Post Quantum Cryptography (PQC) Transition

A Systems Engineering Imperative

Dr. Ron Ross

RONROSSECURE, LLC

1. Introduction

Post-quantum cryptography (PQC) transition is almost universally framed as a cryptography problem: find the vulnerable algorithms, swap them for quantum-resistant ones, and ensure everything is working correctly. That framing is logical but fails to address the foundational architectural issues that ensure the composite system is trustworthy secure and missions are resilient. And it is why many organizations will find themselves vulnerable even after they have finished migrating every algorithm in their inventory.

The real problem is a systems engineering problem. Cryptography does not operate in isolation. It operates within an ecosystem of complex, interconnected, decades-old systems that were not designed with *trustworthiness* as an engineering outcome. It's called technical debt. Replacing traditional algorithms without addressing the underlying system architecture is, to borrow a phrase, implementing the "gold standard" of cryptographic protection on an underlying house of cards.

> *The 2020 SolarWinds breach proved the point. Cryptographic code-signing was in place and was bypassed entirely. Attackers compromised the build pipeline upstream; the cryptography worked exactly as designed — and was irrelevant. The surrounding system was not engineered to be trustworthy.*

NIST Special Publication 800-160, Volume 1, *Engineering Trustworthy Secure Systems*, provides the conceptual framework, system lifecycle process, and security design principles for understanding why PQC transition must be approached as a systems engineering effort and what that approach requires in practice.

2. Security as an Emergent System Property

The foundational principle of NIST SP 800-160 is that security is not a feature added to a system. It is an *emergent property* of a well-engineered system, arising from the deliberate application of sound engineering principles throughout the system lifecycle, from concept development through disposal. Like safety in an airplane or reliability in a bridge, it cannot be bolted on after the fact.

Many legacy systems violate this basic principle by design. They were built in an era when security was treated as a stovepipe activity — a set of controls applied to a system to achieve a level of compliance rather than a property that emerged from an engineered-system directly related to mission objectives and stakeholder protection needs. PQC transition, as currently envisioned, inherits this fundamental

© 2026 RONROSSECURE, LLC | CC BY 4.0.

deficiency. Organizations are applying new cryptographic algorithms to system architectures that were never engineered to support the security protections those algorithms are intended to provide.

Until that underlying system architecture is addressed, PQC migration is a necessary step, but it is not a sufficient one. Organizations that complete algorithm migration and consider the problem solved will have a false sense of security that may prove more dangerous than the acknowledged vulnerability.

3. Security Design Principles

NIST SP 800-160, Volume 1 describes a comprehensive set of security design principles applicable across all phases of the system lifecycle. Several of these design principles bear directly on why PQC transition must be treated as a systems engineering effort rather than an algorithm substitution exercise.

Domain Separation

Systems should be designed to separate different security domains and enforce the boundaries between those domains, preventing unauthorized information flow or interaction between them. Domain separation has direct and specific implications for PQC transition. During migration, agencies must simultaneously operate traditional and post-quantum cryptographic environments, a hybrid state in which domain boundaries are especially critical. Systems that have completed PQC migration must be architecturally separated from systems that have not, with enforced boundaries that prevent a compromise in a legacy traditional-algorithm domain from propagating into a post-quantum domain. Domain separation also applies to key management: the domain where keys are generated and stored must be strictly separated from the domain where they are used, and both must be separated from general-purpose computing environments. Without enforced domain boundaries, the security properties of PQC algorithms cannot be fully realized regardless of their cryptographic strength.

Least Privilege

Systems should be designed so that each system element has only the access, privileges, and authorizations necessary to perform its intended function. Applied to PQC transition, cryptographic keys and operations must be accessible only to the system components that require them. Many legacy systems violate this principle. Keys are broadly accessible, processes carry excessive permissions (especially with agentic AI), and there is no architectural enforcement of access boundaries. Migrating to ML-KEM, SLH-DSA, or ML-DSA in these environments may place stronger algorithms into a system architecture that does not protect them.

Mediated Access

Systems should be designed so that all access to system elements is mediated by an authorized and trusted access control mechanism — no direct, unmediated access is permitted. Mediated access is foundational to PQC transition because cryptographic keys and operations are precisely the class of high-value asset this principle targets. In most legacy system architectures, cryptographic services are accessed directly by applications, with no interposed control mechanism enforcing who can request a cryptographic operation, under what conditions, and with what level of accountability. PQC migration is the opportunity to correct this. A well-engineered post-migration architecture routes all cryptographic operations through a trusted mediation layer, one that enforces policy, logs access and can be audited.

© 2026 RONROSSECURE, LLC | CC BY 4.0.

Without mediated access, even the strongest post-quantum algorithms operate in an environment where their use cannot be controlled, monitored, or verified.

