P Q C

POST-QUANTUM CRYPTOGRAPHY

The Post-Quantum Cryptography Field Guide

A Practitioner's Handbook

Arnulfo “Noof” Hernandez

9 chapters · 20 sections · ~210 min

Contents

FRONT MATTER

Foreword

A few years ago, I came across a TED talk presented by Quantum Physicist, Dr. Shohini Ghose.[1] It had been many years since I last attended an academic course on anything physics much less quantum. During her talk she explained the concept of superposition and entanglement amongst other topics related to quantum mechanics. I had heard of these concepts in previous readings and academic studies, but I took notice considering the recent grumblings about a future world where quantum computers will break modern encryption.

Quantum physics is a weird spooky world. Particles and atoms behave differently in their sub-atomic world versus our classical macro world. Qubits, which are the unit of measure in quantum can exist in multiple states, and they can even share information with each other over vast distances.

It is these attributes that make Quantum so powerful. It has the power to transform medicine and find cures for diseases like Parkinson's. It will impact energy and how we use power harvesting technology like batteries. It could even unlock new information about how the universe works and how humans form consciousness.

But what does all this quantum mechanics stuff have to do with cryptography and our networks? What is Q-day and why should we care?

My recent conversations with customers and colleagues around post quantum cryptography have led to this book. We hope to shed some light on the weird world of quantum and more importantly we hope to guide you on your post-quantum cryptography journey.

If you are curious about Quantum and need to know how to prepare for Q-day then this book is for you.

The Clock We Can't See

Somewhere in the world, there's an adversary capturing your organization's encrypted traffic right now. This information can't be read today—that's not currently possible. But they're relying on the idea that within the next decade or so, a sufficiently powerful quantum computer will let them decrypt everything they've been patiently collecting. The security community calls this **Harvest Now, Decrypt Later (HNDL)**,[2] and it means the quantum threat isn't a future problem. It's a here today data exfiltration campaign with a delayed payoff.

The day a cryptographically relevant quantum computer (CRQC) becomes operational—also known as **"Q-Day"**—isn't something we'll see coming with a press release. It may happen in a university lab or maybe in a classified nation state bunker. It may be announced months or years after it becomes operational. The point is: we won't know the day it happens. All we can do now is prepare our networks for a future with Quantum.

The Mandates Are Already Here

If the HNDL argument feels too abstract, here's the concrete version: the compliance clock is already ticking. NIST published the first three finalized post-quantum cryptography (PQC) standards in August 2024.[3] The NSA's CNSA 2.0 requires all new National Security Systems acquisitions to support quantum-resistant algorithms by January 2027.[4] Federal agencies must submit PQC transition plans by April 2026.[5] The Quantum

Computing Cybersecurity Preparedness Act—that's federal law, not an executive order that can be rescinded—mandates ongoing cryptographic inventories and migration planning.[6]

This isn't a "maybe someday" situation. If you sell to, partner with, or operate within the federal ecosystem, post-quantum cryptography is now a procurement, compliance, and architectural requirement. And the private sector won't be far behind—the EU has already published its own PQC migration roadmap targeting critical infrastructure by 2030.[7]

Why We Wrote This

We've spent years working with public sector customers—DoD, federal civilian, intelligence community, and state and local agencies—helping them architect, deploy, and secure some of the most complex network environments in the world. What we've learned is that the hardest part of any major technology transition isn't the technology itself. It's the gap between knowing something and then knowing how to act on it.

That gap is enormous in post-quantum cryptography right now. There's no shortage of whitepapers explaining Shor's algorithm or listing NIST's new standards. What's missing is a practical, pragmatic guide that answers the questions we hear in every customer conversation:

- "Okay, but where do I start?"
- "How do I even find all the cryptography in my environment?"
- "What breaks when I start changing things?"
- "How do I explain this to my leadership without losing them in the math?"
- "What's required vs. what's recommended vs. what's actually possible right now?"

This field guide is our attempt to close that gap. We've pulled together the regulatory mandates, the technical details, the migration frameworks, and the operational reality into something you can hold in one hand and actually use.

> **A QUICK DISCLAIMER** Let's address the elephant in the room. The primary author's of this guide work for F5, Inc. F5 makes products that live in the middle of network traffic—load balancers, SSL/TLS terminators, application delivery controllers, API gateways—and some of those products are directly relevant to a PQC migration. We're not going to pretend otherwise. But we made a deliberate choice when writing this book: **every chapter presents vendor-neutral guidance first.** Where F5 capabilities are relevant to a specific migration challenge, we'll call them out—clearly labeled, never disguised as generic advice. If you work with a different vendor stack, this book should still be one of the most useful things on your desk. If you happen to use F5, you'll get some bonus context on where those tools fit. Our credibility depends on your trust, and we'd rather you find this guide genuinely useful than feel like you got handed a sales pitch.

What This Book Is (and Isn't)

This is a practitioner's field guide—concise, opinionated, and focused on getting you from "aware" to "acting." It's designed to be carried to meetings, marked up with sticky notes, and dog-eared at the chapters that matter

to your role.

This is not a cryptography textbook. We'll explain the algorithms and the math when it's necessary to understand why something matters, but we won't bury you in lattice theory. If you want a deep academic treatment, we'll point you to excellent resources. Our job is to help you build a plan, make decisions, and start moving.

This is not an academic course on Quantum Mechanics. However, we will cover the basics because a firm understanding of why we are here is critical. We will cover core topics like Superposition and Entanglement. These topics will give you a firm understanding of what makes quantum so powerful and how it applies to cryptography.

Everything in this guide is grounded in primary sources: NIST standards, NSA guidance, CISA publications, IETF RFCs, and executive-level mandates. Every factual claim is cited. Where the landscape is uncertain or actively changing—and there's a lot of that right now—we'll tell you so.

Notes

The following sources support specific claims made in the Foreword. Full bibliographic entries for all sources referenced throughout the book appear in the Bibliography at the end of this guide.

1. Ghose, Shohini. "Quantum Computing Explained in 10 Minutes." TED Talk. TED Conferences, 2018. Available at: https://www.ted.com/talks/shohini_ghose_a_beginner_s_guide_to_quantum_computing

2. NSA Cybersecurity Advisory. "Quantum Computing and Post-Quantum Cryptography." The "Harvest Now, Decrypt Later" threat model is widely referenced across NSA, CISA, and NIST guidance as a primary motivation for urgent PQC adoption. See also: CISA, "Post-Quantum Cryptography Initiative," https://www.cisa.gov/quantum

3. National Institute of Standards and Technology. FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard; FIPS 204: Module-Lattice-Based Digital Signature Standard; FIPS 205: Stateless Hash-Based Digital Signature Standard. Published August 13, 2024. https://csrc.nist.gov/news/2024/postquantum-cryptography-fips-approved

4. National Security Agency. "Commercial National Security Algorithm Suite 2.0 (CNSA 2.0) Algorithms." PP-22-1338, Ver. 1.0, September 2022; updated FAQ Ver. 2.1, December 2024. CNSA 2.0 requires that all new NSS acquisitions support CNSA 2.0 algorithms by January 1, 2027. https://media.defense.gov/2025/May/30/2003728741/-1/-1/0/CSA_CNSA_2.0_ALGORITHMS.PDF

5. The White House. National Security Memorandum on Promoting United States Leadership in Quantum Computing While Mitigating Risks to Vulnerable Cryptographic Systems (NSM-10), May 4, 2022. Office of Management and Budget Memorandum M-23-02, November 18, 2022, requires FCEB agencies to submit cryptographic system inventories and migration plans. The April 2026 milestone for transition plan submissions is derived from NSM-10 agency timelines.

6. Quantum Computing Cybersecurity Preparedness Act, Pub. L. No. 117-349, signed December 21, 2022. Requires OMB to issue guidance on PQC migration, mandates agency cryptographic inventories, and requires

annual progress reports to Congress. As federal statute, its requirements cannot be rescinded by executive order.

7. NIS Cooperation Group. “Coordinated Implementation Roadmap for the Transition to Post-Quantum Cryptography.” Published early 2025. Recommends EU member states initiate national PQC transition strategies by end of 2026, transition critical infrastructure by 2030, and complete migration by 2035. In January 2026, the European Commission published a proposed directive amending NIS2 to include explicit PQC requirements.

Let’s get started. The quantum clock doesn’t pause for planning meetings.

FRONT MATTER

How to Read This Book

We designed this guide so you don't have to read it cover to cover. Different roles need different chapters. Here's where to start depending on what you're trying to accomplish.

The Reader's Legend

■ **THE CISO / SECURITY LEADER** "I need to brief my board, justify budget, and understand our compliance exposure." **Start with:** Ch 1 (The Quantum Threat) → Ch 2 (What's Vulnerable) → Ch 4 (Regulatory Landscape) → Ch 6 (Migration Roadmap)

■ **THE SECURITY ARCHITECT** "I need to design our crypto-agile architecture and plan the migration phases." **Start with:** Ch 3 (New Algorithms) → Ch 5 (Crypto Discovery) → Ch 6 (Migration Roadmap) → Ch 7 (Hybrid Mode) → Ch 8 (Protocol Deep Dives)

■ **THE NETWORK / SECURITY ENGINEER** "I need to know what changes in TLS, IPsec, SSH, and PKI—and what breaks." **Start with:** Ch 3 (New Algorithms) → Ch 7 (Hybrid Mode) → Ch 8 (Protocol Deep Dives) → Ch 9 (Day-2 Operations)

■ **THE FEDERAL / DoD PROGRAM MANAGER** "I need to understand compliance timelines, procurement language, and ATO impact." **Start with:** Ch 4 (Regulatory Landscape) → Ch 5 (Crypto Discovery) → Ch 6 (Migration Roadmap) → Appendix (Compliance Checklist)

Chapter Quick Reference

Ch	Title	You'll Walk Away With…	Depth
1	**The Quantum Threat**	Board-ready explanation of why PQC matters now	Conceptual
2	**What's Vulnerable & What's Not**	Clear map of which algorithms and protocols are at risk	Moderate
3	**The New Algorithms**	Plain-language understanding of ML-KEM, ML-DSA, SLH-DSA	Moderate-Technical
4	**The Regulatory Landscape**	Consolidated mandate timeline with what's law vs. policy	Strategic
5	**Cryptographic Discovery**	Step-by-step methodology for building your crypto inventory	Hands-On
6	**Building Your Migration Roadmap**	Phased migration plan template you can adapt	Strategic-Technical
7	**Hybrid Mode**	How to run classical + PQC side by side during transition	Technical
8	**Protocol Deep Dives**	What changes in TLS, IPsec, SSH, and PKI specifically	Technical
9	**Day-2 Operations**	Monitoring, tuning, and sustaining a PQC environment	Operational

Ch	Title	You'll Walk Away With…	Depth
App	**Appendices**	Glossary, algorithm cheat sheet, compliance checklist, links	Reference

Conventions Used in This Guide

Throughout this book, you'll see a few recurring elements designed to help you navigate quickly:

> ⚠ **MANDATE ALERT** Highlights specific compliance requirements with dates and sources. If you're in a regulated environment, don't skip these.

> **PLAIN-LANGUAGE SIDEBAR** When we need to explain a complex technical concept, these sidebars give you the "tell it to me like I'm briefing the general" version alongside the technical detail.

> **F5 PERSPECTIVE** Clearly marked sections where we show how F5 capabilities map to a specific migration challenge. Vendor-neutral guidance always comes first. Skip these if your stack doesn't include F5—you won't miss any core content.

FRONT MATTER

About the Author

Arnulfo “Noof” Hernandez

CISSP | CCSP | F5 Certified Solution Expert (Security & Cloud)

Arnulfo “Noof” Hernandez is a Solutions Architect at F5, Inc. supporting public sector customers including the Department of War, Intelligence communities, and federal civilian agencies. Arnulfo’s current focus areas include AI security, Zero Trust Architectures, and post-quantum cryptography—particularly as they apply to classified and high-security environments. He holds a master’s degree in cyber Intelligence from the University of South Florida and maintains dual 400-level F5 certifications in Security and Cloud alongside his CISSP and CCSP credentials.

Connect: LinkedIn ↗

CHAPTER 1

The Quantum Threat: Why This Matters Now

Before we can understand why post-quantum cryptography matters, we need to understand the strange and beautiful science that makes it necessary. This chapter is a guided tour—from the birth of quantum mechanics over a century ago, through the core concepts that make quantum computing so powerful, to the specific moment where that power threatens every encrypted system on the planet.

We promise to keep the math and complexity to a minimum. But some of these ideas are genuinely weird and understanding why they're weird is essential to understanding what we're up against. So, hang in there it's about to get a little nerdy. Grab a cup of coffee and let's go.

A Brief History of Quantum Mechanics

The story of quantum mechanics begins, like many great scientific stories, with a problem nobody could solve.

In 1900, German physicist Max Planck was wrestling with a puzzle called the "blackbody radiation problem"—the question of why hot objects glow certain colors at certain temperatures. Classical physics predicted that a hot object should radiate infinite energy at short wavelengths, which clearly didn't match reality. In what he later described as "an act of desperation," Planck proposed a radical idea: energy isn't emitted in a continuous stream. Instead, it comes in discrete packets—tiny, indivisible bundles that he called quanta.[1]

Planck didn't fully appreciate what he'd done. It fell to a 26-year-old patent clerk named Albert Einstein to take the next leap. In 1905—the same year he published the theory of special relativity—Einstein proposed that light itself exists as particles (later called photons), not just waves. This was heresy. Maxwell's equations had convinced the physics world that light was a wave phenomenon. Einstein was saying it was both.[2]

In 1913, the Danish physicist Niels Bohr took quantum ideas and applied them to the atom. His model proposed that electrons orbit the nucleus in fixed, quantized energy levels—and that they can "jump" between these levels by absorbing or emitting photons. Bohr's model accurately predicted the spectral lines of hydrogen, lending real credibility to quantum ideas.[3]

Then, in 1924, French physicist Louis de Broglie proposed something even stranger: if light (a wave) can behave like a particle, then particles—like electrons—should also behave like waves.[4] This idea of wave-particle duality was confirmed experimentally and opened the door to the modern quantum revolution.

The floodgates opened in 1925–1927. Werner Heisenberg developed matrix mechanics, Erwin Schrödinger developed wave mechanics (and the famous equation bearing his name), and Max Born provided the probabilistic interpretation of the wave function—the idea that quantum mechanics doesn't tell us where a particle is, but rather the probability of finding it in any given place.[5]

In 1927, Heisenberg published his uncertainty principle: you cannot simultaneously know both the exact position and the exact momentum of a particle. This isn't a measurement limitation—it's a fundamental property of nature.[6]

These discoveries shook the foundations of physics. Einstein himself was deeply uncomfortable with the probabilistic nature of quantum mechanics, famously objecting: “God does not play dice with the universe.” But experiment after experiment confirmed that at the subatomic level, the universe does exactly that.[7]

> **PLAIN-LANGUAGE SIDEBAR** Think of quantum mechanics as the operating system of reality at the smallest scales. Just as your computer’s behavior is governed by code you never see, every atom in the universe follows quantum rules—rules that are fundamentally different from the physics we experience in everyday life. The key insight: at the quantum level, things aren’t certain. They’re probabilistic.

