TECHNICAL WHITE PAPER

The Post Quantum Cryptography Imperative: Why Device Makers Must Act Now

A Technical Guide to Post-Quantum Cryptography Migration for Embedded Systems, IoT, and Connected Devices

March 2026 | SEALSQ Corp

Executive Summary

The cryptographic foundations that secure virtually every connected device on the planet are facing a well-defined and time-bound cryptographic transition. Quantum computers, leveraging Shor's algorithm and the physics of quantum superposition, will render the public-key cryptographic algorithms underpinning today's secure communications — RSA, Elliptic Curve Cryptography (ECC), and Diffie-Hellman — efficiently solvable by a sufficiently large quantum computer. This is not a speculative future concern: it is an engineering deadline.

For device makers, this creates a uniquely acute challenge. Unlike enterprise software that can be patched overnight, embedded systems, secure microcontrollers, IoT devices, and connected hardware often operate for 10–20 years in the field. A device designed today and still active in 2032 may rely on cryptographic primitives that are, by that point, demonstrably broken. The window for action is narrowing rapidly.

While timelines for cryptographically relevant quantum computers remain uncertain, the long lifecycle of embedded devices requires action well before full-scale quantum capabilities are realized.

This white paper provides device architects, security engineers, and product managers with a rigorous technical understanding of the quantum threat, the current state of post-quantum cryptographic (PQC) standards, and a practical migration framework. It also introduces SEALSQ's portfolio of quantum-safe semiconductor and security IP solutions, purpose-built for the embedded and IoT market.

KEY FINDING	*Industry consensus, captured in a 2025 Economist Impact survey of quantum professionals across North America, Europe, the UK, and Asia, is that 83% of experts expect quantum utility — the point at which a quantum computer outperforms classical computers at commercially relevant tasks — within a decade. Device engineers must design for a post-quantum world today.*

1. The Quantum Threat: A Rigorous Technical Assessment

1.1 Why Public-Key Cryptography Is Uniquely Vulnerable

Thesecurity of today's public-keyinfrastructure restsonthecomputationalintractability of specific mathematical problems on classical hardware:

- RSA: hardness of integer factorization — given N = P × Q, recovering P and Q from N alone.
- Elliptic Curve Cryptography (ECC): hardness of the elliptic curve discrete logarithm problem (ECDLP) — given a point a·G on curve E(F_p), recovering the scalar a.
- Diffie-Hellman (DH/ECDH): hardness of the discrete logarithm problem in a cyclic group — given g^a mod p (or a point a·G on an elliptic curve), recovering the scalar a.

These problems share a fundamental structural property: periodicity. The mathematical relationships between public and private keys contain harmonic, periodic structure. On classical hardware, exploiting this periodicity requires exponential time — effectively making brute-force attacks infeasible for key sizes used today (RSA-2048, P-256, etc.).

The critical insight is that this periodicity is the vulnerability, not merely an incidental property. A computational model capable of efficiently detecting periods in mathematical functions — precisely what quantum computers can do via the Quantum Fourier Transform, can collapse the difficulty of these problems from exponential to polynomial time.

1.2 Shor's Algorithm: The Mathematical Weapon

In1994, PeterShor demonstratedthat aquantum computercould solve integer factorization and discrete logarithm problems in polynomial time. The algorithm exploits quantum superposition and the Quantum Fourier Transform (QFT) to detect periodicity with exponential speedup over classical approaches.

Shor’s algorithm reduces the complexity of factoring and discrete logarithms from **sub-exponential classical complexity to polynomial-time quantum complexity**, making currently secure key sizes (e.g., RSA-2048, ECC P-256) vulnerable to sufficiently large quantum computers.

TECHNICAL NOTE	*A May 2025 Google Security analysis estimated that breaking 2048-bit RSA would require approximately 1 million physical qubits running for one week, accounting for error correction overhead. Current quantum processors operate in the **hundreds to low thousands of physical qubits**, far below the scale required for cryptographically relevant attacks, but progressing steadily.*

1.3 Grover's Algorithm: A Weaker But Real Threat to Symmetric Crypto

Symmetric algorithms (AES,SHA-2, SHA-3) donotrely onmathematicalrelationships between public and private keys. They are therefore immune to Shor's algorithm. However, Grover's algorithm provides a quadratic speedup for brute-force search over unstructured problem spaces — a category that includes brute-forcing symmetric key spaces.

