Beyond the Limits of PQC

Blind, Fragmented, and Out of Time: The Crypto-Agility Gap in PQC Migration

Markus Rautell

BUSINESS FINLAND

Introduction

Motivation

- Quantum computing amplifies the long-recognized fragility of cryptographic longevity.
- Past transitions (e.g., MD5, SHA-1) were slow and incomplete, exposing systemic challenges in retiring weakened primitives.
- Security collapses rapidly when infrastructures cannot remove vulnerable mechanisms.
- Policy and standards bodies now enforce firm deprecation timelines, making routine cryptographic replacement an operational necessity.
- Real infrastructures may require 15+ years to transition due to deeply embedded cryptography, hardware refresh cycles, and supply-chain constraints.
- PQC migration highlights a structural weakness: global infrastructures lack the agility required for large-scale cryptographic updates.
- Without standardized abstraction and policy layers, organizations build bespoke and heterogeneous solutions that increase complexity and undermine interoperability.
- NIST stated in 2016 that agencies should be ready to replace current public-key algorithms within about 10 years and that maintaining crypto-agility was imperative [1]; however, few organizations acted on this guidance.

Core Concept

- **Capability**: Replace, update, or retire cryptographic mechanisms with minimal operational impact.
- **Visibility**: Accurate cryptographic inventories and CBoMs provide the asset awareness required for safe and sequenced transitions.
- **Modularity**: Algorithm-agnostic application design, clear abstraction layers, and policy-driven interfaces enable practical technical agility.
- **Governance**: Coordinated workflows and enterprise-level change management make transitions repeatable and sustainable.

Real-World Limitations and Research Gaps

The Invisible and the Undefined: Why Agility Fails Before It Begins

- No unified model of crypto-agility exists; definitions and architectural assumptions diverge across the literature, creating inconsistent design paradigms [2, 3].
- Ecosystem layers—including standards bodies, vendors, and enterprises—evolve at different speeds, producing structural misalignment that prevents coordinated PQC migration [4].
- Organizations frequently lack cryptographic inventories, dependency visibility, and governance structures, resulting in outdated and fragmented knowledge of deployed cryptographic assets [5, 6].
- Discovery practices remain immature: despite emerging standards such as CycloneDX [7, 8], tools vary in accuracy, lack automation, rarely differentiate initial discovery from continuous monitoring, and no widely adopted solutions support heterogeneous environments [3, 9].
- Visibility gaps across firmware, proprietary hardware, legacy middleware, and network equipment prevent reliable sequencing of system-wide cryptographic transitions [9, 10].
- No system-level mechanisms track migration issues across software stacks, hardware dependencies, protocols, and operational workflows, leaving organizations blind to end-to-end bottlenecks [11].
- Even when individual protocols become crypto-agile, the ecosystem lacks a unified architectural model to coordinate them, leaving migrations fragmented and undermining cross-layer PQC readiness [11].

Organizational Drag: The Migration Killer

- Cryptographic maintenance is routinely deprioritized under operational pressure; training burdens, distributed deployments, and logistical constraints lead to improvised and prolonged transitions that elevate security and operational risk [12].
- Enterprises lack practical orchestration mechanisms for transitioning non-agile, continuously operating systems, even though maturity models [13], risk frameworks [14], and high-level roadmaps [15, 16, 17] exist.
- Enterprise frameworks offer limited support for system-wide transitions, lacking structured processes for staged rollout, coordinated testing, rollback, and non-disruptive evolution across heterogeneous environments [5].
- Developers lack confidence in performing cryptographic updates, perceiving them as risky; this leads to delayed migrations, improvised fixes, and a persistent fear of breaking security [18].
- Most real-world cryptographic vulnerabilities stem from incorrect API use, insecure parameters, and misconfiguration; this indicates that developer knowledge gaps, not primitive weaknesses, remain a primary barrier to safe cryptographic evolution [19].
- PQC transition amplifies these human and organizational barriers: larger parameters, hybrid deployments, and legacy dependencies increase the complexity of non-agile environments [18, 20].