Superposition: Being Everything at Once

Of all the strange concepts in quantum mechanics, **superposition** is the one that most directly enables quantum computing—and most directly threatens our cryptographic systems.

In our classical, everyday world, things have definite states. A light switch is either on or off. A coin on a table is either heads or tails. But in the quantum world, a particle can exist in multiple states simultaneously until it is measured. An electron doesn’t have to spin “up” or “down”—it can be in a superposition of both at the same time.

This isn’t a metaphor. It’s not that we don’t know the state and it could be either one. The particle genuinely exists in both states at once, described by a mathematical wave function that encodes the probability of each possible outcome. Only when we measure the particle does this superposition “collapse” into a definite result. More on the “measure” in a moment.

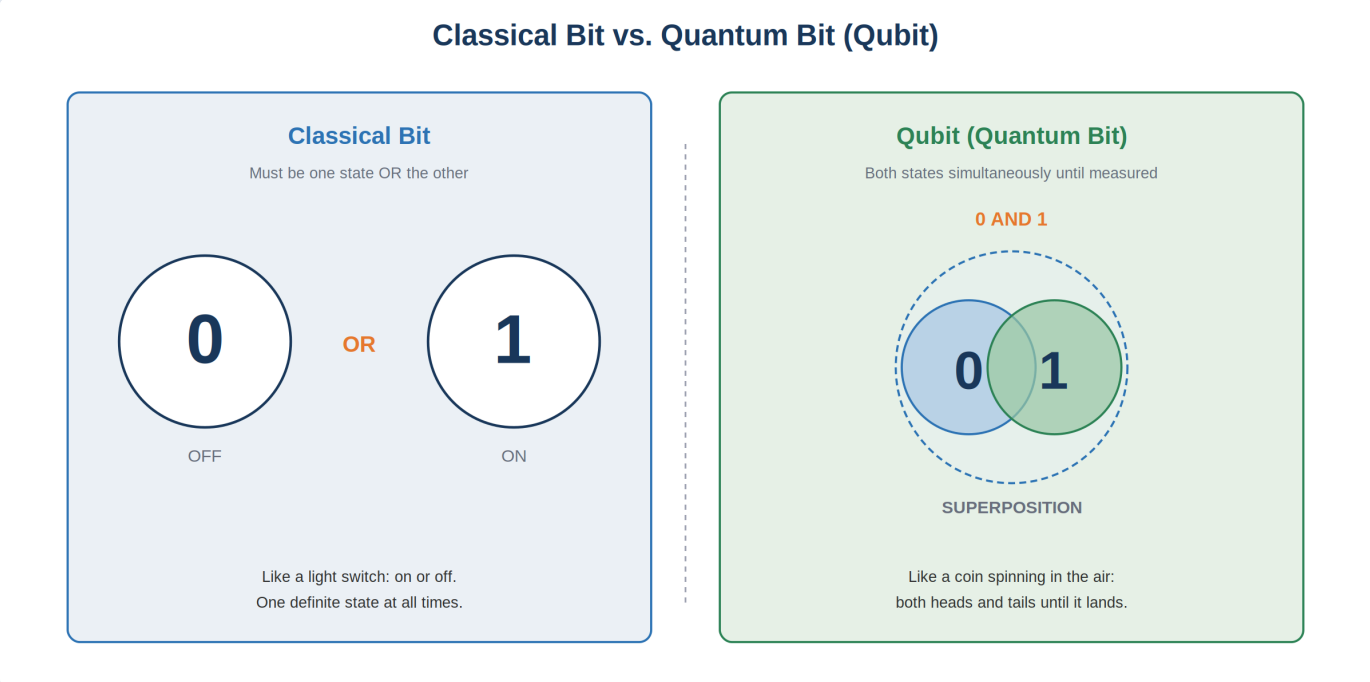

Figure 1.1 — Classical Bit vs. Quantum Bit (Qubit)

Schrödinger's Cat: A Thought Experiment

In 1935, Erwin Schrödinger devised a famous thought experiment to illustrate just how strange superposition is when scaled up.[8] Imagine a cat sealed in a box with a vial of poison, a radioactive atom, and a Geiger counter. If the atom decays (a quantum event), the Geiger counter triggers, breaks the vial, and the cat dies. If the atom doesn't decay, the cat lives.

According to quantum mechanics, until someone opens the box and observes the result, the radioactive atom is in a superposition of "decayed" and "not decayed." And since the cat's fate is tied to that atom, the cat is—in a quantum mechanical sense—both alive and dead simultaneously.

Schrödinger intended this as a reductio ad absurdum—a way to show how absurd quantum mechanics becomes at macroscopic scales. But the math works. The cat's superposition is real within the formalism, even if we never observe it in practice because large objects interact with their environment and "decohere" almost instantly. Think about this for a moment. We have all heard the phrase "if a tree falls in a forest and no one is there to see it, did it really fall?" In our classical world, our logical thinking says yes, the tree fell, just because no one was there to "measure or observe" it falling doesn't mean it didn't happen. Well in the quantum world this is not the case. The tree exists in both states until measured. Seems ridiculous I know, but this is how things work in the quantum world. More on measure or observer effect in a bit.

For quantum computing, superposition is not a paradox—it's a feature. A quantum bit (qubit) in superposition can represent 0 and 1 at the same time, combine this with entanglement and you suddenly reach exponential compute. This is what gives quantum computers their extraordinary potential power.

Entanglement: Spooky Action at a Distance

If superposition is strange, **entanglement** is downright out of science fiction.

In 1935, Einstein, Boris Podolsky, and Nathan Rosen published a paper (known as the EPR paradox) arguing that quantum mechanics must be incomplete.[9] Their thought experiment showed that two particles could be prepared in a way that measuring one would instantly determine the state of the other—no matter how far apart they were. Einstein called this "spukhafte Fernwirkung": spooky action at a distance.

Here's how it works: two particles interact and become "entangled." Their quantum states are now correlated. Separate them by inches or by light-years—it doesn't matter. Measure the spin of particle A and you will instantly know the spin of particle B, because their states are linked. Distance has no bearing on the relationship…spooky…indeed.

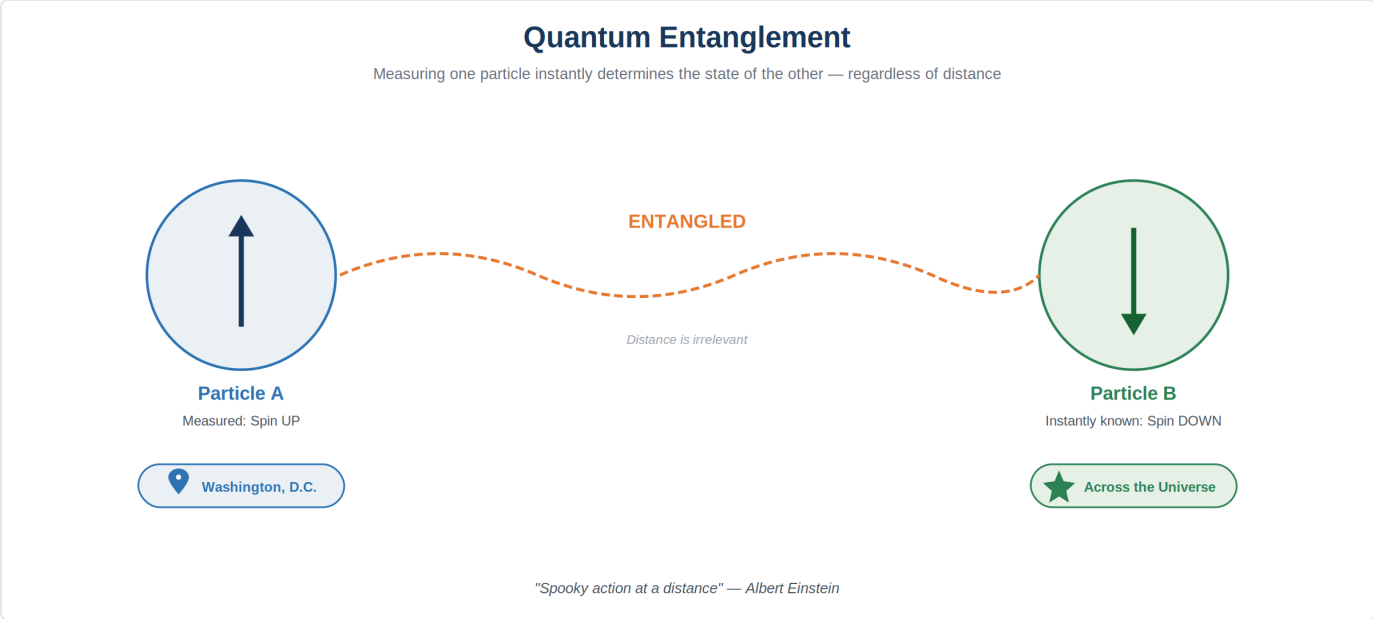

Figure 1.2 — Quantum Entanglement

This doesn't violate Einstein's speed-of-light limit (you can't use it to send information faster than light), but it reveals that entangled particles share information in a way that has no classical analogue. When Irish physicist John Bell proposed a test in 1964—and experiments by Alain Aspect and others confirmed it in the 1980s—the physics community had to accept that entanglement is real.[10]

For quantum computing, entanglement is the secret sauce. It allows qubits to be correlated in ways that classical bits cannot, enabling quantum algorithms to explore vast solution spaces simultaneously. Combined with superposition, entanglement is what makes a quantum computer fundamentally more powerful than a classical one for certain problems.

The Observer Effect: Measurement Changes Reality

One of the most counterintuitive aspects of quantum mechanics is the role of observation itself.

In the classical world, you can measure something without changing it. You can look at a thermometer or read a speedometer and the thing being measured doesn't care. But in the quantum world, the act of measurement **fundamentally alters the system being observed**.

When a particle is in superposition and you measure it, the wave function "collapses"—the particle snaps into one definite state. Before measurement: all possibilities exist. After measurement: only one reality remains. This is sometimes called the observer effect or the "measurement problem," and it sits at the heart of every interpretation of quantum mechanics.

The most dramatic demonstration is the now famous **double-slit experiment**. Fire electrons one at a time at a barrier with two slits and a detector screen behind it. Without observation, the electrons produce an interference pattern—as if each electron passed through both slits simultaneously as a wave. But place a detector at the slits to observe which path the electron takes, and the interference pattern vanishes. The electron behaves like a particle going through one slit or the other.[11] It's as if the particles "know" they are being watched!

The mere act of looking changes the outcome. For our purposes, this has a practical implication: quantum states are fragile. This fragility is both a challenge for building quantum computers (qubits are notoriously difficult to keep stable) and a feature of quantum cryptography (any attempt to eavesdrop on a quantum communication channel disturbs the signal and is detectable).

Quantum in the Wild: Nature Got There First

You might think quantum effects are limited to laboratory conditions near absolute zero. Nature disagrees.

One of the most fascinating discoveries of the past two decades is that **photosynthesis**—the process by which plants convert sunlight into chemical energy—may exploit quantum mechanics to achieve its remarkable efficiency.[12]

Here's the puzzle: when a photon of sunlight hits a chlorophyll molecule in a plant leaf, it dislodges an electron, creating an "exciton" (a paired electron-hole that acts like a tiny battery). This exciton must travel through a maze of molecular structures to reach the reaction center where photosynthesis actually happens. Classical physics would predict a random walk—the exciton bouncing from molecule to molecule like a pinball until it stumbles upon the right destination. That random process would be slow and inefficient, most of the energy would be lossed.

But photosynthesis operates at near-perfect quantum efficiency—virtually every photon captured is converted to usable energy. In 2007, researchers at UC Berkeley led by Graham Fleming observed something remarkable in the Fenna-Matthews-Olson (FMO) photosynthetic complex of green sulfur bacteria: the exciton wasn't hopping randomly. It was moving as a quantum wave, exploring multiple pathways simultaneously through quantum coherence—then collapsing along the most efficient route.[13]

> **PLAIN-LANGUAGE SIDEBAR** Think of it this way: a classical exciton is like a lost tourist wandering city streets, trying random turns until they find their hotel. A quantum exciton is like that same tourist existing on every street at once, then instantly appearing at the hotel via the shortest path. Nature figured out quantum computing long before we did.

The scientific community continues to debate the exact role of quantum effects in photosynthesis—some recent studies suggest that molecular vibrations, rather than purely electronic quantum coherence, may explain the observed efficiency.[14] But the broader point stands: quantum phenomena operate in warm, wet, noisy biological systems, not just in pristine laboratory conditions. Researchers have also found evidence of quantum effects in bird navigation (the "quantum compass" used by European robins), enzyme catalysis, and even the human sense of smell.[15]

Why does this matter for a book about cryptography? Because it demonstrates that quantum mechanics isn't abstract theory trapped in a physics journal. It's an operational reality—one that engineers are learning to harness for computing, and one that will inevitably transform the security landscape.

From Physics to Computing: The Qubit

A classical computer stores information in **bits**—binary digits that are either 0 or 1. Every computation, from loading a webpage to encrypting a file, is ultimately a sequence of operations on billions of these binary values.

A quantum computer replaces bits with **qubits** (quantum bits). Thanks to superposition, a single qubit can represent 0, 1, or both simultaneously. And when you entangle multiple qubits, their combined state space grows exponentially. Two entangled qubits can represent 4 states simultaneously (2^2). Three qubits: 8 states (2^3). Ten qubits: 1,024 states. Three hundred qubits can represent more states than there are atoms in the observable universe (2^{300}).[16] That's right...more atoms than the observable universe contains.

This isn't just more computing power—it's a fundamentally different kind of computing power. A quantum computer doesn't just try answers faster. It can explore an enormous number of possibilities at once, using interference to amplify correct answers and cancel out wrong ones. For certain types of problems, this approach is exponentially more efficient than anything a classical computer can do.

The key phrase is "certain types of problems." Quantum computers won't replace your laptop. They're not faster at Excel or email. But for specific mathematical problems—including the ones that underpin modern encryption—they are supremely effective.

