launching a global program to call for proposals for next-generation public-key cryptographic algorithms (NGCC-PK), cryptographic hash algorithms (NGCC-CH) and block cipher algorithms (NGCC-BC), according to the arrangement of Chinese Cryptography Standardization Technical Committee. The candidate algorithms will be evaluated in terms of security, performance and other features, and the finalists will be considered for standardization. ICCS looks forward to global algorithm submissions and comments, and encourages international cooperations in algorithm design.

Further notifications of the program will be released on www.niccs.org.cn

Institute of Commercial Cryptography Standards, China

February 5, 2025

Regional PQC initiatives

A short overview

-  **China:** Competition program to select standards (NGCC)
www.niccs.org.cn, submission deadline June 2026
-  **Japan:** No local standards, CRYPTREC (NIST equivalent) reviewed PQC algorithm, not formal guidelines yet www.cryptrec.go.jp

● **CRYPTREC 暗号技術ガイドライン
(耐量子計算機暗号) 2024年度版**

3.1.1	LWE 問題の紹介
3.1.2	NTRU 問題の紹介
3.1.3	格子問題の公開チャレンジの求解状況
	格子に基づく代表的な暗号方式
3.2.1	Hash-and-Sign に基づく署名方式の格子問題への拡張
3.2.2	Fiat-Shamir 署名方式の格子問題への拡張
	格子に基づく主要な暗号方式
3.3.1	FIPS 203 : Module-Lattice-Based Key-Encapsulation Mechanism Standard (ML-KEM)
3.3.1.1	ML-KEM における数論変換
3.3.1.2	ML-KEM の基本構成と処理概要
3.3.1.3	暗号パラメータ
3.3.2	FIPS 204: Module-Lattice-Based Digital Signature Standard (ML-DSA)
3.3.2.1	ML-DSA における数論変換
3.3.2.2	ML-DSA の構成と処理概要
3.3.2.3	暗号パラメータ
3.3.2.4	CRYSTALS-Dilithium との違い
3.3.3	FALCON

Regional PQC initiatives

A short overview

-  **China:** Competition program to select standards (NGCC)
www.niccs.org.cn, submission deadline June 2026
-  **Japan:** No local standards, CRYPTREC (NIST equivalent) reviewed PQC algorithm, not formal guidelines yet www.cryptrec.go.jp
-  **Singapore:** CSA Quantum-Safe Handbook and Readiness Index
www.csa.gov.sg/resources/publications/quantum-safe-handbook-and-quantum-readiness-index/