The practical consequence: AES-128 effectively becomes equivalent in strength to a 64-bit key against a Cryptographically Relevant Quantum Computer (CRQC) running Grover's. AES-256 degrades to 128-bit equivalent security — still considered computationally infeasible for the foreseeable future.

Mitigation is straightforward: migrate from AES-128 to AES-256, and from SHA-256 to SHA-384 or SHA-512. These transitions involve no algorithm replacement, only parameter upgrades — a manageable engineering task for most device platforms.

⚠ Critical distinction: Symmetric crypto (AES, SHA) is threatened but manageable with key- length upgrades. Public-key crypto (RSA, ECC, DH) faces complete algorithmic compromise by a sufficiently capable quantum computer. These require full algorithm replacement — not parameter tuning.

1.4 The 'Harvest Now, Decrypt Later' Attack Vector

The mostimmediatethreat to device makersisnotafutureattack on operational devices. It is data exfiltration happening right now.

Adversaries — nation-state actors in particular — are actively intercepting and archiving encrypted communications, firmware images, key exchange sessions, and device telemetry today. The decryption of this harvested material will occur retroactively once a CRQC becomes available. This is the 'Harvest Now, Decrypt Later' (HNDL) threat model.

For device makers, this means:

- Any firmware signing key material transmitted over classical public-key channels today is potentially captured and stored by adversaries.
- Confidential device provisioning data, root-of-trust certificates, and key establishment sessions recorded in transit are at retroactive risk.
- The effective deadline for protecting data with long-term sensitivity has already passed — the threat is present, not future.

2. State of the Art in Quantum Hardware

2.1 The Road to a Cryptographically Relevant Quantum Computer

ToexecuteShor's algorithm against RSA-2048, a quantum computer requires approximately 4,000 error-corrected logical qubits. Due to the overhead of quantum error correction codes, current estimates translate this to roughly 1 million physical qubits with today's error rates — a number that shrinks as error correction improves.

Progress benchmarks as of early 2026:

- IBM “Nighthawk” research programs explore **circuits with thousands of gates on ~100+ qubits**
- Google Willow (2024): demonstrated quantum error correction below threshold for the first time — a critical theoretical milestone for scalable fault-tolerant quantum computing.
- Google Security (May 2025): revised down the qubit requirement for breaking RSA-2048 from prior estimates, citing algorithmic improvements in surface code compilation.

The trajectory is not linear, but the directional trend is unambiguous: each generation of quantum hardware roughly doubles the effective qubit count, and algorithmic improvements continue to reduce the resource threshold for cryptographically relevant attacks.

2.2 Industry Consensus on Timeline

AnApril 2025 survey commissionedby TheEconomist Impact, covering quantum professionals across the UK, Europe, North America, and Asia, reported that 83% of respondents expect quantum utility — the ability to perform commercially relevant computations beyond classical capability — within 10 years.

This aligns with the CNSA 2.0 timeline established by the NSA and NIST: mandatory exclusive use of post-quantum algorithms for National Security Systems by 2033. This is not a precautionary target — it is a calibrated engineering deadline based on threat modeling.

PLANNING HORIZON	*If a device under development today has a projected operational life of 7 or more years, it will be operational during the estimated CRQC availability window. The design choices made in the next 12–18 months will determine whether that device remains secure or becomes a liability.*

3. NIST Post-Quantum Cryptography Standards

3.1 Finalized FIPS Standards (August 2024)

InAugust 2024,NISTpublishedthefirstfinalizedPost-Quantum Cryptography standards, concluding a standardization process that began in 2017. Three algorithms were standardized:

FIPS 203 — ML-KEM (Module-Lattice Key Encapsulation Mechanism)

Formerly known as Kyber. ML-KEM is the primary quantum-safe key establishment algorithm and the direct replacement for ECDH and RSA-OAEP in key exchange protocols. Security is based on the hardness of the Module Learning With Errors (MLWE) problem which is widely believed to be resistant to both classical and quantum attacks.