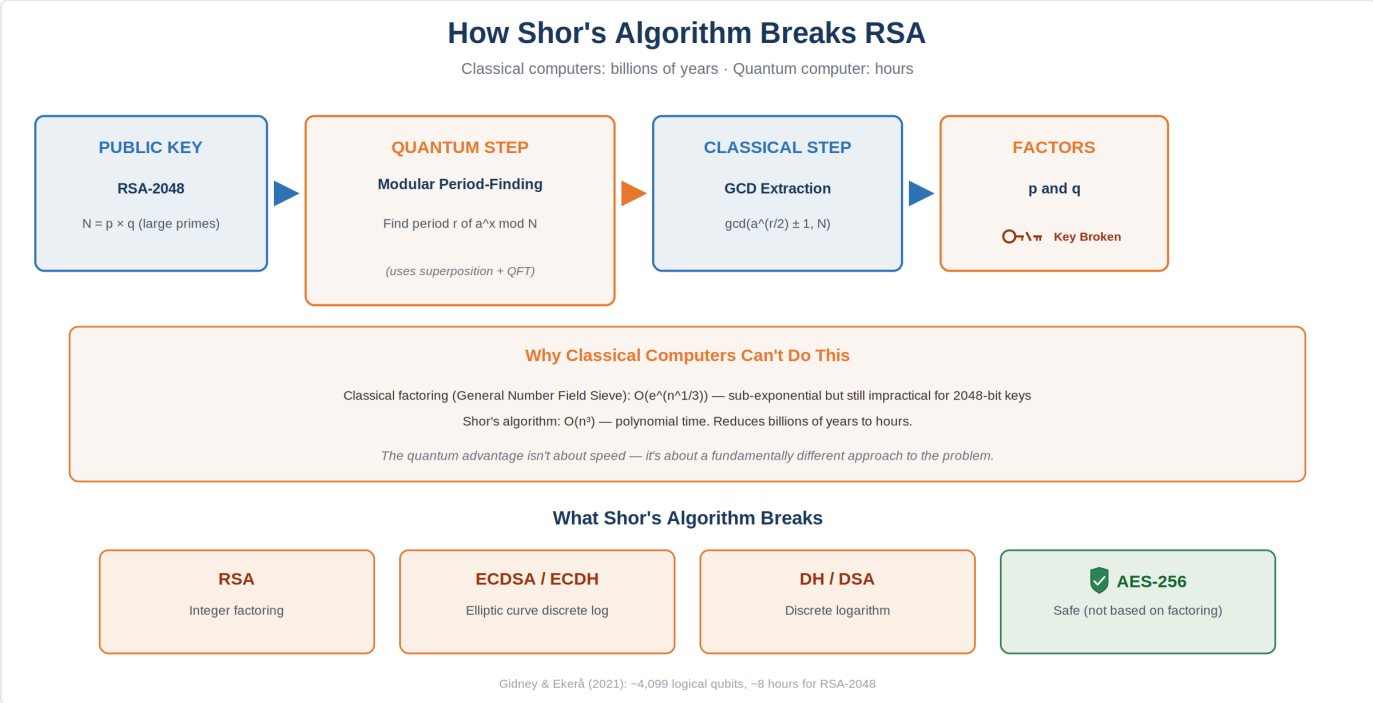

Figure 1.3 — How Shor's Algorithm Breaks RSA

The Qubit Reality Check: Physical vs. Logical

Before we go further, we need to address an important distinction that trips up even experienced technologists: not all qubits are created equal.

The qubits we've been describing—the ones that can exist in superposition and become entangled—are **physical qubits**. These are the actual hardware: superconducting circuits cooled to near absolute zero, trapped ions

held in electromagnetic fields, photons routed through optical networks, or any of the other physical systems being explored today. The problem with physical qubits is that they are extraordinarily fragile. They are susceptible to environmental noise, thermal fluctuations, and electromagnetic interference. A stray vibration or a slight temperature change can cause a qubit to lose its quantum state—a phenomenon called decoherence. And when a qubit decoheres, it introduces errors into the computation.

This is where **logical qubits** come in. A logical qubit is an error-corrected qubit—a single, reliable computational unit built from many physical qubits working together. The additional physical qubits act as a safety net, continuously detecting and correcting errors in real time through a process called **quantum error correction (QEC)**. Think of it like RAID in storage: you use redundancy to protect against failure.[16]

The overhead is staggering. Depending on the error rate of the underlying hardware and the error correction scheme used (the most common today is the surface code), you might need **1,000 to 10,000 physical qubits to create a single reliable logical qubit.** This is why quantum computing resource estimates are often expressed in physical qubit counts—they represent the total hardware required, including all the error correction overhead.

> **PLAIN-LANGUAGE SIDEBAR** When a vendor announces a "1,000-qubit processor," they're talking about physical qubits—the raw hardware. After error correction, that 1,000-qubit chip might yield just one or two usable logical qubits. It's the logical qubits that perform the meaningful computation. Shor's algorithm needs roughly 4,000–6,000 logical qubits to factor RSA-2048. When researchers say you need "20 million physical qubits" for that job, the vast majority of those qubits are dedicated to error correction—keeping those few thousand logical qubits stable long enough to finish the calculation.

This is both good news and bad news. The good news: we're a long way from having enough stable logical qubits to threaten real-world encryption. The bad news: error correction techniques are improving rapidly, physical qubit quality is increasing, and alternative error correction codes (such as quantum LDPC codes) promise to dramatically reduce the overhead ratio.[16] The distance between where we are and where an adversary needs to be is shrinking from both directions—better hardware and better algorithms.

How Modern Encryption Actually Works

Before we can understand what quantum computing breaks, we need to understand what it's breaking. Let's talk about the mathematical trick that secures nearly everything on the internet.

The Trapdoor: Easy One Way, Impossible the Other

Modern public-key cryptography is built on a concept mathematicians call a **trapdoor function**—a mathematical operation that is easy to perform in one direction but practically impossible to reverse without a secret key.[17]

Here's the simplest example. Take two large prime numbers—let's call them p and q—and multiply them together to get a product N. A fourth grader can multiply two numbers together, and a computer can do it in microseconds, even when those primes are hundreds of digits long. But now try going backward: given only N (the

product), figure out which two primes were multiplied to create it. That's the trapdoor. Multiplication is easy. Factoring is extraordinarily hard.

> **PLAIN-LANGUAGE SIDEBAR** Think of it like mixing paint. If someone hands you a can of yellow paint and a can of blue paint, you can mix them into green in seconds. But if someone hands you a can of green paint and says "tell me the exact shades of yellow and blue that were mixed to make this," you're in trouble. That's the trapdoor: easy to combine, practically impossible to decompose.

RSA, the most widely deployed public-key algorithm in history, depends on exactly this asymmetry.[18] When you generate an RSA key pair, you pick two huge random prime numbers (each typically 1,024 bits long—roughly 300 digits), multiply them together, and publish the result as your **public key**. Anyone can use that public key to encrypt a message to you. But only someone who knows the original two primes—your **private key**—can decrypt it.

Elliptic Curve Cryptography (ECC) and Diffie-Hellman key exchange work on different mathematical structures (elliptic curves and discrete logarithms, respectively), but they share the same fundamental design principle: there's a mathematical operation that's trivially easy in one direction and computationally infeasible to reverse.

Why Classical Computers Can't Break the Lock

To factor a 2048-bit RSA key, a classical computer would need to find the two prime factors of a number that is 617 digits long. The best known classical factoring algorithm (the General Number Field Sieve) would require approximately 2^{112} operations for this key size.[19] To put that in perspective: if you harnessed every classical computer on Earth and ran them until the sun burned out, you still wouldn't come close.

This isn't a guess or an approximation—it's a mathematical wall. Classical computers can only test potential factors one at a time (or a few at a time with parallelism), and the number of possibilities grows exponentially with key size. The entire security model of the modern internet rests on this simple bet: **no classical computer will ever be powerful enough to reverse the trapdoor.**

For over 40 years, that bet has held. RSA was published in 1977. The largest RSA key ever factored by a classical computer is RSA-250 (829 bits), and it took a distributed computing effort spanning years. RSA-2048 is in an entirely different league.

The Algorithms That Break Our Locks

Shor's Algorithm: The Lock Pick That Changes Everything

In 1994, mathematician Peter Shor published a paper that fundamentally changed the security landscape—even though the hardware to execute his idea didn't exist yet.[20] He proved that a quantum computer, using the unique properties of superposition and entanglement, could factor large numbers exponentially faster than any known classical algorithm. The trapdoor—the one-way function that the entire internet depends on—suddenly had a back door.

Here's how it works, in plain language:

Remember that a classical computer trying to factor a large number is essentially guessing and checking—trying one potential factor after another, sequentially. A 2048-bit RSA key has a solution space so vast that a classical machine could search it for billions of years without success.

Shor's algorithm takes a completely different approach. Rather than trying to find the factors directly, it converts the factoring problem into a **period-finding problem**—a question about repeating patterns in a mathematical function. This is the key insight: finding the period (the repeating cycle) of a specific modular arithmetic function reveals information that can be used to calculate the prime factors.[21]

Why does this matter? Because **quantum computers are exceptionally good at finding periods.** This is where superposition becomes a weapon. A quantum computer can prepare a massive superposition of all possible inputs simultaneously—not testing them one by one, but evaluating them all at once. Through a process called the Quantum Fourier Transform, the computer then uses interference to amplify the periodic patterns and suppress everything else. The correct period rises to the surface like a signal emerging from noise.

Once you have the period, extracting the prime factors is straightforward classical math—a few lines of algebra. The quantum computer doesn't do the factoring. It finds the hidden pattern that makes factoring trivially easy.

> **PLAIN-LANGUAGE SIDEBAR** Imagine you're trying to crack a combination lock with a billion possible combinations. A classical computer tries them one at a time: 000-000-001, 000-000-002, and so on. It'll be at this for a while. A quantum computer doesn't try combinations at all. Instead, it does something like shaking the lock in a very specific way and listening for resonance. The quantum properties of superposition let it test all billion combinations simultaneously, and interference makes the correct answer ring louder than the wrong ones. It's not brute force. It's a fundamentally different strategy that only works because of quantum mechanics.

Shor's algorithm doesn't just work on RSA. It also solves the **discrete logarithm problem** and the **elliptic curve discrete logarithm problem**—the mathematical foundations of Diffie-Hellman key exchange, DSA, ECDSA, and ECDH. Every major public-key algorithm deployed today relies on a trapdoor that Shor's algorithm can reverse. When a sufficiently powerful quantum computer runs Shor's algorithm, the entire public-key infrastructure collapses simultaneously.

How Close Are We? The Numbers Keep Dropping

Shor's algorithm itself needs only a few thousand logical qubits to factor RSA-2048—roughly 4,000 to 6,000, depending on the implementation.[22] But as we discussed earlier, each logical qubit requires thousands of physical qubits for error correction. That's why the total hardware estimates are so large.

In 2021, researchers Craig Gidney and Martin Ekerå estimated that breaking RSA-2048 would require approximately **20 million physical qubits** (supporting roughly 6,150 logical qubits with surface code error correction) operating for about 8 hours.[23] In 2025, Gidney published a further optimization—leveraging innovations in approximate arithmetic, more efficient qubit storage, and a technique called magic state cultivation—that reduced the estimate to **fewer than 1 million physical qubits** over roughly one week.[24] That's a 20x reduction in hardware requirements in just four years, using the same underlying hardware assumptions.

These numbers matter because they define the finish line for adversaries. While no quantum computer exists today with this capacity, the trajectory is clear: multiple hardware vendors have published roadmaps targeting

millions of physical qubits by the early 2030s. And as we've seen, progress comes from two fronts simultaneously—better physical hardware and smarter algorithms that reduce how many qubits you need in the first place.

> ⚠ **MANDATE ALERT** Shor's algorithm doesn't just threaten RSA. It breaks every public-key cryptosystem based on integer factorization, discrete logarithms, or elliptic curve discrete logarithms. This includes RSA, DSA, ECDSA, ECDH, and DH—the foundational algorithms of TLS, IPsec, SSH, S/MIME, code signing, and certificate-based authentication. When a CRQC arrives, all of these are compromised simultaneously.

The Solution Is Changing the Math

Here's the critical insight that sets up the rest of this book: **the problem isn't with encryption itself. The problem is with the specific mathematical problems we chose as our trapdoors.**

Factoring large numbers and computing discrete logarithms happen to be problems that quantum computers solve efficiently. But there are other mathematical problems—problems based on lattice geometry, error-correcting codes, hash functions, and other structures—that we have no reason to believe quantum computers can solve any faster than classical ones.

That's what post-quantum cryptography (PQC) is: replacing the math. We keep the concept of public-key encryption—the idea of trapdoor functions, key pairs, digital signatures—but we swap the underlying mathematical problem for one that resists both classical and quantum attack. The new NIST standards (ML-KEM, ML-DSA, SLH-DSA, which we'll cover in Chapter 3) are built on exactly these quantum-resistant mathematical foundations.

The protocols stay the same. TLS still does a handshake. IPsec still builds a tunnel. SSH still authenticates. But inside those protocols, the algorithms that generate keys, exchange secrets, and sign data are being swapped for new ones—ones where the trapdoor holds even against a quantum adversary. That migration—changing the math inside the protocols you already run—is the operational challenge this book is designed to help you navigate.

Grover's Algorithm: The Other Half of the Equation

Shor's algorithm targets public-key (asymmetric) cryptography—the trapdoor math that protects key exchange and digital signatures. But asymmetric crypto is only half the story. To understand the second quantum threat, we need to understand how asymmetric and **symmetric** encryption work together—because in practice, they're inseparable.

Why Symmetric Encryption Matters: The Workhorse You Don't See

Here's something that surprises many people: **public-key encryption doesn't actually protect your data.** Not directly. It's too slow for that. Encrypting a large file or a video stream with RSA would be orders of magnitude slower than using a symmetric algorithm like AES. Public-key crypto exists to solve one specific problem: how do two parties who've never met securely agree on a shared secret key?

Once that shared secret is established, **symmetric encryption** takes over and does the actual work. Symmetric algorithms like AES (Advanced Encryption Standard) use the same key for both encryption and de-

cryption. They're fast, efficient, and handle the bulk encryption of every email, file transfer, web session, VPN tunnel, and database connection on the planet.

The relationship between asymmetric and symmetric encryption is like a diplomatic courier and a secure phone line. The courier (asymmetric crypto) makes a dangerous trip across enemy territory to deliver a codebook. Once the codebook is delivered, the two parties use it to have fast, secure conversations over the phone (symmetric crypto). The courier's job is brief but critical; the phone line does the heavy lifting.

How This Plays Out in TLS: A Session in Two Acts

Let's walk through what actually happens when your browser connects to a website over HTTPS, because this is where both types of cryptography meet—and where both quantum threats apply.

Act 1: The Handshake (Asymmetric). When a TLS session begins, the client and server perform a handshake. In modern TLS 1.3, this uses an algorithm like **ECDHE** (Elliptic Curve Diffie-Hellman Ephemeral) to agree on a shared secret. The server also presents a digital certificate signed with an algorithm like **ECDSA** or **RSA** so the client can verify it's talking to the right server and not an impostor. This handshake phase is where Shor's algorithm strikes—it can break ECDHE, ECDSA, and RSA, allowing an attacker to either impersonate the server or recover the shared secret.