(Draft for Public Consultation)
**QUANTUM-SAFE
MIGRATION
HANDBOOK**

(DRAFT FOR PUBLIC CONSULTATION)
**QUANTUM
READINESS INDEX**



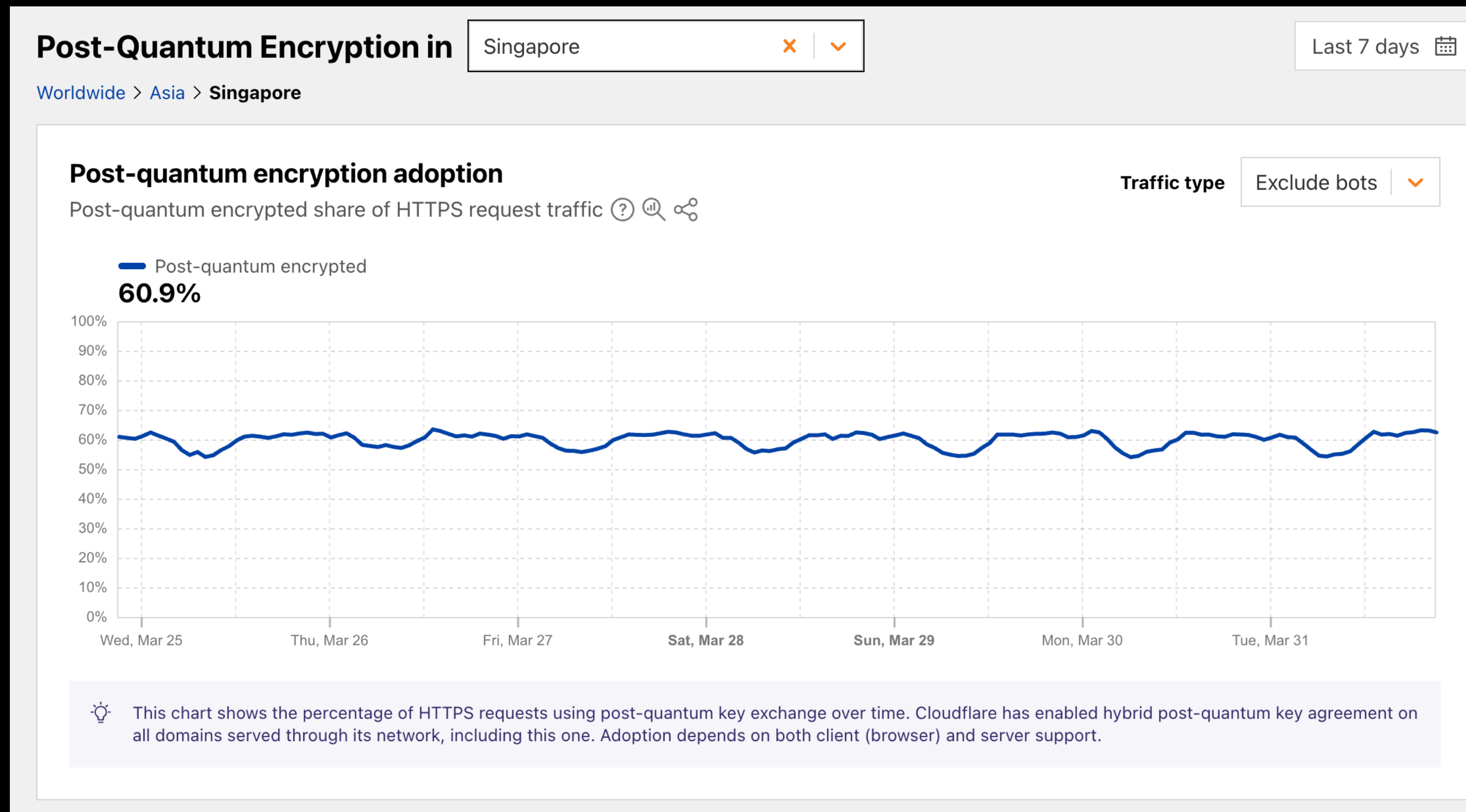
Regional PQC initiatives

A short overview

-  **China:** Competition program to select standards (NGCC)
www.niccs.org.cn, submission deadline June 2026
-  **Japan:** No local standards, CRYPTREC (NIST equivalent) reviewed PQC algorithm, not formal guidelines yet www.cryptrec.go.jp
-  **Singapore:** CSA Quantum-Safe Handbook and Readiness Index
www.csa.gov.sg/resources/publications/quantum-safe-handbook-and-quantum-readiness-index/
-  **South Korea:** KpqC competition kpqc.or.kr 2022-2025
 - KEM: NTRU+, SMAUG-T (lattice-based)
 - Signature: AlMer (**MitH**), HAETAE (lattice-based)

🔒 PQ HTTPS adoption

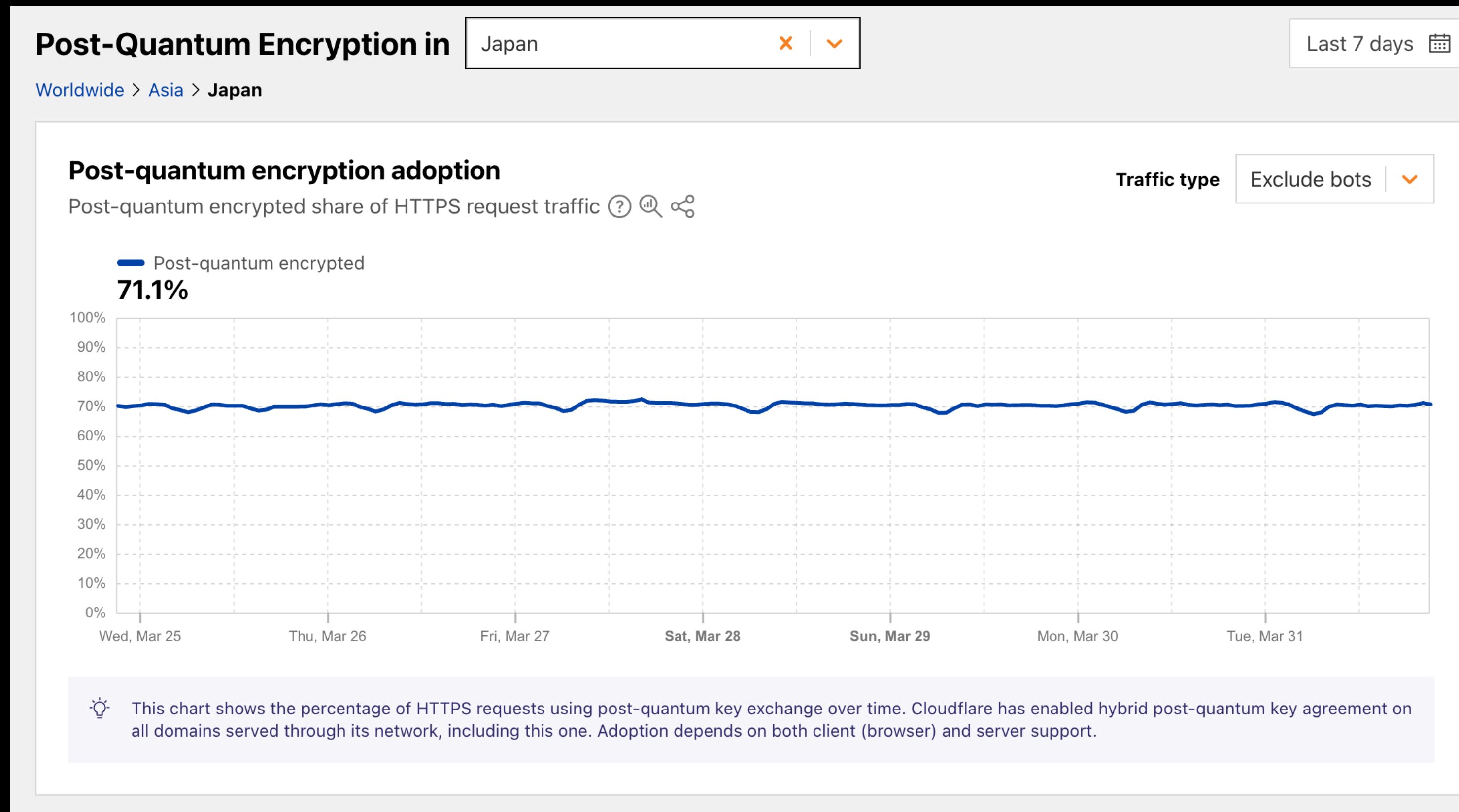
HTTPS traffic estimates by Cloudflare 🇸🇬