- Three security levels: ML-KEM-512 (~128-bit), ML-KEM-768 (~192-bit), ML-KEM-1024 (~256-bit).
- CNSA 2.0 mandates ML-KEM-1024 for NSS key establishment.
- Known implementation concern: KyberSlash timing side-channel — require constant-time hardware implementations.

FIPS 204 — ML-DSA (Module-Lattice Digital Signature Algorithm)

Formerly known as Dilithium. ML-DSA replaces ECDSA and RSA-PSS for digital signatures, including firmware signing and certificate authority operations. Security rests on both MLWE and the Module Short Integer Solution (MSIS) problem.

- ML-DSA-44, ML-DSA-65, ML-DSA-87
  CNSA 2.0 aligns with the highest security level (Level V), corresponding to ML-DSA-87.
- Signature size: 2,420–4,595 bytes — significantly larger than ECDSA (64 bytes). Boot ROM and flash storage requirements must account for this.
- Side-channel exposure is increased due to the number of programmable parameters; hardware acceleration with DPA countermeasures is strongly recommended.

FIPS 205 — SLH-DSA (Stateless Hash-Based Digital Signature Algorithm)

Formerly SPHINCS+. A hash-based signature scheme whose security reduces entirely to the security of the underlying hash function (SHA-256 or SHAKE). Unlike ML-DSA, it makes no lattice security assumptions — providing algorithmic diversity as a hedge against potential future lattice vulnerabilities.

- Not included in CNSA 2.0 — use as a secondary/diversity option alongside ML-DSA, not as a primary replacement.
- Signature size at security level 3: 16,224 bytes — suitable for code signing contexts where transmission overhead is less constrained.

FN-DSA (FALCON) — Candidate Algorithm (Expected Future Standardization)

FN-DSA (based on FALCON) is under consideration for standardization but **has not yet been finalized as a FIPS standard**.

HQC (Upcoming Standard)

NIST has selected **HQC (Hamming Quasi-Cyclic)** as an additional KEM to complement ML-KEM, providing **algorithmic diversity**.

3.2 Stateful Hash-Based Signatures (CNSA 2.0 Approved)

NIST SP 800-208 standardizes LMS (Leighton-Micali Signature Scheme) and XMSS (eXtended Merkle Signature Scheme). Both are approved in CNSA 2.0 for firmware signing. Security requires careful state management — signing the same key state twice catastrophically undermines security. Appropriate for secure boot signing chains with disciplined key management infrastructure.

3.3 Algorithm Comparison for Device Designers

Algorithm	Pub. Key (B)	Sig. Size (B)	Perf.	CNSA 2.0 Status	Recommendation
RSA-2048	256	256	Fast (HW)	**Deprecated / transition away**	**Migrate away — quantum -vulnerable**
Elliptic Curve P-256	33	64	Fast (HW)	**Deprecated / transition away**	**Migrate away — quantum -vulnerable**
ML-KEM-768	1,184	N/A (KEM)	Moderate	**CNSA-aligned (L5 required for NSS)**	**Primary choice for IoT / embedded key exchange**
ML-DSA (Dilithium L3)	1,952	3,309	Moderate	**CNSA-aligned (L5 required for NSS)**	**Primary choice for device signing / secure boot**
SLH-DSA (SPHINCS+)	48	16,224	Slow	**FIPS 205 (not CNSA primary)**	**Algorithm diversity / high-assurance fallback**
FN-DSA (FALCON)	897	752	Moderate	**FIPS 206 pending**	**Future optimization (size-sensitive use cases)**
LMS / XMSS	60–64	2,512–2,692	Moderate	**CNSA 2.0 approved (stateful)**	**Suitable for firmware signing with strict key management**

4. The Regulatory Countdown: CNSA 2.0 Timeline

The NSA's Commercial National Security Algorithm Suite 2.0 (CNSA 2.0) provides the most authoritative migration timeline for cryptographic transitions. While CNSA 2.0 is framed around National Security Systems, it reliably sets the compliance trajectory that cascades into commercial, industrial, and regulated IoT markets.