Act 2: Bulk Encryption (Symmetric). Once the handshake is complete and both sides have the shared secret, they derive symmetric session keys and switch to **AES-256-GCM** (or a similar symmetric cipher) for all subsequent data transfer. Every byte of actual content—HTML pages, API responses, file uploads, credentials—is encrypted with AES using the session keys from Act 1. This is the workhorse phase, and it's where Grover's algorithm applies.

> **PLAIN-LANGUAGE SIDEBAR** Think of a TLS session as a two-step process: Step 1 (the handshake): Public-key crypto securely delivers the key. This is the part Shor's breaks. Step 2 (the data transfer): Symmetric crypto uses that key to encrypt everything. This is the part Grover's weakens. If an attacker breaks Step 1, they get the key—and Step 2 is useless because the attacker can now decrypt everything with the stolen key. This is why Shor's is the existential threat. But even if Step 1 is quantum-safe, Step 2 needs to be strong enough to resist Grover's on its own. Both halves matter.

What Grover's Algorithm Actually Does

In 1996, Lov Grover published an algorithm for searching unsorted databases quadratically faster than any classical approach.[25] While this doesn't sound as dramatic as Shor's exponential speedup, the impact on symmetric cryptography is straightforward and significant.

Classically, if you want to brute-force an AES-256 key, you need to try up to 2^{256} possible keys—a number so large it would take every computer on Earth longer than the age of the universe. Grover's algorithm lets a quantum computer search that same key space in only the square root of the number of attempts. That means AES-256's effective security strength drops from 2^{256} to 2^{128} operations. AES-128 drops to 2^{64}—which begins to enter the range of a feasible attack.

Grover's also impacts **hash functions**—the algorithms (like SHA-256) used for digital fingerprinting, certificate validation, password storage, and blockchain integrity. A hash function's security against collision and

preimage attacks is similarly halved by Grover's. SHA-256 drops from 256-bit security to 128-bit equivalent.

The Practical Takeaway: Upgrade, Don't Panic

Unlike Shor's algorithm, which completely obliterates public-key crypto, Grover's merely weakens symmetric crypto—and the fix is simple: **double the key size.**

- **AES-256 → drops to 128-bit effective security.** Still computationally infeasible to break. AES-256 remains quantum-safe for any foreseeable future.
- **AES-128 → drops to 64-bit effective security.** Potentially vulnerable. Should be upgraded to AES-256.
- **SHA-256 → drops to 128-bit collision resistance.** Still practically secure for most applications.
- **3DES, Blowfish, and other legacy symmetric ciphers** with key sizes under 128 bits become unacceptable.

The message is reassuring but clear: Shor's algorithm is the existential crisis. Grover's algorithm is a maintenance upgrade. But that maintenance upgrade needs to happen—and for organizations still running AES-128 or SHA-1 in production environments (and you'd be surprised how many are), Grover's adds genuine urgency to a cleanup that should have happened years ago.

Q-Day: When Theory Becomes Threat

The security community uses the term **"Q-Day"** to describe the moment a cryptographically relevant quantum computer (CRQC) becomes operational—a machine powerful enough to run Shor's algorithm against real-world encryption keys.

When will Q-Day arrive? Honest answer: nobody knows for certain. Here's what we do know:

- As of early 2026, the largest quantum processors have roughly 1,000–1,200 physical qubits (e.g., IBM's Condor at 1,121 qubits, Google's Willow at 105 high-quality logical qubits). Breaking RSA-2048 requires roughly 1 million to 20 million physical qubits or roughly 4000-6000 logical qubits, depending on architecture.
- Multiple vendors—IBM, Google, Quantinuum, PsiQuantum, Microsoft—have published roadmaps targeting systems with hundreds of thousands to millions of qubits by the early 2030s.
- Algorithmic improvements continue to reduce the hardware threshold. Gidney's 2025 paper cut the required qubits by 20x compared to the 2021 estimate.[24]
- Error correction remains the critical bottleneck. Today's qubits are noisy; thousands of physical qubits are needed to create one reliable logical qubit.

Most expert assessments place Q-Day somewhere between 2030 and 2045, with significant uncertainty.[26] But the exact date is less important than a simple fact: **cryptographic migration takes 10–15 years for large organizations.** If Q-Day arrives in 2035 and you haven't started migrating, you're already too late.

⚠ **MANDATE ALERT** Industry Q-Day estimates have shortened materially since this book was first drafted. On March 25, 2026, Google's VP of Security Engineering announced an internal target to migrate all Google infrastructure to PQC by **2029**, citing faster-than-expected progress on quantum hardware, error correction, and factoring resource estimates. Cloudflare followed within days with a matching 2029 timeline. Scott Aaronson—historically the field's most prominent skeptic of overheated quantum claims—wrote in May 2026 that quantum-hardware and error-correction researchers he trusts now believe a fault-tolerant machine capable of breaking deployed cryptography "ought to be possible by around 2029." None of these are formal Q-Day announcements, but when the most cautious voices and the largest internet-scale infrastructure providers converge on a single year, the planning assumption shifts. Don't anchor your migration on 2035.[28]

Harvest Now, Decrypt Later: A Present-Tense Threat

There's a reason this chapter is called "The Quantum Threat" and not "The Quantum Future." The threat is already here.

The strategy is called **Harvest Now, Decrypt Later (HNDL)**, and it works like this: an adversary captures encrypted data today—intercepting TLS sessions, VPN tunnels, classified communications, financial transactions, healthcare records—and stores it. The adversary can't read the data now. But when a CRQC becomes available, they can retroactively decrypt everything they've collected.[27]

For data with a long sensitivity lifetime—classified intelligence, trade secrets, medical records, attorney-client communications, strategic military plans—HNDL means the clock started ticking the moment the data was transmitted. If your organization sent encrypted data over the wire in 2020 and a CRQC appears in 2035, that 2020 data is compromised.

HNDL transforms the quantum threat from a speculative future risk into a present-day data exfiltration campaign with a delayed decryption payoff. It's the reason NSA, CISA, and NIST have all emphasized that migration to post-quantum cryptography must begin now—not when Q-Day arrives, but years before it.

What's Next

Now that we understand the quantum mechanics, the computing power, and the specific threat algorithms, a natural question emerges: what exactly is at risk?

In Chapter 2, we'll map out precisely which cryptographic algorithms are vulnerable to quantum attack, which ones are safe, and how to classify your organization's data and systems by quantum risk level. That inventory is the foundation of everything that follows.

Notes

The following sources support specific claims made in Chapter 1. Full bibliographic entries appear in the Bibliography.

1. Planck, Max. "Über das Gesetz der Energieverteilung im Normalspektrum." Annalen der Physik 309 (1901): 553–563. Planck introduced the concept of energy quanta to resolve the blackbody radiation problem.

2. Einstein, Albert. “Über einen die Erzeugung und Verwandlung des Lichtes betreffenden heuristischen Gesichtspunkt.” Annalen der Physik 322, no. 6 (1905): 132–148. This paper proposed the photon hypothesis and is one of Einstein’s three revolutionary 1905 papers.

3. Bohr, Niels. “On the Constitution of Atoms and Molecules.” Philosophical Magazine 26 (1913): 1–25. Bohr’s atomic model introduced quantized electron orbits.

4. de Broglie, Louis. “Recherches sur la théorie des quanta.” PhD thesis, University of Paris, 1924. Proposed wave-particle duality for matter.

5. For the development of matrix mechanics: Heisenberg, W. “Über quantentheoretische Umdeutung kinematischer und mechanischer Beziehungen.” Zeitschrift für Physik 33 (1925): 879–893. For wave mechanics: Schrödinger, E. “Quantisierung als Eigenwertproblem.” Annalen der Physik (1926). For the probabilistic interpretation: Born, M. “Zur Quantenmechanik der Stoßvorgänge.” Zeitschrift für Physik 37 (1926): 863–867.

6. Heisenberg, Werner. “Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik.” Zeitschrift für Physik 43 (1927): 172–198. Introduces the uncertainty principle.

7. Pais, Abraham. “Suttle is the Lord: The Science and the Life of Albert Einstein.” Oxford University Press, 1982. Documents Einstein’s philosophical objections to quantum mechanics.

8. Schrödinger, Erwin. “Die gegenwärtige Situation in der Quantenmechanik.” Naturwissenschaften 23 (1935): 807–812, 823–828, 844–849. Introduces the cat thought experiment.

9. Einstein, A., Podolsky, B., and Rosen, N. “Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?” Physical Review 47, no. 10 (1935): 777–780. The EPR paradox paper.

10. Bell, J.S. “On the Einstein Podolsky Rosen Paradox.” Physics 1, no. 3 (1964): 195–200. Aspect, A., Dalibard, J., and Roger, G. “Experimental Realization of Einstein-Podolsky-Rosen-Bohm Gedankenexperiment: A New Violation of Bell’s Inequalities.” Physical Review Letters 49 (1982): 1804–1807. Confirmed entanglement experimentally.

11. The double-slit experiment has been demonstrated many times since Thomas Young’s original 1801 light experiment. A landmark electron version: Tonomura, A., et al. “Demonstration of single-electron buildup of an interference pattern.” American Journal of Physics 57 (1989): 117–120.

12. Engel, G.S., et al. “Evidence for wavelike energy transfer through quantum coherence in photosynthetic systems.” Nature 446 (2007): 782–786. The landmark paper reporting quantum coherence in photosynthesis.

13. Panitchayangkoon, G., et al. “Long-lived quantum coherence in photosynthetic complexes at physiological temperature.” Proceedings of the National Academy of Sciences 107, no. 29 (2010): 12766–12770.

14. Cao, J., et al. “Quantum biology revisited.” Science Advances 6, no. 14 (2020). This review argues that while quantum effects are present in photosynthesis, long-lived interexciton coherences may not be functionally significant; molecular vibrations and environment-assisted transport may better explain efficiency.

15. Lambert, N., et al. “Quantum biology.” Nature Physics 9 (2013): 10–18. Reviews quantum effects in photosynthesis, avian magnetoreception, enzyme catalysis, and olfaction.

16. Nielsen, Michael A. and Chuang, Isaac L. “Quantum Computation and Quantum Information.” Cambridge University Press, 10th Anniversary Edition (2010). The standard reference for quantum computing fundamentals.

17. Trapdoor functions are formally defined in: Diffie, Whitfield and Hellman, Martin. “New Directions in Cryptography.” IEEE Transactions on Information Theory 22, no. 6 (1976): 644–654. This landmark paper introduced the concept of public-key cryptography and one-way trapdoor functions.

18. Rivest, R.L., Shamir, A., and Adleman, L. “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems.” Communications of the ACM 21, no. 2 (1978): 120–126. The RSA algorithm paper.

19. For RSA-2048 classical factoring difficulty, see: Lenstra, Arjen K. “Key Lengths.” The Handbook of Information Security (2004). The General Number Field Sieve (GNFS) is the best known classical factoring algorithm. NIST SP 800-57 Part 1 Rev. 5 estimates RSA-2048 at approximately 112-bit classical security strength.

20. Shor, Peter W. “Algorithms for Quantum Computation: Discrete Logarithms and Factoring.” Proceedings of the 35th Annual Symposium on Foundations of Computer Science (1994): 124–134.

21. Shor’s algorithm converts integer factoring into period-finding via modular exponentiation. For an accessible explanation: Mermin, N. David. “Quantum Computer Science: An Introduction.” Cambridge University Press (2007), Chapter 3. For the original formulation: Shor (1994), ibid.

22. For logical qubit estimates for Shor’s algorithm: Gidney and Ekerå (2021) use roughly 3n logical qubits for an n-bit RSA key, yielding approximately 6,150 logical qubits for RSA-2048. Beauregard’s space-optimized circuit requires 2n+3 logical qubits (≈4,099 for RSA-2048). Chevignard et al. (2024) reduced this to approximately 0.85n (≈1,730 logical qubits) at the cost of significantly more operations. Physical qubit overhead depends on error correction scheme and hardware error rates.

23. Gidney, Craig and Ekerå, Martin. “How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits.” Quantum 5 (2021): 433. Assumes superconducting qubits on a square grid with 0.1% gate error rate and surface code error correction.

24. Gidney, Craig. “Factoring integers with sublinear resources on a superconducting quantum processor.” arXiv:2505.15917 (May 2025). Reduces RSA-2048 factoring estimate to fewer than 1 million physical qubits in approximately one week, using approximate residue arithmetic, yoked surface codes, and magic state cultivation.

25. Grover, Lov K. “A fast quantum mechanical algorithm for database search.” Proceedings of the 28th Annual ACM Symposium on Theory of Computing (1996): 212–219.

26. Estimates vary widely. The Global Risk Institute publishes annual assessments; Mosca and Piani (2022) estimated a 50% chance of a CRQC by 2031–2033. The Chinese Academy of Sciences projects RSA-2048 vulnerability beyond 2045 under current models. A 2025 MITRE analysis suggests RSA-2048 could remain secure until 2055–2060 absent major breakthroughs.

27. The HNDL threat model is referenced across NSA, CISA, and NIST guidance. See: CISA, “Post-Quantum Cryptography Initiative,” https://www.cisa.gov/quantum. Also: NSA Cybersecurity Advisory, “Quantum Computing and Post-Quantum Cryptography” (2022).

28. Heather Adkins (VP, Security Engineering, Google), “Quantum Frontiers May Be Closer Than They Appear,” blog.google, March 25, 2026. Google announced an internal target to migrate all infrastructure to PQC by 2029. Cloudflare followed with a matching 2029 timeline within the same week. Underlying technical drivers cited include the Google Quantum AI ECDLP-256 resource estimate (approximately 20× fewer physical qubits than prior estimates) and continued reductions in Gidney’s RSA-2048 factoring estimates beyond his May 2025 sublinear-resources paper. Note: Android 17 (June 2026) integrates PQC digital signatures using ML-DSA at the OS hardware root of trust. See also: Scott Aaronson, “Will You Heed My Warnings?” Shtetl-Optimized, May 2026, citing senior quantum-hardware and error-correction researchers projecting a fault-tolerant CRQC capable of breaking deployed cryptography by approximately 2029.

Next: Chapter 2 — What’s Vulnerable and What’s Not

CHAPTER 2

What’s Vulnerable and What’s Not

Chapter 1 explained why quantum computing threatens our cryptographic infrastructure. This chapter answers the next logical question: what, specifically, is at risk?

Not everything breaks. That’s important to understand upfront. Quantum computing breaks certain categories of cryptographic algorithms while leaving others largely intact. Knowing which is which—and understanding how they’re layered inside the protocols you actually run—is the foundation of every migration decision you’ll make.