radar.cloudflare.com/post-quantum/sg

🔒 PQ HTTPS adoption

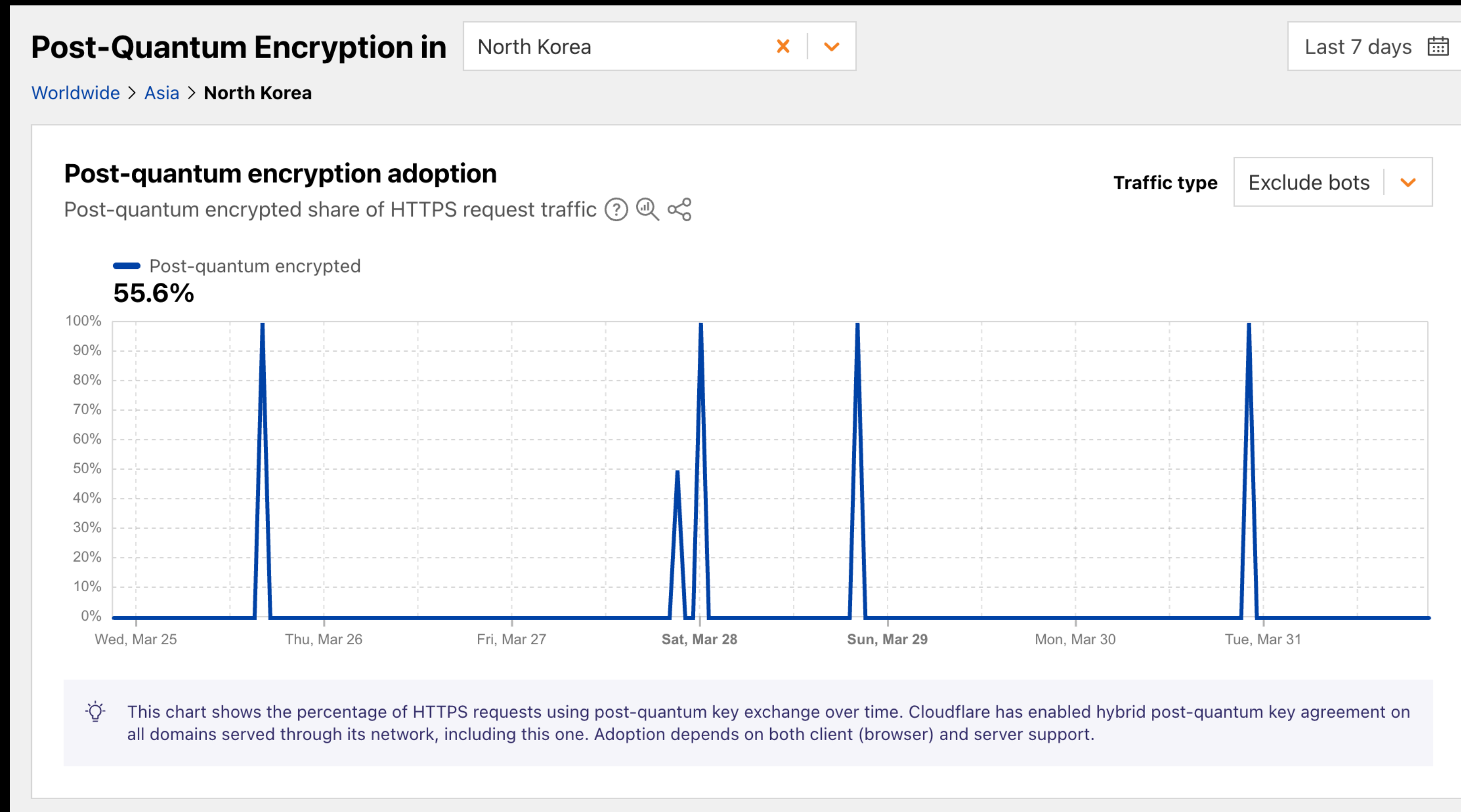
HTTPS traffic estimates by Cloudflare 🇯🇵



radar.cloudflare.com/post-quantum/jp

🔒 PQ HTTPS adoption

HTTPS traffic estimates by Cloudflare 🇰🇵



radar.cloudflare.com/post-quantum/kp

3. Tech you can use today

- ☁ Cloud KMS
- 📱 Mobile platforms
- 🌐 Other infrastructure
- ⚙ Open-source software