Year	Milestone
2025	CNSA 2.0 algorithms should be supported and preferred
2030	PQC required for software/firmware signing and traditional networking
2033	PQC required for OS, web, cloud, and general NSS
2035	Full transition target for National Security Systems

Device makers supplying to defense, critical infrastructure, healthcare, financial services, and telecommunications sectors should treat these dates as hard deadlines — the CNSA 2.0 timeline will propagate into procurement specifications, certification requirements (Common Criteria, FIPS 140), and contractual obligations.

5. The Device Maker's Unique Exposure

5.1 The Long-Lifecycle Problem

Enterprise software operates on an updatecadence of months. Embedded hardware operates on a lifecycle of years to decades. A secure microcontroller designed today for an industrial controller, a medical device, or a smart meter may ship in 2027 and remain operational in 2037 — well within the CRQC availability window.

This creates a structural asymmetry: the cryptographic algorithm baked into a hardware root of trust at tape-out cannot be patched after deployment. Secure boot keys, device attestation certificates, and provisioning chains embedded in immutable ROM are permanent. If those algorithms are quantum-vulnerable, the device is permanently vulnerable.

⚠ Any device currently in design using RSA-2048 or P-256 for secure boot verification, device attestation, or firmware signing that will still be in operation by 2030 is on a trajectory toward non-compliance and cryptographic vulnerability. This is not a hypothetical risk — it is a design flaw being committed today.

5.2 Secure Boot: The Critical Vulnerability

Thesecure boot chain is the mostcriticalandleastflexible cryptographic component in any embedded system. It typically involves:

1. A root public key burned into OTP (one-time programmable) memory or ROM at manufacturing.
2. Firmware images signed with the corresponding private key at build time.
3. Boot ROM verification of the signature against the burned root key before any code execution.

Each of these three elements presents a distinct migration challenge in the post-quantum context:

- Root key in OTP/ROM: immutable after manufacturing. If it is an RSA or ECC key, it cannot be replaced without hardware intervention or a new device SKU.
- Firmware signing infrastructure: must support larger PQC signature sizes (ML-DSA at 3,309 bytes vs. ECDSA at 64 bytes). Boot ROM size, flash layout, and verification timing budgets all require re-evaluation.
- Verification performance: ML-DSA verification on an ARM Cortex-M4 requires approximately 2,415 kcycles — roughly 17× more than ECDSA verification. Devices with tight boot time SLAs must benchmark carefully.

5.3 The Memory and Bandwidth Impact

Post-quantum algorithmsimposeconcretehardware resource requirements that device architects must plan for:

- Public keys: ML-KEM-768 public keys are ~1,184 bytes; ML-KEM-1024 ~1,568 bytes; ML-DSA-65 public keys are 1,952 bytes. Compare with 33 bytes for P-256. Certificate chains embedding these keys require larger flash and RAM allocations.
- Signatures: ML-DSA-65 signatures are 3,309 bytes vs. 64 bytes for ECDSA P-256. Firmware images carrying embedded signatures, OTA update packages, and TLS handshake buffers all require resizing.
- RAM overhead: KEM operations require additional working memory for key generation and encapsulation. Constrained devices (< 32KB RAM) require careful profiling.
- TLS/DTLS handshakes: Initial handshake size increases significantly when using ML-KEM for key exchange and ML-DSA for authentication — UDP-based protocols may require fragmentation handling.

5.4 The Legacy Hardware Problem

A substantial installed base of devices already in the field will never receive a PQC-capable firmware update. For product lines with active maintenance, a triage framework is needed:

- Category A — Updateable, crypto-agile: Devices with software-configurable cryptographic libraries where OTA updates are possible. Prioritize ML-KEM and ML-DSA library integration in next firmware release.
- Category B — Updateable, hardware-limited: Devices where firmware can be updated but hardware crypto accelerators only support classical algorithms. Implement software PQC libraries; accept performance penalties. Plan hardware transition for next product generation.
- Category C — Non-updateable: Devices where firmware signing keys are burned to ROM and cannot be migrated. Implement network-layer compensating controls (PQC at gateway/proxy layer). Accelerate end-of-life planning.