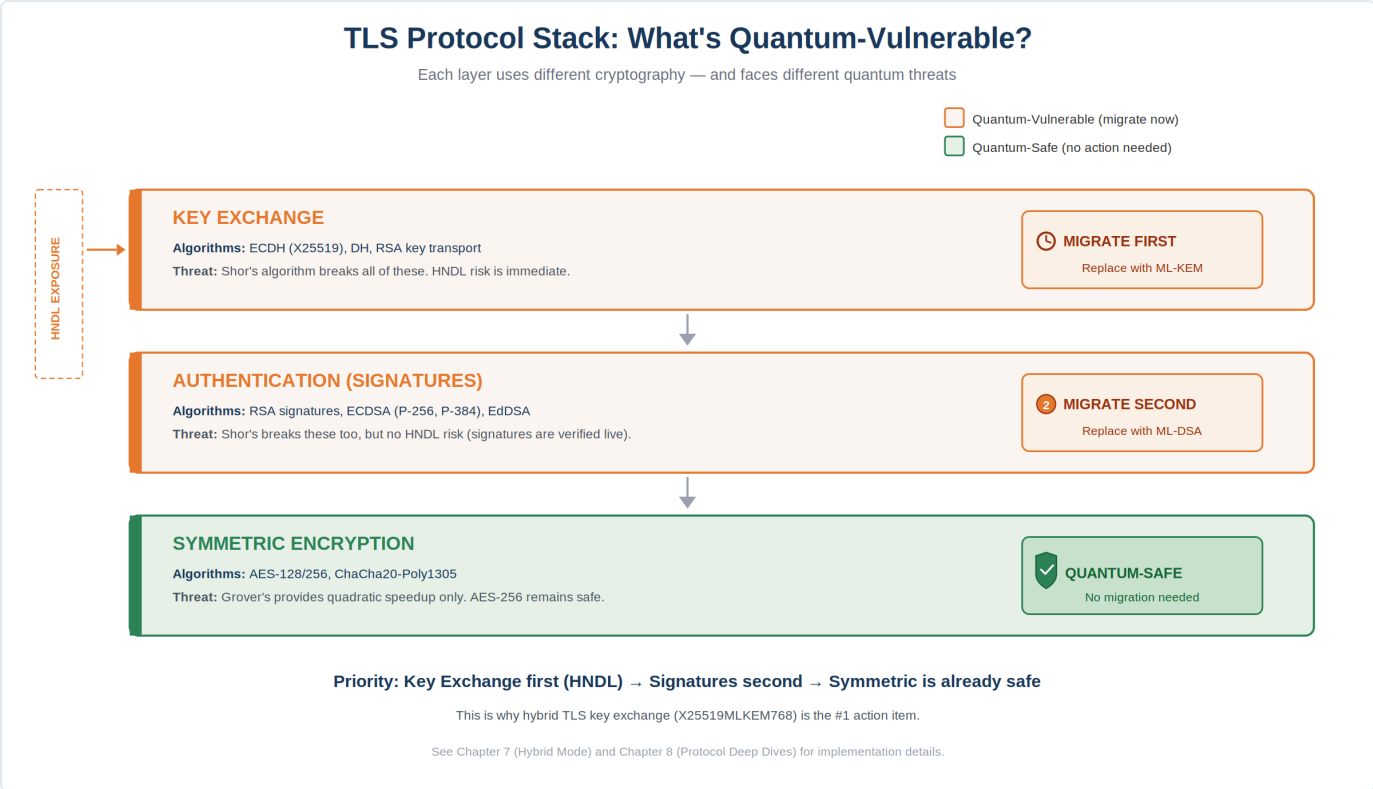

Figure 2.1 — TLS Protocol Stack: What’s Quantum-Vulnerable?

The Quantum Risk Scorecard

We can classify every cryptographic algorithm currently in widespread use into one of three categories based on its vulnerability to quantum attack:

❌ **BROKEN — Destroyed by Shor’s Algorithm** All public-key algorithms based on integer factorization, discrete logarithms, or elliptic curve discrete logarithms. A cryptographically relevant quantum computer (CRQC) renders these completely insecure—no reasonable increase in key size can help.

⚠ **WEAKENED — Reduced by Grover's Algorithm** All symmetric encryption and hash functions. Grover's algorithm halves their effective security strength. The fix is straightforward: double the key size. AES-256 remains safe; AES-128 needs upgrading.

✅ **SAFE — No Known Quantum Advantage** Post-quantum algorithms (ML-KEM, ML-DSA, SLH-DSA) and sufficiently strong symmetric algorithms (AES-256, SHA-384/512). These are built on mathematical problems for which no efficient quantum algorithm is known.

Algorithm-by-Algorithm: The Vulnerability Map

The following table is your reference sheet. Tear this page out and tape it to your monitor if you have to—this is the single most important classification in the book.

Algorithm	Type	Quantum Threat	Status	Action Required
RSA (all key sizes)	Key exchange, digital signatures	**Shor's — Broken**	Deprecated 2030, disallowed 2035	Replace with ML-KEM (key exchange) or ML-DSA (signatures)
ECDSA / EdDSA	Digital signatures	**Shor's — Broken**	Deprecated 2030, disallowed 2035	Replace with ML-DSA or SLH-DSA
ECDH / X25519 / X448	Key agreement	**Shor's — Broken**	Deprecated 2030, disallowed 2035	Replace with ML-KEM
DH / DHE	Key exchange	**Shor's — Broken**	Deprecated 2030, disallowed 2035	Replace with ML-KEM
DSA	Digital signatures	**Shor's — Broken**	Already deprecated by NIST	Replace with ML-DSA
AES-128	Symmetric encryption	**Grover's — Weakened**	Upgrade recommended	Upgrade to AES-256
SHA-1	Hash function	**Grover's + classical weaknesses**	Already broken classically	Replace immediately (SHA-256 minimum)
3DES / Blowfish	Symmetric encryption	**Grover's — Unacceptable**	Already deprecated	Replace immediately (AES-256)
AES-256	Symmetric encryption	**Grover's — Adequate**	128-bit effective — safe	No change needed
SHA-256 / SHA-384 / SHA-512	Hash functions	**Grover's — Adequate**	128-bit+ effective — safe	No change needed (avoid SHA-1)
HMAC-SHA-256+	Message authentication	**No known quantum advantage**	Safe	No change needed

Source: Classification based on NIST IR 8547 (Transition to Post-Quantum Cryptography) and NIST SP 800-131A Rev. 3.[1]

Protocol by Protocol: Where Quantum Hits Your Network

Algorithms don't exist in isolation—they're embedded inside protocols. The same RSA key might appear in a TLS certificate, an IPsec IKE negotiation, and an SSH login. Understanding which protocols are affected, and where inside each protocol the vulnerable algorithms sit, is essential for planning your migration.

TLS (Transport Layer Security)

TLS is the most widely deployed security protocol on the planet. It protects web traffic (HTTPS), API calls, email transmission (STARTTLS), and countless other connections. In Chapter 1, we walked through the TLS 1.3 handshake in two acts. Here's where quantum hits each act:

Handshake (key exchange): TLS 1.3 uses ECDHE (X25519 or P-256) for key agreement. **Broken by Shor's.** Must be replaced with ML-KEM or a hybrid (ML-KEM + X25519).[2]

Handshake (authentication): Server certificates typically use ECDSA or RSA signatures. **Broken by Shor's.** Must be replaced with ML-DSA or SLH-DSA signatures.[3]

Bulk encryption: AES-128-GCM or AES-256-GCM. AES-256 is **safe.** AES-128 should be upgraded to AES-256 as a precaution.

Record integrity: HMAC or AEAD (built into GCM). **Safe.** No change needed.

> **PLAIN-LANGUAGE SIDEBAR** In a TLS session, the handshake is the vulnerable part. Once the handshake completes and symmetric keys are established, the bulk data transfer is quantum-safe (assuming AES-256). This is why the PQC migration for TLS focuses almost entirely on the handshake—replacing ECDHE with ML-KEM and replacing ECDSA/RSA certificate signatures with ML-DSA.

IPsec / IKEv2

IPsec protects site-to-site VPNs, remote access VPNs, and classified network tunnels (including many DoD and federal environments). The IKEv2 protocol handles key exchange and authentication before the IPsec tunnel is established.

IKE key exchange: Uses DH or ECDH groups. **Broken by Shor's.** Must be replaced with ML-KEM. The IETF is developing PQC profiles for IKEv2, and the NSA's CNSA 2.0 specifies ML-KEM-1024 for IPsec.[4]

IKE authentication: Typically RSA or ECDSA certificate-based, or pre-shared keys (PSKs). Certificate-based auth is **broken by Shor's.** PSK-based authentication is **quantum-safe** (symmetric). This is why the NSA has recommended post-quantum pre-shared keys (PPKs) as an interim measure.[5]

ESP encryption: AES-256-GCM or AES-256-CBC. **Safe.**

SSH (Secure Shell)

SSH is the primary remote administration protocol for Linux/Unix systems, network devices, and cloud infrastructure. It's also widely used for secure file transfer (SFTP/SCP) and Git operations.

Key exchange: Typically ECDH (curve25519-sha256) or DH. **Broken by Shor's.** OpenSSH 9.0+ introduced a hybrid post-quantum key exchange (sntrup761x25519-sha512) using a lattice-based algorithm. OpenSSH 10.0 defaults to mlkem768x25519-sha256.[6]

Authentication: RSA, ECDSA, or Ed25519 keys for user/host auth. **Broken by Shor's.** Must be replaced with PQC signature algorithms. Note: password-based auth (derived via symmetric hashing) is not directly affected by Shor's but has its own well-known security limitations.

Session encryption: AES-256 (chacha20-poly1305 or aes256-gcm). **Safe.**

PKI and Digital Certificates

Public Key Infrastructure is the trust fabric of the internet. Every TLS certificate, code signing certificate, email certificate, and device identity certificate relies on public-key cryptography for its digital signature.

This is arguably the most complex PQC migration challenge. Certificates are everywhere—embedded in web servers, load balancers, API gateways, IoT devices, mobile apps, firmware, smart cards, and hardware security modules (HSMs). They're issued by Certificate Authorities (CAs) in hierarchical trust chains, and changing the signature algorithm means updating every link in that chain.[7]

Root and intermediate CA signatures: RSA or ECDSA. **Broken by Shor's.** Every certificate in the chain must eventually use PQC signatures.

End-entity certificates: RSA or ECDSA public keys and signatures. **Broken by Shor's.** Certificate sizes will increase significantly—ML-DSA-87 signatures are 4,627 bytes vs. 64 bytes for ECDSA P-256.[8]

> ⚠ **MANDATE ALERT** Certificate size explosion is a real operational concern. An ML-DSA-87 public key is 2,592 bytes; an ML-DSA-87 signature is 4,627 bytes. Compare that to ECDSA P-256: 64-byte public key, 64-byte signature. A TLS certificate chain with three PQC certificates could exceed 20 KB—potentially fragmenting the TLS handshake across multiple TCP packets and causing performance issues on constrained networks. Chapter 8 covers this in detail.

Code Signing and Software Supply Chain

Every signed software package, firmware update, OS patch, and container image relies on digital signatures to prove authenticity and integrity. These signatures almost universally use RSA or ECDSA.

Code signing signatures: RSA or ECDSA. **Broken by Shor's.** An attacker with a CRQC could forge signatures on malicious software, making it appear to come from a trusted vendor. This is why the NSA's CNSA 2.0 timeline prioritizes software and firmware signing first—with exclusive use of PQC signatures required by 2030.[9]

This has particular implications for long-lived systems: embedded devices, SCADA controllers, medical equipment, and military platforms that may run the same firmware for a decade or more. A signature that was secure when applied in 2024 may be forgeable by 2035. The integrity of every software update these systems have ever received becomes retroactively questionable.

Email Security (S/MIME, PGP)

Encrypted and signed email protocols rely on public-key cryptography for both confidentiality (encrypting the message to the recipient's public key) and authentication (signing the message with the sender's private key). Both operations are **broken by Shor's.**

Emails encrypted with RSA or ECC and captured today can be retroactively decrypted once a CRQC is available. For organizations handling attorney-client privileged communications, classified information, medical records, or trade secrets via encrypted email, the HNDL risk is acute and present.

Not All Data Is Equal: The HNDL Risk Matrix

Harvest Now, Decrypt Later doesn't affect all data equally. The risk depends on two factors: **how long the data remains sensitive** and **how likely it is to have been intercepted.**

We can use these two dimensions to classify your organization's data into quantum risk tiers:

Risk Tier	Data Sensitivity Lifetime	Examples	HNDL Urgency
CRITICAL	25+ years (classified, strategic)	National security intel, weapons designs, long-term trade secrets, diplomatic comms	**Immediate. Already at risk from HNDL. Migrate now.**
HIGH	10–25 years	Medical records, financial data, legal privilege, M&A strategy, PII under GDPR/HIPAA	Urgent. Begin migration planning now. Prioritize HNDL-exposed channels.
MODERATE	3–10 years	Business strategy, competitive analysis, customer databases, internal comms	Plan migration within NIST timeline. Prioritize based on exposure.
LOW	< 3 years	Session tokens, ephemeral API keys, transient web traffic, public marketing content	Migrate per normal upgrade cycles. Low HNDL exposure.

The critical insight: **your migration priority should be driven by data sensitivity lifetime, not by when you think Q-Day will arrive.** If your organization handles data in the CRITICAL or HIGH tiers and that data crosses a network—even an encrypted one—the HNDL clock is already running.

The Official Clock: NIST's Deprecation Timeline

In late 2024, NIST published IR 8547, "Transition to Post-Quantum Cryptography," which for the first time set explicit deprecation dates for quantum-vulnerable algorithms.[1] This document transformed PQC migration from a best-practice recommendation into a compliance requirement with hard deadlines:

- **By 2030:** RSA, ECDSA, EdDSA, DH, and ECDH will be **deprecated** at the 112-bit security level. Organizations should have migration plans in place and active.
- **By 2035:** All quantum-vulnerable public-key algorithms will be **disallowed**—completely removed from NIST standards. No exceptions.[10]

For context, "deprecated" means the algorithm is still technically permitted but actively discouraged—new systems should not use it. "Disallowed" means NIST-compliant systems cannot use it at all. If your organization's

compliance framework references NIST standards (and nearly all federal and most private-sector frameworks do), 2035 is the hard stop.

> ⚠ **MANDATE ALERT** Don't let the 2035 date create a false sense of comfort. The SHA-1 to SHA-2 migration—a far simpler cryptographic transition than PQC—took the industry over 12 years. The PQC transition involves replacing algorithms across every protocol layer, re-issuing every certificate, updating every HSM, and testing interoperability across every vendor in your stack. If you start in 2030, you are almost certainly too late for the 2035 deadline.

What's Not Vulnerable: A Reassuring List

It's easy to read the preceding sections and feel like everything is on fire (insert dog in computer room on fire meme). It's not. Here's what you can stop worrying about:

- **AES-256 is quantum-safe.** The symmetric workhorse of the internet isn't going anywhere. If you're already using AES-256, you're good.
- **SHA-256 and SHA-384/512 are quantum-safe for practical purposes.** Grover's weakens them, but the effective security levels remain computationally infeasible to attack.
- **HMAC constructions are safe.** Message authentication codes built on SHA-256+ are not meaningfully threatened.
- **Symmetric key derivation functions (HKDF, PBKDF2) are safe.** These are symmetric operations and inherit the Grover's-halving property—but with 256-bit inputs, the remaining 128-bit security is more than sufficient.
- **Random number generation is safe.** CSPRNGs (cryptographically secure pseudorandom number generators) are not affected by known quantum algorithms. The randomness foundation of your cryptographic stack remains solid.