Cloud KMS

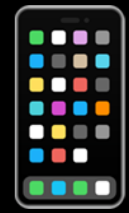
Key Management Services



- **AWS:** ML-DSA signing
- **Google:** ML-DSA + ML-KEM and X-Wing (hybrid) for decryption/wrapping
- **Azure:** no PQC support documented

Cloud HSMs back-ends don't appear to support PQC yet (AWS in preview?)

For consistency, **PQ TLS** should be used for the client <> SaaS link



Mobile platforms

iOS, Android

iOS 26

support.apple.com/guide/security/quantum-secure-cryptography-apple-devices-secc7c82e533/web

- TLS defaults to PQ KEM (via URLSession)
- CryptoKit API integrates ML-DSA and ML-KEM
- Secure Enclave support

Quantum-secure cryptography in Apple operating systems

Android 17

security.googleblog.com/2026/03/post-quantum-cryptography-in-android.html

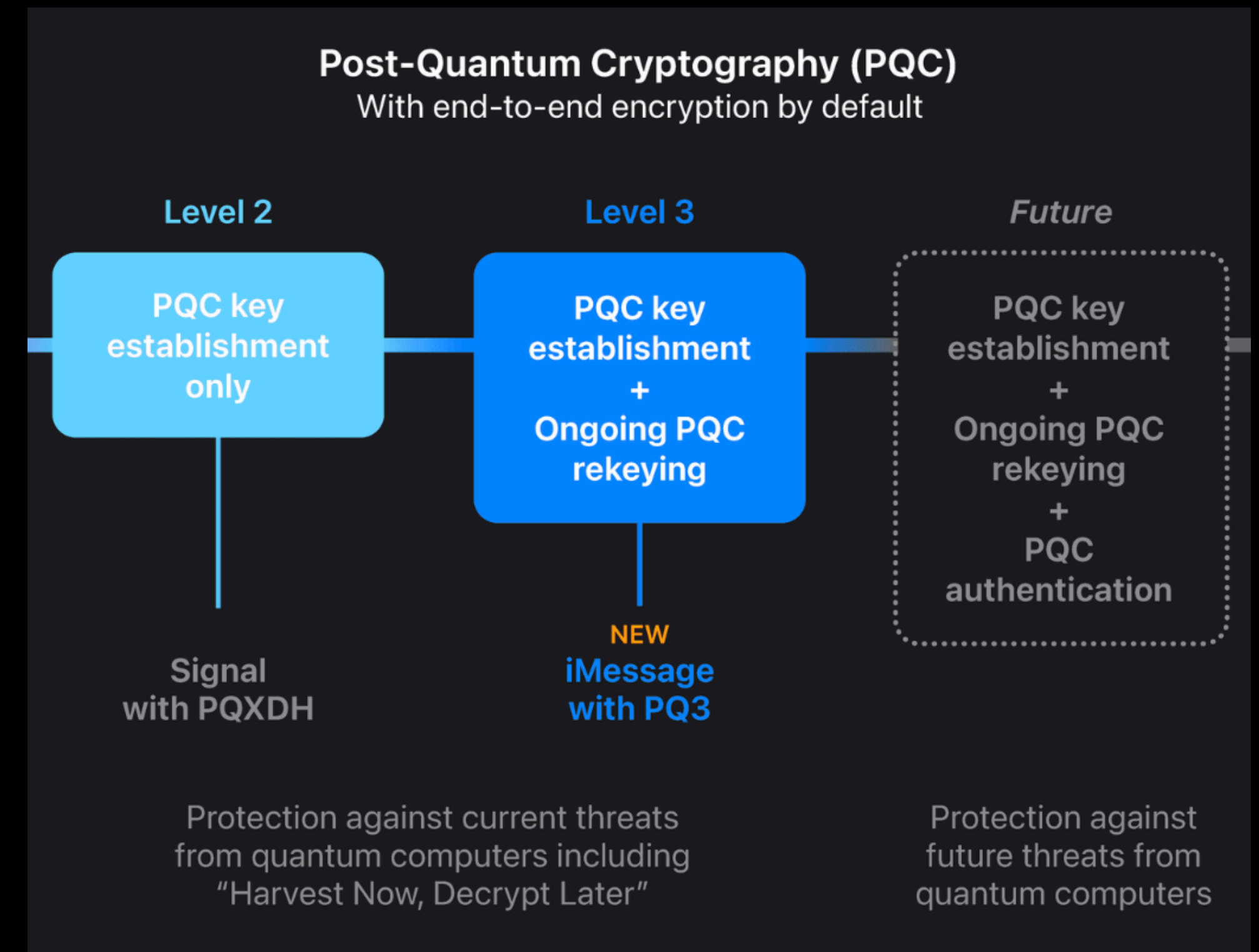
- ML-DSA in Keystore (via KeyPairGenerator)
- Bootloader integrity chain evolving to be PQ
- PQ TLS not enabled by default (custom Conscript needed)