Project is funded by Business Finland, see www.pqc.fi for further information.

The Technical Debt Wall: Architecture and Tooling Failures

- Cryptographic mechanisms remain tightly coupled to application logic, and systems lack the abstraction layers, versioned metadata, and algorithm identifiers needed to support safe and controlled algorithm substitution [6, 3].
- Encrypted-data migration is obstructed by ciphertext formats that lack versioning or provenance markers, causing legacy encrypted data to accumulate as technical debt [3].
- Agile development and CI/CD pipelines fall short of cryptographic testing needs; they lack crypto-specific regression tests, downgrade-resistance checks, and structured rollout/rollback support [21, 6, 3].
- Testing support for PQC migration is inadequate: existing methods lack formal models of migration mechanisms, downgrade-attack analysis, and structured validation frameworks for assessing agility schemes [22].
- Developer tooling remains limited; automated discovery mechanisms, safe algorithm-substitution libraries, and end-to-end crypto-testing harnesses are largely absent [6].
- No comprehensive catalog of PQC migration challenges exists, hindering issue tracking, maturity assessment, and validation at scale [11].
- Actionable engineering methodologies and operational migration procedures remain missing, limiting the translation of research into deployable transition practices [3].

The Clock Is Ticking: External Forces Amplifying the Transition Gap

- PQC migration is estimated to require 2-15+ years depending on enterprise size [23, 24], conflicting with NIST and EU transition timelines [25, 17, 26].
- Aggressive early-2030s government deadlines leave little slack for legacy infrastructures that struggle with synchronized, large-scale updates [27].
- Migration costs estimates range from 250k$ to over 1B$, scaling with organizational size, system complexity, and required modernization efforts [24, 27].
- Vendors often treat cryptographic agility as a cost center without near-term benefit, creating economic and policy disincentives for timely migration [22].
- Limited international coordination leads countries to progress at uneven speeds, risking interoperability failures and inconsistent security guarantees across jurisdictions [27].
- Existing PQC guidance focuses on preparation and deployment, with limited support for detecting quantum-enabled compromise, responding to cryptographic failures, or recovering from hybrid-interoperability issues [28].
- Organizations lack a shared operational knowledge base. Early adopters report degradation, integration failures, and incompatibilities, but these lessons remain isolated, forcing others to rediscover the same pitfalls [28].
- Fallback, rollback, and recovery mechanisms for PQC failures remain largely unexplored [28].

The Missing Discipline of Cryptographic Transitions

- Research remains centered on primitives, proofs, and protocol behavior, while the engineering mechanics of real-world cryptographic transitions remain largely underdeveloped [29].
- Operational tasks essential for transitions (dependency management, multi-stakeholder coordination, and system evolution) receive far less attention than algorithm- and protocol-centric work [29].
- The field lacks lifecycle-oriented frameworks that treat agility as a system-wide engineering problem; few models explain how to plan, sequence, and execute transitions across heterogeneous infrastructures [29].
- Fragmentation across research domains (algorithm design, protocol evolution, systems engineering, governance) inhibits the development of integrated, end-to-end migration methodologies [4].
- Although agility is studied across technical, operational, organizational, and ecosystem layers, these insights have not yet matured into unified, transition-focused engineering approaches [4].

Closing the Transition Gap

- **We must establish shared models of agility and deploy automated, open-source cryptographic inventory and CBoM tooling so organizations can finally see what must be migrated.**
- **Organizations need coordinated governance, repeatable workflows, and trained teams to plan and execute cryptographic change at scale.**
- **Systems must adopt standardized crypto-agile abstraction and policy layers, supported by modular interfaces, versioned formats, and tooling for safe substitution and continuous transition.**
- **Migration planning must accelerate now, as long transition timelines and early-2030s deadlines demand immediate investment and international coordination.**
- **The field must develop a dedicated engineering discipline for cryptographic transitions, with lifecycle frameworks, metrics, and validated migration methodologies.**

Contact: Markus Rautell, Research Scientist
Tel. +358 50 531 5094, markus.rautell@vtt.fi