The quantum threat is real but targeted. It's an asymmetric crypto problem first and foremost, with a manageable symmetric cleanup alongside it. The sky is falling on RSA, ECC, and DH. The sky is holding just fine over AES-256.

What's Next

Now that we know what is broken and what is safe, the next question is: what are we replacing the broken algorithms with? Chapter 3 takes you inside the new NIST post-quantum standards—ML-KEM, ML-DSA, SLH-DSA, and the upcoming FN-DSA and HQC—and explains how they work, what their tradeoffs are, and why NIST chose the mathematical foundations it did.

Notes

The following sources support specific claims made in Chapter 2. Full bibliographic entries appear in the Bibliography.

1. National Institute of Standards and Technology. NIST IR 8547 (Initial Public Draft), "Transition to Post-Quantum Cryptography." November 2024. Tables 1–5 classify quantum-vulnerable and quantum-resistant algorithms. NIST also published SP 800-131A Rev. 3 (November 2024) updating transition guidance with PQC inclusion and setting deprecation targets. Dustin Moody (NIST) confirmed the 2030 deprecation / 2035 disallowance timeline at the RWC PQC Conference, March 2025.

2. IETF draft-ietf-tls-mlkem-key-agreement specifies ML-KEM integration into TLS 1.3 key exchange. Hybrid approaches combining ML-KEM with X25519 are already deployed in Chrome (Google) and Cloudflare as of 2024.

3. TLS certificate signature migration is tracked in IETF drafts and the PKI Consortium's PQC working group. Hybrid certificates (containing both classical and PQC signatures) are under active development to ease the transition period.

4. NSA CNSA 2.0 specifies ML-KEM-1024 for key establishment in IPsec (National Security Systems). See: draft-guthrie-cnsa2-ipsec-profile for the CNSA 2.0 IPsec integration profile. Traditional networking equipment must support CNSA 2.0 by 2026 and use it exclusively by 2030.

5. Post-Quantum Pre-Shared Keys (PPKs) for IKEv2 are specified in RFC 8784. This allows organizations to add a quantum-resistant layer to existing IPsec tunnels without waiting for full PQC algorithm integration—effectively a stopgap measure.

6. OpenSSH 9.0 (April 2022) introduced sntrup761x25519-sha512 hybrid key exchange by default. OpenSSH 10.0 (April 2025) switched the default to mlkem768x25519-sha256, aligning with NIST's ML-KEM standard. See IETF draft-ietf-sshm-mlkem-hybrid-kex.

7. NIST SP 1800-38A (Migration to Post-Quantum Cryptography, Vol. A) identifies PKI migration as one of the most complex aspects of the PQC transition, noting that certificate chains, trust hierarchies, and cross-certification relationships all require coordinated updates.

8. ML-DSA-87 key and signature sizes from FIPS 204. Public key: 2,592 bytes. Signature: 4,627 bytes. Compare to ECDSA P-256: public key 64 bytes, signature 64 bytes. ML-KEM-1024 ciphertext is 1,568 bytes vs. 32 bytes for X25519. These size increases have meaningful performance implications for constrained networks and devices.

9. NSA CNSA 2.0 Algorithm Guidance (PP-22-1338, Ver. 1.0, September 2022). Software and firmware signing must begin transitioning immediately, support CNSA 2.0 by 2025, and exclusively use CNSA 2.0 by 2030. This is the earliest mandatory deadline in the CNSA 2.0 timeline.

10. NIST IR 8547 timeline: quantum-vulnerable algorithms at 112-bit security strength deprecated after 2030; all quantum-vulnerable algorithms at any security strength disallowed after 2035. This applies to RSA, DSA, ECDSA, EdDSA, DH, ECDH, and related schemes as listed in Tables 2 and 4 of the document.

Next: Chapter 3 — The New Algorithms: A Practitioner's Guide

CHAPTER 3

The New Algorithms: A Practitioner's Guide

In Chapter 1, we learned that quantum computing breaks the math behind today's encryption—not the concept of encryption itself. In Chapter 2, we mapped exactly which algorithms and protocols are vulnerable. Now we answer the next question: what are we replacing them with?

The good news is that the replacements are already here. NIST finalized the first three post-quantum standards in August 2024 and selected a fourth in March 2025, with a fifth in draft. These aren't experimental—they're the result of an eight-year international competition involving 82 submissions from 25 countries, whittled down through multiple rounds of analysis, attack, and optimization.[1]

This chapter introduces each algorithm in plain language: what mathematical problem it's built on, what it does, how big its keys and signatures are, and when to use it. We're not going to teach you the math—we're going to give you the intuition you need to make architecture and procurement decisions.

New Math, Same Mission

As we discussed in Chapter 1, today's public-key cryptography relies on trapdoor functions built from two mathematical problems: **integer factorization** (RSA) and **discrete logarithms** (DH, ECC). Shor's algorithm cracks both. The solution is to build trapdoor functions from different mathematical problems—problems that resist both classical and quantum attack.

NIST's post-quantum standards draw from three distinct mathematical families, each offering a different set of tradeoffs:

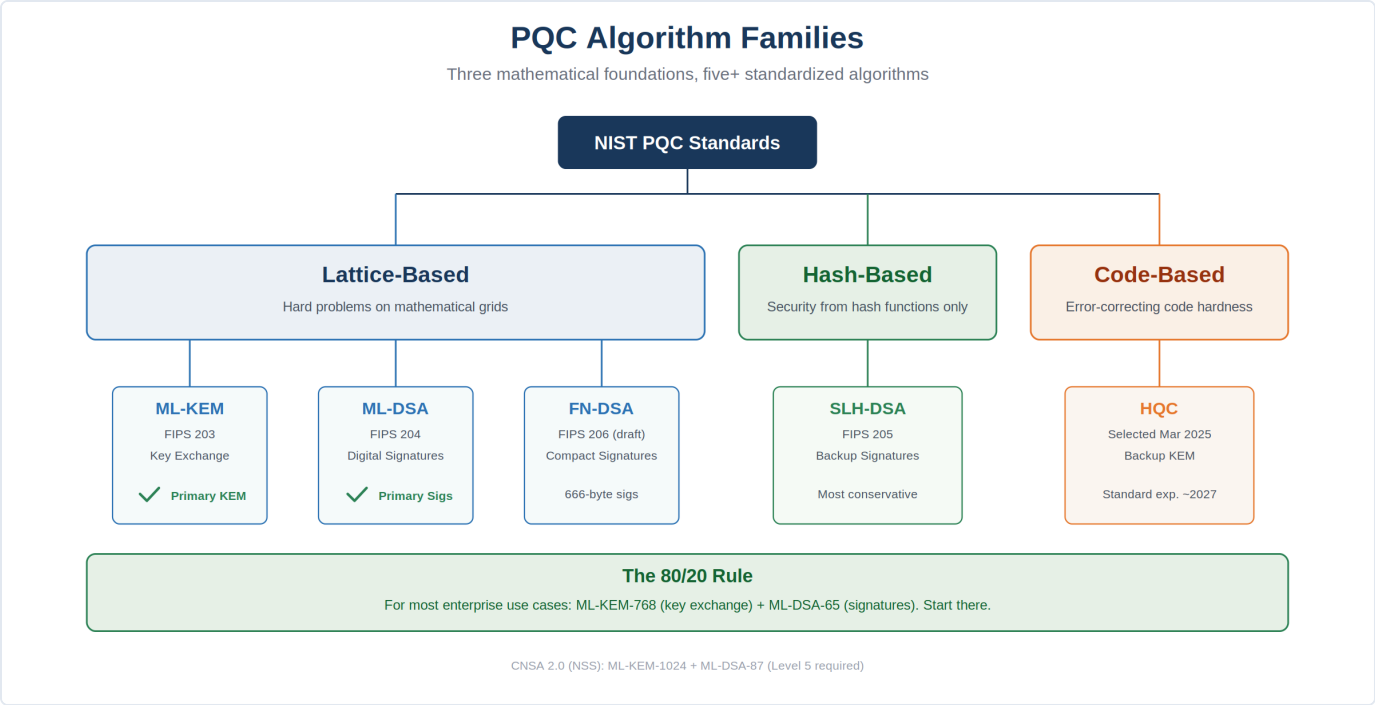

Figure 3.1 — PQC Algorithm Families

Math Family	Core Problem	Algorithms	Tradeoff Profile
Lattice-Based	Shortest Vector Problem (SVP) in high-dimensional lattices	ML-KEM (FIPS 203), ML-DSA (FIPS 204), FN-DSA (FIPS 206 draft)	Fast, moderate key sizes. The work-horse family. Most algorithms use this.
Hash-Based	Security of cryptographic hash functions (SHA-2/SHA-3)	SLH-DSA (FIPS 205), LMS/XMSS (SP 800-208)	Ultra-conservative security. Small keys, very large signatures. Slower.
Code-Based	Syndrome decoding in error-correcting codes	HQC (selected March 2025)	Backup KEM. Larger keys than lattices. Different math for diversity.

NIST deliberately chose algorithms from multiple mathematical families. If a future breakthrough cracks lattice problems, the hash-based and code-based backups still stand. This **algorithmic diversity** is a core design principle of the post-quantum landscape.[2]

> **PLAIN-LANGUAGE SIDEBAR** Think of it like building three separate bridges across a canyon, each using different engineering principles—one suspension, one arch, one cantilever. If an earthquake reveals a flaw in suspension bridge design, the arch and cantilever bridges still hold. NIST built multiple mathematical bridges for the same reason.

ML-KEM: The New Key Exchange (FIPS 203)

What it replaces: RSA key exchange, ECDH, X25519, Diffie-Hellman—any algorithm used to establish a shared secret between two parties.

What it is: ML-KEM stands for Module-Lattice-Based Key-Encapsulation Mechanism. It's based on the CRYSTALS-Kyber algorithm, which won NIST's competition and was renamed for standardization.[3]

How It Works (The Intuition)

ML-KEM's security is based on the **Module Learning With Errors (MLWE)** problem. Here's the intuition: imagine a system of equations where the answers are slightly wrong—each one has a small random error added to it. Given the noisy answers, recovering the original secret is computationally infeasible, even for a quantum computer.

> **PLAIN-LANGUAGE SIDEBAR** Think of it this way: someone gives you the results of 1,000 math equations, but each answer is slightly off by a random amount. Could you figure out the original variables? For low-dimensional problems, maybe. But in the high-dimensional spaces ML-KEM operates in (hundreds of dimensions), the noise makes the problem impossibly hard. That hardness is the trapdoor.

Technically, ML-KEM is a **Key Encapsulation Mechanism (KEM)**, not a traditional key exchange. The distinction matters: in a KEM, one party generates a shared secret and "encapsulates" it using the other party's public key. The recipient "decapsulates" it with their private key. The result is the same—a shared secret both

parties can use for symmetric encryption—but the mechanics are slightly different from Diffie-Hellman's interactive exchange.[4]

The Numbers

Parameter Set	Public Key	Ciphertext	Secret Key	NIST Security Level
ML-KEM-512	800 bytes	768 bytes	1,632 bytes	Level 1 (AES-128)
ML-KEM-768	1,184 bytes	1,088 bytes	2,400 bytes	Level 3 (AES-192)
ML-KEM-1024	1,568 bytes	1,568 bytes	3,168 bytes	Level 5 (AES-256)
X25519 (classical)	32 bytes	32 bytes	32 bytes	— (broken by Shor's)

The size increase is real but manageable. ML-KEM-768 (the recommended default for most applications) adds about 1.1 KB to each side of a handshake. For a TLS connection, that's barely noticeable. The performance story is actually encouraging: **ML-KEM is often faster than the elliptic curve algorithms it replaces** for key generation and encapsulation operations.[5]

> ⚠ **MANDATE ALERT** CNSA 2.0 requires ML-KEM-1024 for National Security Systems (not ML-KEM-768). Commercial and non-NSS federal systems may use ML-KEM-768. Plan your parameter set selection based on your compliance requirements.

ML-DSA: The New Digital Signature (FIPS 204)

What it replaces: RSA signatures, ECDSA, EdDSA—any algorithm used to prove identity and verify data integrity in certificates, code signing, and authentication.

What it is: ML-DSA stands for Module-Lattice-Based Digital Signature Algorithm, based on CRYSTALS-Dilithium. It's NIST's primary recommended signature scheme—the general-purpose workhorse.[6]

How It Works (The Intuition)

ML-DSA uses a technique called **Fiat-Shamir with Aborts.** The signer creates a mathematical commitment, generates a challenge by hashing the message and commitment together, then computes a response. The clever part is the "aborts" mechanism: before publishing the response, the algorithm checks whether it would leak any information about the private key. If it does, it throws the result away and tries again with fresh randomness. This rejection sampling ensures the final signature is mathematically independent of the secret key's internal structure.[7]

The Numbers

Parameter Set	Public Key	Signature	Secret Key	NIST Level
ML-DSA-44	1,312 bytes	2,420 bytes	2,560 bytes	Level 2 (AES-128)
ML-DSA-65	1,952 bytes	3,309 bytes	4,032 bytes	Level 3 (AES-192)
ML-DSA-87	2,592 bytes	4,627 bytes	4,896 bytes	Level 5 (AES-256)

Parameter Set	Public Key	Signature	Secret Key	NIST Level
ECDSA P-256 (classical)	64 bytes	64 bytes	32 bytes	— (broken)

This is where the sticker shock hits. An ML-DSA-65 signature is **3,309 bytes versus 64 bytes for ECDSA P-256**—a 50x increase. Public keys grow from 64 bytes to nearly 2 KB. For a TLS certificate chain with three certificates, the total signature and key payload could exceed 15–20 KB. We'll dig into the protocol-level implications of this in Chapter 8.

The silver lining: **ML-DSA is actually faster than RSA-2048 for both signing and verification**—roughly 10x faster for signing operations.[8] The performance penalty compared to ECDSA exists but is modest on modern hardware. The real challenge isn't CPU time—it's bandwidth and packet size.

SLH-DSA: The Conservative Backup (FIPS 205)

What it replaces: Same use cases as ML-DSA (digital signatures), but intended as a backup with a different security foundation.

What it is: SLH-DSA stands for Stateless Hash-Based Digital Signature Algorithm, based on SPHINCS+. Its security rests entirely on the strength of hash functions (SHA-2 or SHAKE)—mathematical objects that have been studied intensively for decades and are extremely well understood.[9]

Why It Matters

SLH-DSA is NIST's insurance policy. If a future mathematical breakthrough weakens lattice-based cryptography (threatening both ML-KEM and ML-DSA), SLH-DSA remains standing because it's built on completely different mathematics. Its security assumptions are the most conservative in the entire PQC portfolio.