6. Migration Framework: A Technical Roadmap

6.1 Step 1 — Cryptographic Inventory

Beforeanymigrationcan begin,device teamsmust produce a complete cryptographic bill of materials (CBOM). This is analogous to a software bill of materials (SBOM) but focused on cryptographic primitives:

- Enumerate every cryptographic operation in firmware: key generation, signing, verification, key exchange, encryption, hashing, random number generation.
- Identify the algorithm, key size, and library/hardware implementation for each operation.
- Map each operation to the threat model: is it protecting data in transit, at rest, or in use? What is the required data confidentiality horizon?
- Classify by replaceability: is the algorithm in ROM, in updateable flash, or in a configurable software stack?

6.2 Step 2 — Crypto Agility Architecture

Crypto agility is the designprinciple of building systems that can switch cryptographic algorithms without hardware replacement. It is the single most important architectural investment a device maker can make for long-term security. Implementation approaches, in order of robustness:

4. Software cryptographic libraries with runtime algorithm selection: appropriate for application processors with sufficient RAM and compute. Libraries such as liboqs (Open Quantum Safe) provide production-ready ML-KEM, ML-DSA, and SLH-DSA implementations.
5. FPGA-based cryptographic cores: allows algorithm reconfiguration post-deployment. Suitable for mid-range industrial controllers and networking equipment where FPGA fabric is already present.
6. Embedded FPGA (eFPGA) fabrics in SoC designs: purpose-built for this use case. An eFPGA fabric co-designed with a security processor allows cryptographic algorithm updates via bitstream OTA — combining the performance of hardware acceleration with the flexibility of software reconfiguration.

For constrained microcontrollers without FPGA fabric, the path to crypto agility is architectural: design the secure boot chain to support a secondary signature algorithm alongside the primary, and implement the secondary in updateable flash. This creates a migration path when the primary algorithm is deprecated.

6.3 Step 3 — Hybrid Cryptography

During thetransitionperiod —whilenew PQC algorithms are gaining operational trust and the installed base of classical infrastructure is being migrated — hybrid cryptography provides the

optimal security posture. A hybrid scheme combines a classical algorithm with a PQC algorithm such that security is maintained if either remains unbroken.

Formally: a hybrid key exchange produces a shared secret K = KDF(K_classical || K_pqc). An adversary must simultaneously break both the classical and PQC algorithms to recover K.

- Signal Protocol's PQXDH: combines X25519 (ECDH) with ML-KEM-1024 (Kyber) — the most widely deployed PQC hybrid in production messaging infrastructure.
- Google FIDO2 hybrid: combines ECDSA P-256 with ML-DSA for hardware security key attestation.
- TLS 1.3 hybrid: RFC drafts for X25519Kyber768 are in active IETF standardization and supported in Chrome, Firefox, and major TLS libraries.

For device firmware signing, a dual-signature approach — embedding both an ECDSA P-256 and an ML-DSA-65 signature in the firmware header — allows verification against either or both algorithms depending on the verifier's capability. This enables a graceful transition of the installed base without forking firmware images.

6.4 Step 4 — Risk Prioritization

Not all cryptographic operations carry equal urgency. A structured prioritization framework based on three factors:

- Data sensitivity horizon: How long must the data protected by this operation remain confidential? State secrets, medical records, and financial data may require 20+ year confidentiality — these operations have the highest HNDL risk and must be migrated first.
- Algorithm vulnerability: RSA and ECC are fully broken by Shor's — immediate priority. AES-128 is weakened — upgrade key size, moderate priority. SHA-256 — acceptable for near-term, low urgency.
- Replaceability: ROM-burned operations must be addressed in next hardware revision — highest urgency for design teams. OTA-updateable operations can be deferred but not ignored.

A practical priority stack for embedded device teams:

7. Secure boot root key and signing infrastructure — migrate to ML-DSA-65 or LMS in next hardware tape-out.
8. Device attestation and provisioning key establishment — migrate to ML-KEM-1024 + hybrid.
9. TLS/DTLS client stack — integrate PQC hybrid key exchange in next firmware update cycle.
10. Certificate authority infrastructure — migrate PKI to ML-DSA-based certificates.