Security for the Quantum Era: Implementing Post-Quantum Cryptography in Android
March 25, 2026

Other infrastructure (1/3)

Apple

- **iMessage** is PQ (hybrid key agreement)
- **VPN** support (incl. IPsec API)
- Apple **Watch** <> iPhone comm is PQ



security.apple.com/blog/imessage-pq3/

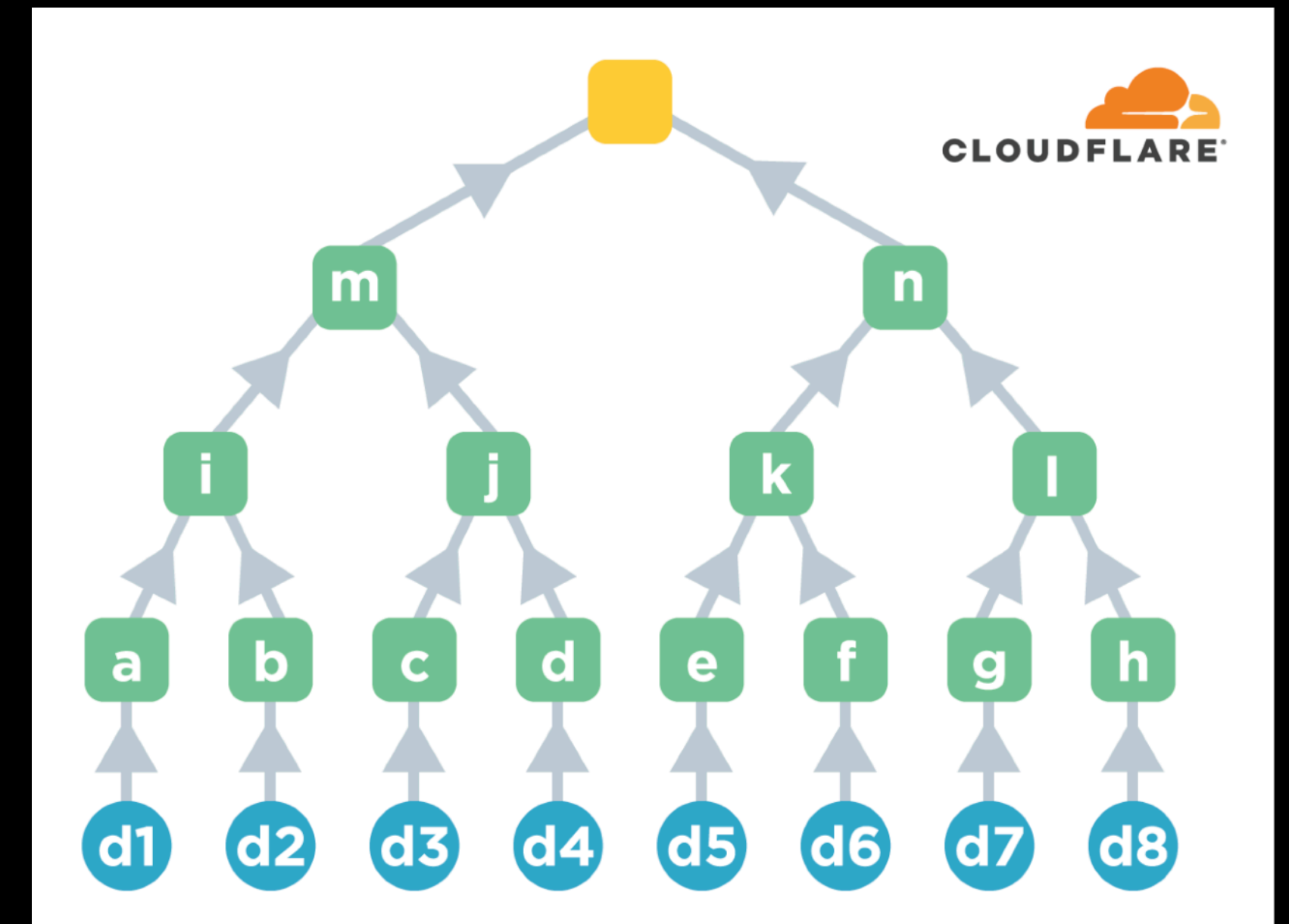
More in support.apple.com/guide/security/quantum-secure-cryptography-apple-devices-secc7c82e533/web