The tradeoff is severe: SLH-DSA signatures are enormous. At the NIST Level 1 security ("small" variant), a signature is 7,856 bytes. At Level 5 with the "fast" variant, signatures can reach nearly 50 KB.[10] Public keys are tiny (32–64 bytes), but the signing process is significantly slower than ML-DSA.

SLH-DSA comes in two flavors for each security level: **"f" (fast)** optimizes for speed at the cost of larger signatures, and **"s" (small)** optimizes for signature size at the cost of speed.

Best use cases: Firmware signing, long-term document authentication, root CA certificates, and any context where the signature is created once and verified rarely but must remain trustworthy for decades. Not ideal for high-volume, latency-sensitive applications like TLS handshakes.

FN-DSA: Compact Signatures (FIPS 206 — Draft)

What it is: FN-DSA stands for FFT over NTRU-Lattice-Based Digital Signature Algorithm, based on the FALCON submission. It offers the smallest signatures of any PQC signature scheme—roughly 666 bytes at Level 1 and 1,280 bytes at Level 5—making it attractive for bandwidth-constrained environments.[11]

Why it isn't the primary standard: FN-DSA's implementation requires Gaussian sampling using floating-point arithmetic, which is notoriously difficult to implement correctly and prone to side-channel attacks. NIST

selected ML-DSA as the primary standard specifically because it is easier to implement securely. FN-DSA is the specialist tool, not the default.[12]

Best use cases: IoT devices, embedded systems, DNSSEC, and other environments where signature size is a critical constraint and the implementation team has deep cryptographic expertise.

HQC: The Backup KEM (Expected 2027)

What it is: HQC stands for Hamming Quasi-Cyclic, a Key Encapsulation Mechanism built on **error-correcting codes** rather than lattices. NIST selected HQC in March 2025 as a backup to ML-KEM, with a finalized standard expected in 2027.[13]

HQC serves the same purpose as ML-KEM (establishing shared secrets for symmetric encryption) but uses fundamentally different mathematics. If a cryptanalytic breakthrough ever compromised lattice-based schemes, HQC would provide an alternative path to quantum-safe key exchange.

The tradeoff: HQC's keys and ciphertexts are 3–4x larger than ML-KEM at equivalent security levels (roughly 4,500 bytes for a public key at Level 3, compared to 1,184 bytes for ML-KEM-768).[14] It also requires more computation. For now, ML-KEM remains the clear default for production deployments, with HQC as the strategic fallback.

> **PLAIN-LANGUAGE SIDEBAR** You don't need to deploy HQC today. Think of it as the spare tire in your trunk—you hope you never need it, but you're glad it's there. Organizations designing for crypto-agility (covered in Chapter 6) should ensure their architectures can accommodate HQC if ML-KEM ever needs to be swapped out.

Which Algorithm, When: The Practitioner's Decision Guide

With five algorithms across three mathematical families, choosing the right one for each use case can feel overwhelming. It doesn't need to be. Here's the decision tree:

Use Case	Primary Algorithm	Backup / Alternative	Notes
TLS key exchange	ML-KEM-768 (hybrid with X25519)	HQC (when standardized)	Hybrid mode recommended during transition
TLS / web certificates	ML-DSA-65	SLH-DSA (if conservative posture needed)	Watch for certificate size impacts
IPsec / VPN (NSS)	ML-KEM-1024 + ML-DSA-87	— (CNSA 2.0 mandates these)	Level 5 required for National Security Systems
Code signing / firmware	ML-DSA-65 or SLH-DSA	LMS/XMSS (stateful, if supported)	CNSA 2.0 prioritizes this use case first
SSH key exchange	ML-KEM-768 (hybrid with X25519)	— (OpenSSH 10.0 default)	Already deployed in latest OpenSSH
Email encryption (S/MIME)	ML-KEM-768	HQC (future)	Awaiting S/MIME protocol updates

Use Case	Primary Algorithm	Backup / Alternative	Notes
IoT / embedded devices	FN-DSA (when finalized) or ML-DSA-44	SLH-DSA-128s (if signature frequency low)	Evaluate FN-DSA for size-constrained use cases
Long-term archival signatures	SLH-DSA	— (most conservative choice)	Hash-based security — highest long-term confidence

The 80/20 Rule

If the table above feels complex, here's the simplification: **for 80% of use cases, you need exactly two algorithms:**

- **ML-KEM-768** for key exchange (replacing ECDH/X25519/DH)
- **ML-DSA-65** for digital signatures (replacing ECDSA/RSA signatures)

If you're in a CNSA 2.0 environment, bump both to Level 5 (ML-KEM-1024 and ML-DSA-87). If you have specialized needs (ultra-conservative security, bandwidth constraints, firmware signing), the other algorithms fill those niches. But ML-KEM + ML-DSA covers the vast majority of the migration surface area.

> **MANDATE ALERT** **A Note on Implementation Security: Side-Channel Attacks** Even a theoretically secure algorithm can be broken by a careless implementation. PQC algorithms are susceptible to **side-channel attacks**—techniques that extract secret key material by observing execution time, power consumption, or electromagnetic emissions rather than attacking the math directly. Researchers have demonstrated side-channel key recovery against ML-KEM implementations, and Falcon's (FN-DSA) floating-point arithmetic makes constant-time implementation especially difficult. The practical takeaway: always use vetted, validated implementations from established cryptographic libraries (OpenSSL, BoringSSL, liboqs, Windows CNG). Never roll your own PQC. Ensure your vendors' implementations are tested against side-channel attacks, especially if you're deploying on hardware where physical access is possible (IoT, OT, embedded systems). The AIVD/TNO PQC Migration Handbook specifically flags this as a risk that organizations frequently underestimate.

What's Next

You now understand what broke (Chapter 2), what replaces it (this chapter), and why these specific algorithms were chosen. But algorithms don't deploy themselves. The next question is: who says you have to do this, and by when?

Chapter 4 maps the full regulatory landscape—NSM-10, CNSA 2.0, the Quantum Computing Cybersecurity Preparedness Act, NIST IR 8547, and the EU and UK timelines—so you know exactly which mandates apply to your organization and when the deadlines hit.

Notes

The following sources support specific claims made in Chapter 3. Full bibliographic entries appear in the Bibliography.

1. NIST Post-Quantum Cryptography Standardization project. Initiated 2016 with 82 submissions from 25 countries. Final standards FIPS 203, 204, 205 published August 13, 2024. Fifth algorithm (HQC) selected March 2025. See: https://csrc.nist.gov/projects/post-quantum-cryptography

2. NIST explicitly sought algorithmic diversity. Dustin Moody (NIST PQC project lead): "We want to have a backup standard that is based on a different math approach than ML-KEM." NIST news release, March 11, 2025.

3. FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard. August 2024. Based on CRYSTALS-Kyber. Specifies ML-KEM-512, ML-KEM-768, and ML-KEM-1024 parameter sets.

4. NIST SP 800-227 (draft): Recommendations for Key-Encapsulation Mechanisms. Provides formal definitions and guidance for implementing KEMs, distinguishing them from traditional key agreement protocols like DH and ECDH.

5. NIST SP 1800-38C (Preliminary Draft): Quantum Readiness—Testing Draft Standards for Interoperability and Performance. Volume C reports ML-KEM-768 handshake throughput competitive with or exceeding classical ECDH at higher security levels.

6. FIPS 204: Module-Lattice-Based Digital Signature Standard. August 2024. Based on CRYSTALS-Dilithium. Specifies ML-DSA-44, ML-DSA-65, and ML-DSA-87 parameter sets.

7. The Fiat-Shamir with Aborts paradigm for ML-DSA is described in FIPS 204, Section 3. For an accessible explanation: Lyubashevsky, V. "Fiat-Shamir With Aborts: Applications to Lattice and Factoring-Based Signatures." ASIACRYPT 2009.

8. ML-DSA signing performance approximately 100–200 microseconds vs. RSA-2048 at 2–5 milliseconds. See benchmarks in NIST SP 1800-38C and Open Quantum Safe project: https://openquantumsafe.org

9. FIPS 205: Stateless Hash-Based Digital Signature Standard. August 2024. Based on SPHINCS+. Security relies solely on the collision resistance and preimage resistance of SHA-2 or SHA-3 (SHAKE) hash functions.

10. SLH-DSA signature sizes from FIPS 205: SLH-DSA-SHA2-128s = 7,856 bytes; SLH-DSA-SHA2-256f = 49,856 bytes. Public key sizes range from 32 to 64 bytes across all parameter sets.

11. FN-DSA (FALCON) is specified in draft FIPS 206. FN-DSA-512 signature: approximately 666 bytes. FN-DSA-1024 signature: approximately 1,280 bytes. Compact compared to ML-DSA but implementation complexity is significantly higher.

12. NIST noted that FALCON's Gaussian sampling over floating-point arithmetic makes constant-time implementation difficult, increasing vulnerability to side-channel attacks. See NIST PQC Round 3 Report, 2022.

13. NIST announcement: "NIST Selects HQC as Fifth Algorithm for Post-Quantum Encryption." March 11, 2025. HQC is based on Hamming Quasi-Cyclic codes. Draft standard expected 2026, final standard 2027.

14. HQC key and ciphertext sizes at Level 3 (≈192-bit security): public key approximately 4,522 bytes, ciphertext approximately 9,042 bytes. Compare to ML-KEM-768: public key 1,184 bytes, ciphertext 1,088 bytes. Source: HQC specification, https://pqc-hqc.org

Next: Chapter 4 — The Regulatory Landscape

CHAPTER 4

The Regulatory Landscape

If the first three chapters answered "why should we care?" and "what's at risk?", this chapter answers the question that gets CISOs and program managers out of their chairs: "Who says we have to do this, and by when?"

The PQC regulatory landscape is complex and actively evolving. It spans federal law, executive policy, agency-specific mandates, standards body timelines, and international frameworks—each with different enforcement mechanisms and deadlines. This chapter organizes all of it so you can determine exactly which requirements apply to your organization.

First Principles: Law vs. Policy vs. Guidance

Before we map the specific mandates, we need to establish a critical distinction that the PQC conversation often blurs: **not all mandates are created equal.** Understanding the hierarchy matters because it tells you what survives a change in administration and what doesn't.

Type	What It Is	Can It Be Rescinded?	PQC Examples
Federal Law	Passed by Congress, signed by the President. Requires an act of Congress to repeal.	**No. Survives any administration change.**	Quantum Computing Cybersecurity Preparedness Act
National Security Memorandum	Presidential directive on national security policy. Binding on federal agencies.	Yes, by the sitting President. But rarely done.	NSM-10 (not rescinded by Trump admin)
Executive Order	Presidential directive on federal operations. Binding on executive branch agencies.	Yes. Commonly modified or rescinded by subsequent presidents.	EO 14306 (Trump, June 2025)
OMB Memo	Implementation guidance from the Office of Management and Budget. Binding on federal agencies.	Yes, by OMB. But typically remains until replaced.	M-23-02 (crypto inventories)
Agency Guidance / Standard	Recommendations and standards from NIST, NSA, CISA. Compliance often required by federal procurement or accreditation frameworks.	Standards can be revised but rarely fully withdrawn.	NIST IR 8547, CNSA 2.0, CISA PQC product list

This hierarchy matters enormously. The Quantum Computing Cybersecurity Preparedness Act is **federal law**—no executive order can override it. NSM-10 has not been rescinded. NIST standards are embedded in federal procurement requirements across the entire government. Understanding which mandates are durable versus which are politically contingent is essential for building a migration plan that survives the next election cycle.[1]

United States: The Federal PQC Framework

The Quantum Computing Cybersecurity Preparedness Act (Federal Law)

Signed into law on December 21, 2022, this is the **bedrock of US PQC policy**—the one mandate that cannot be rescinded by any president.[2]

The Act requires:

- OMB to issue guidance on migrating federal IT systems to PQC (due within one year of NIST standards publication—approximately August 2025)
- Federal agencies to maintain cryptographic inventories of systems vulnerable to quantum attack
- Agencies to develop and submit PQC migration plans
- Annual progress reports to Congress

Whether OMB actually met the August 2025 statutory deadline for issuing migration guidance remains unclear in the public record as of early 2026. But the legal obligation on agencies to inventory and plan is active regardless.[3]

NSM-10: The Policy Foundation

National Security Memorandum 10, "Promoting United States Leadership in Quantum Computing While Mitigating Risks to Vulnerable Cryptographic Systems," was signed by President Biden on May 4, 2022.[4] NSM-10 established the **2035 end-state target** for completing the transition to quantum-resistant cryptography across federal systems.

Key requirements include:

- FCEB agencies submit annual inventories of quantum-vulnerable IT systems
- Agencies develop and submit transition plans with specific timelines
- NSA provides CNSA 2.0 guidance for National Security Systems
- CISA engages critical infrastructure partners on PQC readiness

> ⚠ **MANDATE ALERT** NSM-10 has not been rescinded by the Trump administration. EO 14306 (June 2025) explicitly references NSM-10 as the foundational document for the PQC transition. The January 20, 2025 mass rescission of Biden-era executive orders did not target NSM-10.

Executive Order 14306 (June 2025): What Changed

President Trump's EO 14306, "Sustaining Select Efforts to Strengthen the Nation's Cybersecurity," modified the Biden-era PQC framework in important ways.[5] Understanding what was kept, what was removed, and what was added is critical for compliance planning:

What was preserved:

- CISA must maintain and regularly update a list of product categories where PQC-capable products are widely available (published January 2026)
- NSA and OMB must issue requirements for TLS 1.3 or successor support by January 2, 2030
- The quantum threat language—including identification of China as the most active cyber threat—was retained verbatim

What was removed:

- The requirement for agencies to adopt PQC "as soon as practicable" upon vendor support
- The mandate for agencies to require vendors to implement PQC in procurement
- Provisions for international promotion of NIST algorithm adoption

The practical effect: EO 14306 loosened the federal urgency of PQC adoption while preserving the direction. Agencies retain latitude on implementation pace, but the destination (quantum-resistant cryptography by 2035) remains unchanged.