11. Symmetric key upgrade — AES-128 → AES-256, SHA-256 → SHA-384 where data has long-term sensitivity.

7. SEALSQ Post-Quantum Security Solutions

7.1 Hardware Security Platforms

SEALSQdevelopssecuresemiconductorsolutions designed to support the transition toward post-quantum cryptography, including:

- **QS7001 Secure Element**
  Designed to support **post-quantumcryptographicalgorithmssuchasML-KEMand ML-DSA**, with hardware-level protections against side-channel attacks.
- **QVault TPM (Trusted Platform Module)**
  A next-generation TPM platform integrating **quantum-resistant cryptographic capabilities**, aligned with evolving standards and enterprise security requirements.

7.2 Cryptographic Capabilities

SEALSQsolutions aredesignedwith:

- Support for **hybrid cryptography (classical + PQC)**
- Hardware acceleration roadmap for **lattice-basedalgorithms**
- Secure key storage and device identity rooted in hardware

7.3 Certification Roadmap

SEALSQ structures its certification strategy around a combination of hardware and software security certifications, aligned with international standards for high-assurance embedded systems.

SEALSQ’s 2026 certification roadmap covers its QS7001 Secure Element and QVault TPM software stack, with a complementary approach:

Hardware Certification (Secure Element)

The QS7001 Secure Element is evaluated as a hardware root of trust and targets:

Common Criteria EAL5+ certification, providing high assurance against physical and logical attacks

Certification focused on the hardware platform and its security architecture

Software Certification (TPM Stack)

The QVault TPM, which runs on the QS7001 secure element, is the component targeted for:

FIPS 140-3 validation, as a cryptographic module implemented in software

Alignment with Trusted Computing Group (TCG) TPM standards

For the QVault TPM product line (including TPM-183 and TPM-185), SEALSQ targets:

- FIPS 140-3 lab submission to NIST in September 2026
- TCG certification in October 2026

Combined Certification Model

SEALSQ's approach enables a combined evaluation model, where:

- The hardware (QS7001) targets certification under Common Criteria (EAL5+)
- The software (QVault TPM) targets validation under under FIPS 140-3

This integrated approach allows customers to build solutions on a pre-evaluated hardware platform with a FIPS-validated cryptographic module, significantly accelerating their own certification processes.

Customer Benefit

By leveraging SEALSQ's certified hardware and FIPS-validated TPM implementation, device manufacturers can:

- Reuse a pre-certified root of trust
- Reduce the scope and cost of their own certification efforts
- Facilitate compliance with regulatory requirements in sectors such as defense, critical
- infrastructure, and industrial IoT

Post-Quantum Integration

In parallel, SEALSQ's roadmap includes the progressive integration of post-quantum cryptographic algorithms into both:

- The hardware platform (QS7001)
- The TPM software stack

Important clarification:

Current certification schemes (FIPS, Common Criteria) are still primarily based on classical cryptography. Post-quantum capabilities are being integrated progressively and are not yet fully reflected in standardized certification frameworks.

8. Actionable Recommendations

The following recommendations are structured by immediacy. Each action is executable within the specified timeframe regardless of where an organization currently sits in its PQC migration journey.

Immediate Actions (0–6 months)

12. Produce a cryptographic inventory (CBOM) of all products currently in design, under development, and in active maintenance. Classify each cryptographic operation by algorithm, replaceability, and data sensitivity horizon.
13. Evaluate all hardware tape-outs planned in the next 18 months. Any product that includes RSA or ECC in ROM or OTP for secure boot purposes should be redesigned to incorporate ML-DSA or a hybrid dual-signature scheme.
14. Assess flash and RAM headroom in existing designs. Confirm that storage and memory budgets can accommodate PQC signature sizes (plan for 4–8KB for signature and public key material in secure boot contexts).
15. Engage SEALSQ to assess cryptographic architectures and define a risk-scored migration roadmap

Near-Term Actions (6–18 months)

16. Integrate post-quantum hybrid TLS into firmware for all devices capable of receiving OTA updates. Use X25519+ML-KEM-768 for key exchange and begin evaluation of ML-DSA for mutual authentication.
17. Migrate internal signing infrastructure. Code-signing servers, build systems, and firmware release pipelines should transition to ML-DSA signing alongside existing ECDSA, producing dual-signature firmware images.
18. Engage SEALSQ secure elements or TPM platforms (e.g., QS7001, QVault TPM) for hardware-accelerated PQC integration.
19. Begin quantum-safe PKI enrollment. Issue hybrid or post-quantum-ready X.509 device identity certificates using compatible PKI infrastructure aligned with emerging PQC standards.