Other infrastructure (2/3)

Cloudflare

- Cloudflare **Tunnel** PQ by default
- PQ **TLS** 1.3 with hybrid key agreement
- Targets full product suite PQness by **2029**
- Co-introduced **Merkle Tree Certificates** for efficient PQ cert adoption

More in www.cloudflare.com/pqc/



<https://blog.cloudflare.com/bootstrap-mtc/>

Other infrastructure (3/3)

Other systems

- **HSMs**: most support some PQC; ACVP certificate but no FIPS-validated module
- **IPsec**: PQ key agreement defined for IKEv2, in strongSwan (not Libreswan)
- **Kubernetes**: TLS defaults to PQC via Go 1.24's TLS stack
- **Red Hat**: RHEL 10.1+'s DEFAULT policy uses PQC (SSH, TLS stacks)
- **Signal**: e2ee protocol PQ, via the PQXDH protocol (not the Noise transport)
- **WhatsApp**: NOT PQ yet

Open-source software

Likely no need to write your own code

PQC software

- [SUPERCOP](#) - Benchmarks for cryptographic software

General-purpose libraries with PQC support

Does not include TLS implementations listed later:

- [AWS-LC](#) - Rust bindings in [aws-lc-rs](#)
- [Botan](#) - C++
- [Bouncy Castle](#) - Java/C#
- [CIRCL \(Cloudflare Interoperable, Reusable Cryptographic Library\)](#) - Go
- [Google Tink](#) - Multi-language (C++, Go, Java, Obj-C, Python)

TLS implementations with PQC support

- [aws/s2n-tls](#)
- [BoringSSL](#)
- [OpenSSL](#)
- [Go crypto/tls](#)
- [wolfSSL](#)

PQC libraries and language-specific software

C:

- [algorand/falcon](#) - Deterministic FALCON implementation
- [liboqs](#) - From [Open Quantum Safe](#)
- [mupq/pqm4](#) - PQC library for the ARM Cortex-M4
- [PQ Code Package](#) - A Linux Foundation [PQCA](#) project building high-assurance impl track algorithms

Go:

- [Go crypto/mlkem](#) - Official Go implementation of Kyber/ML-KEM

JavaScript:

- [paulmillr/noble-post-quantum](#) - ML-KEM, ML-DSA, SLH-DSA, Falcon, and hybrids

.NET:

- [Post-Quantum Cryptography in .NET](#)

Rust:

- [libcrux](#) - Formally verified code
- [RustCrypto/KEMs](#) - ML-KEM, FrodoKem
- [RustCrypto/signatures](#) - ML-DSA, SLH-DSA, LMS

Zig:

- [std.crypto](#) - ML-DSA and ML-KEM in the standard library

github.com/veorq/awesome-post-quantum

4. Migration plan

- 👎 Find old crypto
- 👍 Replace with PQ crypto
- 🍺 Done!