Distributed Privilege

Systems should require multiple entities to collaborate or act in concert to achieve high-consequence or high impact operations. Applied to PQC transition: key generation, escrow, rotation, and revocation are precisely the kind of high-consequence/high impact operations this principle targets. Architectures that allow a single process or administrator to perform these critical operations unilaterally are structurally vulnerable, regardless of which algorithms are in use. PQC migration must include strong architectural enforcement of distributed privilege for cryptographic operations.

Defense In Depth

Systems should employ multiple layers of protection so that the failure of one protective measure does not result in severe or catastrophic system compromise. A PQC migration strategy focuses primarily on strengthening the cryptographic layer of protection. If that layer is compromised, and there are no strong structural safeguards to limit and contain the damage, critical system operations and functions will be at risk with potential mission failure. Cryptography must be one layer among many, not the sole line of defense. Defense in depth requires additional architectural protections be in place that remain effective even when cryptographic protections are circumvented.

Least Functionality

Systems should provide only the functionality necessary to support the intended use and mission, and no more. PQC transition paradoxically expands the system attack surface during migration. Systems must operate in hybrid mode, running both traditional and post-quantum algorithms simultaneously for interoperability. This introduces additional protocol versions, compatibility shims, and implementation variants, each of which must be trusted and maintained. Applying the least functionality principle to the hybrid transition means actively eliminating unnecessary algorithm support, deprecated protocol versions, and unused cryptographic services as migration progresses rather than accumulating them.

Protective Defaults

Systems should be designed so that their default configuration and behavior represent the most protective and secure options. Systems are frequently deployed without being fully configured to meet security objectives. During PQC migration, the risk is that systems default to traditional algorithm fallback for compatibility, defeating the migration entirely for connections that negotiate down. Protective defaults require that PQC algorithms be the default, traditional algorithm fallback be an explicit and logged exception, and the migration state of each system component be actively tracked rather than assumed.

Protective Failure

Systems should be designed to fail into a secure (or known) state that protects assets from loss when failures or other adversities occur. Legacy systems frequently fail open — a component failure results in degraded security rather than orderly denial. PQC migration into systems that fail open inherits this vulnerability. Stronger algorithms in an architecture that fails in an insecure or unknown state do not provide the real-world security they appear to provide on paper. Every failure mode of the PQC-migration architecture must be analyzed to ensure it does not create exploitable conditions.

© 2026 RONROSSECURE, LLC | CC BY 4.0.

> *Each principle violated in the underlying system architecture is a gap that PQC algorithm migration cannot close. Algorithm strength and system trustworthiness are not the same thing.*

4. Trustworthy Secure Systems

NIST SP 800-160 defines a *trustworthy secure system* as one that is dependable, safe, reliable, and secure — not because of the controls applied to it, but because of the engineering principles applied in building it. Trustworthiness is not a compliance state. It is a design outcome.

This distinction is critical for PQC transition. An organization can fully migrate to ML-KEM, ML-DSA, and SLH-DSA, achieve 100 percent algorithm compliance, satisfy every audit requirement, and still operate systems that are not trustworthy — because the underlying architecture has not been engineered to the design principles that result in a trustworthy secure system. The cryptography is stronger but the system is not.

Trustworthy secure systems also have a property that most systems in use today lack: resilience to unanticipated threats. A system engineered to the security design principles of domain separation, mediated access, defense in depth, least privilege, least functionality, distributed privilege, protective failure, and protective defaults — degrades gracefully under novel attack conditions. A system that relies on its cryptographic layer without supporting architectural and structural defenses, may fail catastrophically when that layer is breached, regardless of which algorithms are in use.

This is not a theoretical concern. The class of AI-driven cyberattacks now emerging, characterized by autonomous zero-day exploitation, multi-stage attack chains, and rapid low-cost execution, is precisely the class of threat capable of identifying and exploiting architectural weaknesses that PQC migration leaves unaddressed. Organizations that complete algorithm migration and consider their work done will discover, at the worst possible moment, that they have addressed only part of the problem.

5. Mission Resilience: The Engineering Objective

NIST SP 800-160 Volume 2 introduces cyber resiliency engineering as the discipline concerned with ensuring that systems can anticipate, withstand, recover from, and adapt to adverse conditions, including conditions involving sophisticated, persistent adversaries. SP 800-160, Volume 2 is used in conjunction with Volume 1 and provides a four-goal framework and model for mission resilience. It reveals why PQC transition alone is insufficient.

Anticipate. Mission-resilient systems understand how adversaries might defeat their defenses, including through non-cryptographic means. A PQC-centric strategy does not anticipate the class of supply chain, firmware, or architectural attacks that bypass cryptographic protections entirely. The *anticipate* goal requires systems engineering that models adversary behavior across the full attack surface, not just the cryptographic layer.