Strategic Actions (18–36 months)

20. Target full CNSA 2.0 compliance for all new product tape-outs by 2028. Root of trust, firmware signing, and key establishment must be exclusively PQC or hybrid PQC by this date to meet the 2030 deprecation deadline.
21. Define an end-of-life or compensating control strategy for all Category C (non-updateable) devices. Document and communicate to customers.
22. Prepare for FIPS 140-3 and Common Criteria certification requirements as PQC-enabled hardware platforms mature.

9. Conclusion

The cryptographic transition to post-quantum security is not a future engineering problem. It is a present-day design imperative with a fixed regulatory and threat-model deadline. The algorithms that will be broken are already known. The standards that replace them are already finalized. The timeline for mandatory migration is published and contractually binding for an expanding set of markets.

For device makers, the urgency is compounded by the inescapable reality of hardware lifecycles: decisions made in the design lab today will govern the security posture of devices that will still be operating in 2035. The window for addressing this at the architecture level — before ROM masks are committed, before production keys are burned, before millions of units ship — is measured in quarters, not years.

The good news is that the technical solution set is mature, standardized, and available. NIST FIPS 203, 204, and 205 provide production-ready algorithm specifications. Hardware implementations with appropriate side-channel countermeasures are available today. Hybrid approaches enable a graceful transition without sacrificing backward compatibility.

With purpose-built secure elements, TPM platforms, hardware security capabilities, and integration support, SEALSQ helps device makers transition from quantum-vulnerable to quantum-ready architectures.

NEXT STEP *Contact SEALSQ to schedule a cryptographic assessment of your product portfolio.*

References and Further Reading

The following sources and standards informed this white paper:

- NIST FIPS 203 — Module-Lattice-Based Key-Encapsulation Mechanism Standard (ML-KEM), August 2024. csrc.nist.gov/pubs/fips/203/final
- NIST FIPS 204 — Module-Lattice-Based Digital Signature Standard (ML-DSA), August 2024. csrc.nist.gov/pubs/fips/204/final
- NIST FIPS 205 — Stateless Hash-Based Digital Signature Standard (SLH-DSA), August 2024. csrc.nist.gov/pubs/fips/205/final
- FIPS 206 — draft / pending standardization— Fast-Fourier Lattice-Based Digital Signature Standard (FN-DSA)
- NSA CNSA 2.0 Cybersecurity Advisory, September 2022. media.defense.gov
- NIST SP 800-208 — Recommendation for Stateful Hash-Based Signature Schemes (LMS, XMSS). csrc.nist.gov
- NIST IR 8545 — Status Report on the Fourth Round of the NIST PQC Standardization Process, March 2025. csrc.nist.gov/pubs/ir/8545/final
- Google Security Blog — 'Tracking the Cost of Quantum Factoring,' May 2025. security.googleblog.com
- IBM Newsroom — 'IBM Sets the Course to Build World's First Large-Scale, Fault-Tolerant Quantum Computer,' June 2025.
- Economist Impact / Inside HPC — 'Survey: 83% Say Quantum Utility to Be Achieved Within a Decade,' April 2025.
- Open Quantum Safe Project — liboqs Algorithm Benchmarks. openquantumsafe.org/liboqs/algorithms/
- Signal Foundation — 'PQXDH Key Agreement Protocol.' signal.org/blog/pqxdh/
- Rambus Inc. — 'Preparing for the Quantum Era: Practical Approaches to Quantum Safe Cryptography,' Scott Best, 2026.


SEAL SQ
semiconductors + quantum