4. Quantum risk management

- 🎯 General strategy
- 🔭 Observe
- 🔍 Assess
- ⚖️ Decide
- 🚀 Act & monitor



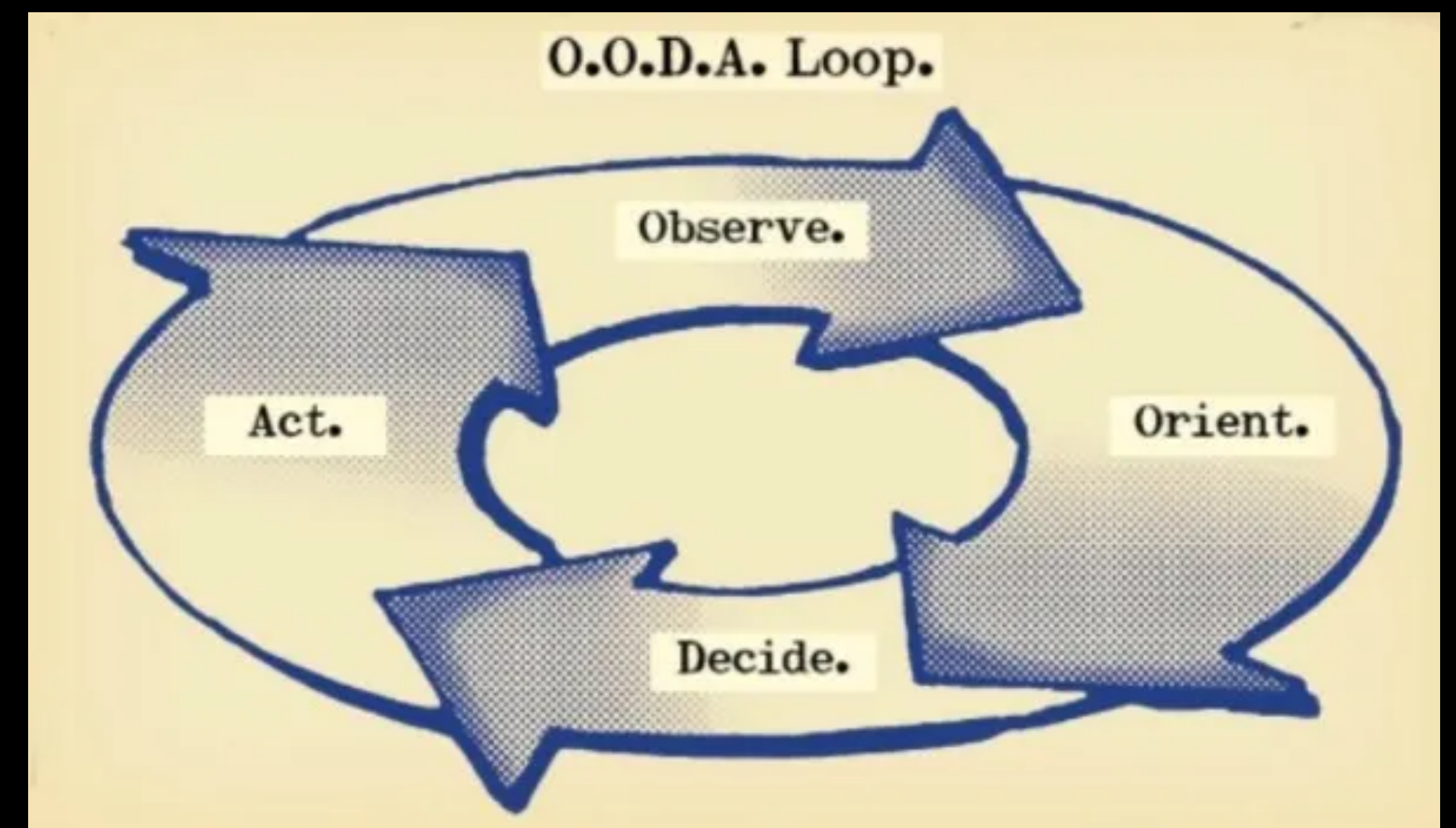
General strategy

Quantum risk management vs. migration

You can **migrate** a product or service to post-quantum cryptography

But for an organization, it's about **continuous quantum risk management**

- **Observe:** Where is crypto used?
- **Orient/assess:** How critical is it?
- **Decide:** What to do about it?
- **Act:** Communicate and apply changes





Inventorize cryptography in your systems *and your suppliers'*

Confidentiality protection

Pubkey encryption & key agreement
KEMs, wrapping of symmetric keys

Encrypted files (if public-key protected)

On internal and shared filesystems, git repos, etc.

Encrypted email and messaging

PGP, S/MIME, messaging apps

Encrypted archives/back-ups

DB backups, cloud storage backups, etc.

Secure transport: TLS, SSH, IPsec, WireGuard, etc.

Pre-shared keys provide partial mitigation

PKI (unless pure auth/signature)

When keys used for key agreement, PGP encryption

KMS and HSM infrastructure

What are they used for & how are they protected

Blockchain & distributed ledger

Is data shared with other parties impacted? Incl. ZK proofs

Signature & Authentication

Access to critical systems must be a priority, even if
"store now decrypt later" does not apply

Code signing (apps, bootloaders, etc.)

Esp. embedded systems with unupdatable integrity checks

PKI / Certificate chains

May include keys used for encryption

Authorization & authentication frameworks

Observe

Methods and tools

Internal systems

- Review documentation and policies (don't modify them at this stage)
- Review configuration files and templates
- Verify alleged PQness on real live systems

Third-party products and services:

- Review documentation and technical settings
- Contact the vendor if needed, ask their PQC roadmap

Tools can help for code/config scan: cbomkit, ssh-audit, commercial solutions

Observe

Example SSH server PQC check: `ssh-audit <host>`

```
→ ~ ssh-audit exe.xyz
# general
(gen) banner: SSH-2.0-SSHPiper
(gen) compatibility: OpenSSH 9.9+ (some functionality from 6.6), Dropbear SSH 2020.79+
(gen) compression: disabled

# key exchange algorithms
(kex) mlkem768x25519-sha256      -- [info] available since OpenSSH 9.9
                                `-- [info] hybrid key exchange based on post-quantum resistant
                                algorithm and proven conventional X25519 algorithm
(kex) curve25519-sha256         -- [info] available since OpenSSH 7.4, Dropbear SSH 2018.76
                                `-- [info] default key exchange from OpenSSH 7.4 to 8.9
(kex) curve25519-sha256@libssh.org -- [info] available since OpenSSH 6.4, Dropbear SSH 2013.62
                                `-- [info] default key exchange from OpenSSH 6.5 to 7.3
```


Assess

Prioritize systems with respect to risk and cost

Risk criteria, for example:

- Data sensitivity lifetime
- Exposure outside the organization
- Assets volume and criticality (PII, key material, classification level)
- Regulatory compliance and contractual obligations

Cost criteria, for example:

- Technical difficulty (mere setting tweak vs. custom development)
- Impact on bandwidth and performance
- Interoperability and certification challenges