Withstand. Systems must continue to provide essential mission functions while under active attack. Cryptographic protection supports the *withstand* goal only if the surrounding system architecture can sustain mission functions when specific components are compromised. Systems not engineered to the

© 2026 RONROSSECURE, LLC | CC BY 4.0.

NIST SP 800-160, Volume 1 design principles (e.g., those that rely on a single defensive perimeter), generally cannot withstand sophisticated multi-stage attacks.

Recover. Systems must be able to absorb an attack and continue to operate, even in a degraded or debilitated state. The *recover* goal is essential to mission resilience. Recovery depends on integrity, that is, knowing precisely what was compromised and having trustworthy system components to initiate system restoration. Both requirements depend on the accountability, traceability, and supply chain assurance that the security design principles in NIST SP 800-160, Volume 1enforce. Systems without these properties cannot reliably recover because they cannot reliably determine the scope of a compromise.

Adapt. Systems must evolve their defenses in response to observed threats. This goal is directly relevant to PQC: the quantum threat landscape will evolve, and cryptographic standards may need to evolve in response. Systems engineered for adaptability — modular, well-documented, with clean interfaces between cryptographic and non-cryptographic components — can absorb algorithm changes with manageable effort. Systems that are not agile will face the current migration challenge again, at greater cost and on a shorter timeline.

> *PQC transition is a necessary condition for mission resilience but it is not a sufficient one. Mission resilience requires the full application of systems security engineering principles, not algorithm migration alone.*

6. What This Means in Practice

Treating PQC transition as a systems engineering problem, rather than an algorithm substitution problem, has concrete implications for how organizations should plan and execute the effort.

Use PQC Transition as a Forcing Function

The PQC migration creates a once-in-a-generation opportunity to revisit system architectures that have accumulated decades of technical debt. Organizations that use PQC transition to apply the NIST SP 800-160 design principles to enforce least privilege, reduce attack surface, engineer defense in depth, and establish accountability mechanisms, will emerge from the effort with systems that are genuinely more trustworthy secure, not merely compliant.

Start with Cryptographic Inventory — Then Go Deeper

Inventory is the minimum necessary starting point, but it is not sufficient on its own. A complete inventory reveals where cryptography lives. Systems security engineering then asks: how does each cryptographic mechanism interact with the surrounding system? What assumptions does it make about component trustworthiness? What happens when it fails? Answering those questions requires systems engineering discipline, not just scanning tools.

Architect for Cryptographic Agility

Cryptographic agility (i.e., the ability to swap cryptographic algorithms with minimal architectural disruption), is a design property that must be engineered deliberately. It requires clean separation between cryptographic mechanisms and the applications that use them, well-defined interfaces, and

© 2026 RONROSSECURE, LLC | CC BY 4.0.

documented dependencies. Organizations that achieve cryptographic agility through this transition will be far better positioned to respond to future algorithm deprecations, which are inevitable.

Rebuild PKI as a Systems Engineering Project

The federal Public Key Infrastructure must be fundamentally re-architected around new algorithms. This is not a software update. It is a ground-up rebuild of how identity and trust work across government — a systems engineering project with cascading dependencies across every agency, partner network, and contractor ecosystem. It deserves dedicated program resources and the rigorous application of NIST SP 800-160 design principles from the start, not a series of patches applied to existing architecture.

Communicate in Mission Resilience Terms

Technical briefings on algorithm deprecation do not necessarily move budgets or policy. Systems security engineering provides the language to reframe the problem in terms of mission resilience: which critical functions are at risk, under which conditions, with what consequences, and over what timeline. That is the argument that reaches decision-makers and it is the argument that only a systems engineering perspective can make credibly.

7. Conclusion

PQC transition is necessary. The quantum threat is real, the harvest-now-decrypt-later attack is already underway, and organizations face an enormous remediation challenge that will take years and significant resources to complete. None of that is in question.

What is in question is whether completing PQC transition will be sufficient to protect mission-critical systems in the 21st century threat environment. The answer, grounded in the principles of NIST SP 800-160, is clearly no.

Security is an emergent property of well-engineered systems, not the result of algorithm compliance. Trustworthy secure systems require the intentional application of security design principles throughout the system lifecycle. Mission resilience — the capacity to anticipate, withstand, recover from, and adapt to adversarial conditions — requires both PQC and the engineering discipline that SP 800-160 describes.

> *The organizations that successfully navigate the quantum transition will not be the ones that finish algorithm migration fastest. They will be the ones that use migration as a forcing function to build the trustworthy, resilient systems that should have been built from the beginning.*

That is the systems engineering imperative. And it is long overdue.

© 2026 RONROSSECURE, LLC | CC BY 4.0.