Decide

Which systems to prioritize, which algorithms

Updates to make to **documentation** and control frameworks:

- Cryptography policy: newly deployed crypto must be PQ
- Key management procedures: adapt to PQC
- Requirements for suppliers and third-party software

Establish a **roadmap**:

- Milestones, from quick wins to multi-year efforts
- Roles & responsibilities
- Deprecation deadlines



Communicate

To all parties concerned

Internal:

- **Board/management:** “we have a plan”
- **Engineers:** requirements for TLS, SSH, etc.: point to libraries, code examples
- **Procurement/TPRM/legal:** requirements for new vendors

External:

- **Retail clients/users:** public roadmap overview
- **Enterprise clients:** detailed internal documents (crypto policy, etc.)
- **Partners/suppliers:** PQ requirements, plan joint upgrade operations

Act

Plan, develop, test, deploy, pray it actually works

You won't see the end of your PQC program, as legacy crypto will be around for a while, thus the **quantum risk management** program

Create processes for continuous loop implementation:

- Add **discovery** controls: scans, audits, etc.
- Include quantum risk in the **risk management** model
- Evaluate **new** features and providers wrt PQC
- Plan **contingencies** if critical data is decrypted later

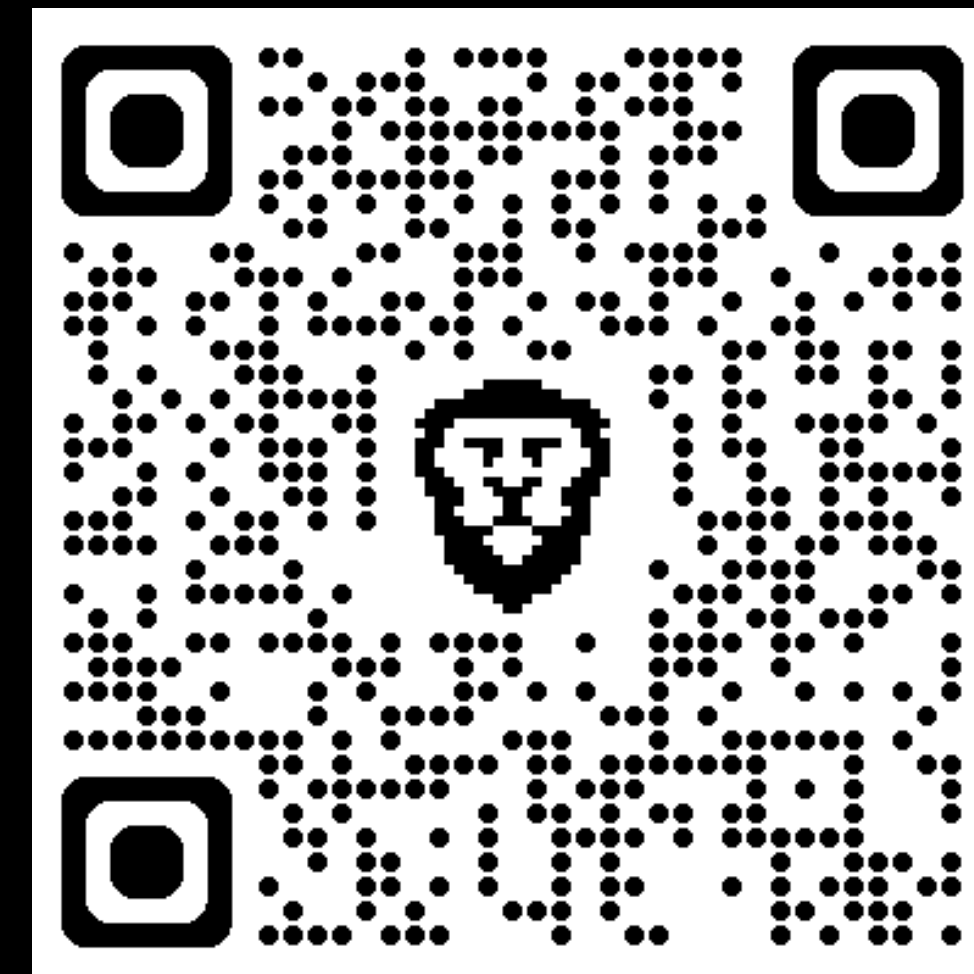


Conclusion

- No need to panic 🌱
- **Integrate** PQC support in your products
- Large organizations must **manage** the risk

Visit github.com/veorq/awesome-post-quantum

- NIST and IETF standards
- Regional migration guidelines
- Tech providers documentation
- Open-source software list



Credits

Thanks for their feedback: Bas Westerbaan, Doug Henkin, Frederic Jacobs, Matthew Green, Nadim Kobeissi, Pierre-Luc Dallaire-Demers, Project Eleven team.

Cat images: [vecteezy.com](https://www.vecteezy.com)



Thank you!

jpa@pm.me
x.com/veorq



Spread the word: store.aumasson.jp