CNSA 2.0: The NSA's Timeline for National Security Systems

For organizations operating in or selling to National Security Systems (NSS) environments—DoD, intelligence community, classified networks—the NSA's Commercial National Security Algorithm Suite 2.0 is the binding standard.[6] CNSA 2.0 is more aggressive than the general federal timeline:

System Category	Support & Prefer CNSA 2.0 By	Exclusively Use CNSA 2.0 By	Notes
Software & firmware signing	**2025 (now)**	**2030**	Earliest mandatory deadline
Web browsers, servers, cloud	**2025 (now)**	2033	
Networking (VPNs, routers)	**2026**	**2030**	F5 BIG-IP, NGINX in scope
Operating systems	2027	2033	
New NSS acquisitions	**January 1, 2027**	—	All new purchases must be CNSA 2.0
Niche / constrained devices	2030	2033	IoT, PKI systems
Legacy / custom applications	—	2033 (update or replace)	Hard deadline

CNSA 2.0 specifies exact algorithms: **ML-KEM-1024** for key establishment and **ML-DSA-87** for general-purpose digital signatures (Level 5 security). For software and firmware signing specifically, **LMS and XMSS** (stateful hash-based signatures from SP 800-208) are approved for immediate use, with ML-DSA approved once FIPS-validated implementations become available.[7]

DoW CIO Direction: The November 2025 Memorandum

CNSA 2.0 specifies the algorithms. NSM-10 sets the destination. But for organizations that operate or deliver into Department of War networks, a third document now sits on top of both: the DoW CIO memorandum *Preparing for Migration to Post Quantum Cryptography*, issued November 18, 2025 by Katherine Arrington (Performing the Duties of the CIO).[13]

The memo creates a new, DoW-specific approval gate. Every DoW Component—Services, Combatant Commands, Defense Agencies, and Field Activities—must now coordinate cryptographic migration through a centralized **DoW CIO PQC Directorate**, led by Dr. Britta Hale. Any "engagement" with PQC technology—defined broadly as testing, evaluating, piloting, researching, investing in, prototyping, demonstrating, implementing, integrating, or any planned or actual acquisition—requires two new authorizations issued by the Directorate:

- **Cryptographic intake approval,** before any test, evaluation, pilot, investment, or acquisition begins.
- **Cryptographic deployment approval,** before any PQC-enabling or PQC-related technology is deployed. Deployment approval is informed by Intelligence Community, NIST, and NSA certification outcomes—meaning FIPS 140-3, NIAP Common Criteria, and NSA CSfC validation are inputs to the decision, not substitutes for it.

If the Directorate identifies issues that cannot be mitigated, the technology is removed from engagement and use immediately. This is a real cessation authority, not a comment-and-recommend role.

The memo also issues three immediate prohibitions and two phase-out deadlines that affect every DoW Component's procurement pipeline today:

- **Quantum confidentiality and keying technologies are out, effective immediately.** No testing, piloting, use, or procurement of QKD; QKD combined with other cryptographic key establishment; quantum communications or networking; non-local quantum randomness generation; or non-FIPS random number generation for confidentiality, authenticity, integrity, key distribution, or randomness generation—absent specific Directorate exception. Chapter 7 explains why this technical position is not new; the November 2025 memo makes it binding for DoW.
- **Commercial PSK-based "quantum resistance" solutions are out, effective immediately.** Pre-shared keys provisioned through NSA KMI for Type 1 devices remain permitted. All other commercial PSK-based quantum-resistance approaches are prohibited from new test, pilot, use, or procurement actions.
- **Commercial symmetric key establishment, key agreement, and key distribution protocols for quantum resistance are out, effective immediately—**same prohibition scope as PSKs.
- **By December 31, 2030,** non-KMI PSK solutions and symmetric key establishment/agreement/distribution protocols must be phased out and replaced with NIST-approved (CNSA 2.0–listed for NSS) asymmetric PQC key establishment.
- **By December 31, 2031,** the same phase-out applies to solutions currently registered with NSA CSfC. Pre-2010 symmetric key distribution use cases are explicitly grandfathered as legacy.

A DoW PQC Strategy is referenced repeatedly throughout the memo and is in preparation. It will be the master execution document. As of this writing, it has not been published; new requirements, approval processes, and updates are being maintained centrally at https://cybersecurityks.osd.mil/DoDcs/pqc. Component-level PQC migration leads were due to the Directorate within twenty days of the memo's issuance, with annual updates each September 30.[14]

> ⚠ **MANDATE ALERT** The DoW CIO PQC Directorate's intake and deployment approvals are an **additional** gate on top of FIPS 140-3, NIAP Common Criteria, and NSA CSfC—not a replacement. A DoW Component cannot lawfully acquire, pilot, deploy, or use any PQC-enabling or PQC-related technology without explicit Directorate approval, even when the underlying product holds every existing federal certification. As of this writing, the operational machinery for this approval (forms, criteria, SLAs, vendor-facing process documentation) is still being stood up; expect this to evolve as the DoW PQC Strategy is published and the Directorate matures.

The practical implication for vendors and integrators: a product that has cleared FIPS 140-3, NIAP CC, and CSfC is no longer sufficient on its own to be acquired or deployed by a DoW Component for PQC purposes. The Directorate's intake and deployment approvals are now decision points in the procurement path, and the artifacts the Directorate will demand—test plans, test results, acquisition artifacts, risk mitigations—should be assembled in parallel with traditional certification work, not after.[15]

NIST IR 8547: The Deprecation Timeline

Published in November 2024 as an initial public draft, NIST IR 8547 ("Transition to Post-Quantum Cryptography Standards") established for the first time a formal deprecation schedule for quantum-vulnerable algorithms.[8]

- **Deprecated after 2030:** All quantum-vulnerable algorithms at the 112-bit security level (RSA-2048, ECC P-256, DH-2048, etc.). "Deprecated" means the algorithm is still permitted but actively discouraged; new systems should not use it.
- **Disallowed after 2035:** All quantum-vulnerable public-key algorithms at any security strength. "Disallowed" means NIST-compliant systems cannot use the algorithm at all. Period.

NIST IR 8547 also supports hybrid cryptographic solutions during the transition—combining classical and PQC algorithms so that the system remains secure as long as at least one algorithm holds. This is particularly relevant for organizations that want to begin migration now without waiting for full PQC ecosystem maturity.

International: The Global Picture

United Kingdom — NCSC Three-Phase Roadmap

The UK's National Cyber Security Centre (part of GCHQ) published its PQC migration timeline in March 2025—the first major regulatory jurisdiction to endorse NIST's standardized algorithms and set concrete deadlines.[9]

- **By 2028:** Complete discovery—identify all cryptographic services needing upgrades, build a migration plan, create a cryptographic inventory.
- **By 2031:** Execute high-priority upgrades for critical systems and refine migration plans as PQC standards mature.
- **By 2035:** Complete migration across all systems, services, and products.

The NCSC prioritizes critical national infrastructure: NHS healthcare systems, City of London financial services, defense, and government. For most SMEs, the transition will happen through routine vendor updates. Larger organizations must take active ownership.

European Union — NIS2 PQC Roadmap

The EU's approach is structured through the NIS Cooperation Group's coordinated implementation roadmap, published in early 2025.[10]

- **By end of 2026:** Member states initiate national PQC transition strategies.
- **By 2030:** Transition critical infrastructure to PQC.
- **By 2035:** Complete migration for as many systems as practically feasible.

In January 2026, the European Commission published a proposed directive amending NIS2 to include **explicit post-quantum cryptography requirements** written directly into the directive text for the first time. ENISA has published guidance recommending hybrid PQ/T schemes (combining classical algorithms with PQC) to smooth interoperability during transition.

Financial Sector — G7 Roadmap

The G7 Cyber Expert Group released a financial sector PQC roadmap on January 13, 2026, co-chaired by the US Treasury and Bank of England.[11] It targets critical financial systems for migration by 2030–2032 and full transition by 2035. FINRA, FS-ISAC, and national financial regulators are expected to align sector-specific guidance with this framework.

Sector-Specific Considerations

The PQC mandates don't apply uniformly. Your migration urgency depends on your sector:

Sector	Primary PQC Drivers	Key Dates
DoD / Intelligence	CNSA 2.0, NSM-10, CNSSP 15, DoW CIO PQC Memo (Nov 2025). Mandatory. DoW Components must obtain intake and deployment approval from the DoW CIO PQC Directorate. No waivers without explicit Directorate or NSA approval.	New acquisitions: Jan 2027. Networking: 2030. Full: 2033–2035.
Federal Civilian (FCEB)	Preparedness Act, NSM-10, M-23-02, EO 14306, NIST IR 8547. Compliance tracked via FISMA.	Transition plans: Apr 2026. Deprecated: 2030. Disallowed: 2035.
Federal Contractors	CISA PQC product list, DFARS/CMMC for DoD suppliers, agency-specific acquisition rules (e.g., USDA AGAR).	PQC readiness expected: Jan 2027. Procurement pressure accelerating.
Financial Services	G7 roadmap, FINRA guidance, FS-ISAC publications. HNDL risk acute for transaction data.	G7 targets critical systems: 2030–2032. Full: 2035.
Healthcare	HIPAA "reasonable safeguards" (evolves with technology). Long data lifetimes (medical records are permanent).	No explicit PQC mandate yet. HNDL risk is extreme due to data lifetime.

Sector	Primary PQC Drivers	Key Dates
Critical Infrastructure	CISA PQC Initiative, EU NIS2. OT/ICS environments have unique migration challenges.	EU: critical infrastructure by 2030. US: varies by sector.
Private Sector (General)	No direct federal mandate (unless federal contractor). Market pressure from PQC-ready competitors and customers.	Follow NIST standards. Plan with 2030 deprecation in mind.

Sector Acquisition Lifecycles: When PQC Requirements Bite

The mandates described above ride on top of existing acquisition frameworks. Understanding when a regulatory date ("CNSA 2.0 for all new NSS acquisitions by January 1, 2027") intersects a program's actual procurement lifecycle matters because the framework, not the calendar, governs when PQC requirements get baked into contracts, solicitations, and systems engineering reviews. Three frameworks dominate federal sector acquisition.[12]

FAA Acquisition Management System (AMS). Codified at fast.faa.gov, the AMS governs FAA capital investments through six lifecycle phases (Service Analysis & Strategic Planning, Concept & Requirements Definition, Initial Investment Analysis, Final Investment Analysis, Solution Implementation, In-Service Management) with decision points overseen by the Joint Resources Council. PQC requirements typically enter at Concept & Requirements Definition through the Information Systems Security Engineering (ISSE) process and are finalized in the Final Investment Analysis phase as part of the Solicitation Information Request, Statement of Work, and Contract Data Requirements List.

DoDI 5000.02 Adaptive Acquisition Framework (AAF). Reissued January 23, 2020 with Change 1 (June 8, 2022), DoDI 5000.02 replaced the traditional one-size-fits-all model with six tailorable pathways: Urgent Capability Acquisition, Middle Tier of Acquisition (rapid prototyping and rapid fielding), Major Capability Acquisition, Software Acquisition, Defense Business Systems, and Acquisition of Services. PQC entry points vary by pathway. Major Capability Acquisition has the most structured milestone gates (Milestone A/B/C) at which PQC requirements can be specified; Software Acquisition moves fastest through iterative deliveries; Middle Tier rapid fielding programs must complete within five years of program start and typically inherit PQC requirements from the parent system.

NASA NPR 7120.5F (Space Flight Program and Project Management). NASA space flight programs follow a two-phase structure—Formulation and Implementation—subdivided into Phase A through Phase F with Key Decision Points (KDPs) as approval gates. PQC requirements for NASA systems enter primarily at Phase A (Concept & Technology Development) for new missions; for existing missions in Phase E (Operations & Sustainment), PQC arrives through capability upgrades or technology refresh cycles. Ground systems supporting space flight operations are explicitly within NPR 7120.5F scope.

The practical implication: a regulatory date is a constraint on when PQC capability must be present in a deployed system, not a directive for how a program acquires it. A DoD Major Capability Acquisition program starting in 2026 may not field initial operational capability until 2030–2033—so PQC must be in the requirements baseline at program start to meet CNSA 2.0's Jan 2027 "new acquisitions" gate, even though the deployed system won't exist until years later. Appendix G provides a detailed crosswalk mapping the book's five-phase PQC migration model against each of these frameworks.

The Master Timeline: Everything in One View

This is the single consolidated reference. We've mapped every major PQC milestone across all mandates and jurisdictions. **Tear this page out.**

Date	Milestone	Source
Aug 2024	NIST publishes FIPS 203/204/205—first finalized PQC standards	NIST
Nov 2024	NIST IR 8547 published—formal deprecation timeline announced	NIST
Mar 2025	NIST selects HQC as fifth PQC algorithm; UK NCSC publishes 3-phase roadmap	NIST, NCSC
Jun 2025	EO 14306 modifies federal PQC posture (preserves direction, loosens urgency)	White House
Nov 2025	**DoW CIO PQC Memo: new intake/deployment approval gate; QKD and commercial PSK/symmetric KE prohibited immediately**	DoW CIO
Jan 2026	CISA publishes PQC product categories list for federal procurement	CISA / EO 14306
Apr 2026	**FCEB agencies submit PQC transition plans (crypto inventory + roadmap)**	NSM-10 / M-23-02
2026–2027	CNSA 2.0: Networking equipment must support PQC. FN-DSA and HQC standards expected.	NSA, NIST
Jan 2027	**All new NSS acquisitions must be CNSA 2.0 compliant. Federal contractors demonstrate PQC readiness.**	CNSA 2.0 / EO 14306
2028	UK NCSC Phase 1 complete: all orgs have crypto inventory and migration plan	UK NCSC
Jan 2030	**All federal systems support TLS 1.3+. CNSA 2.0 exclusive for VPNs/routers and software signing. NIST deprecates 112-bit public-key algorithms. DoW phase-out of non-KMI PSK and symmetric KE protocols complete (CSfC-registered: Dec 2031).**	EO 14306, CNSA 2.0, NIST
2030–2032	EU: critical infrastructure to PQC. G7: critical financial systems migrated. UK Phase 2 complete.	EU NIS2, G7, NCSC
2033	CNSA 2.0 exclusive for web/cloud, operating systems, niche devices, legacy replacement	NSA CNSA 2.0
2035	**NIST disallows all quantum-vulnerable public-key algorithms. NSM-10 full migration target. EU/UK complete.**	NIST, NSM-10, EU, UK

What This Means for You

If you operate NSS or sell to DoD: CNSA 2.0 is your binding standard. New acquisitions must be compliant by January 2027. You are already behind on software signing.

If you're a federal civilian agency: Your crypto inventory and transition plan should be in progress or submitted. 2030 deprecation means no new systems with RSA/ECC after that date.

If you're a federal contractor: PQC readiness is becoming a procurement requirement. The CISA product categories list is already shaping buying decisions. If your products don't support PQC, you risk being excluded from future contracts.

If you're in the private sector: No direct federal mandate (unless you're a contractor), but the NIST deprecation timeline will cascade through every compliance framework that references NIST standards—which is nearly all of them. If your customers are in regulated industries, they will require PQC from their vendors.

If you operate internationally: The UK, EU, and G7 timelines are converging on 2035 with intermediate milestones. Multinational organizations must track requirements across multiple jurisdictions.

> **PLAIN-LANGUAGE SIDEBAR** The bottom line: regardless of your sector, 2035 is the hard stop. But the real operational deadlines are earlier—January 2027 for DoD, 2028 for UK discovery, 2030 for NIST deprecation and CNSA 2.0 networking. The organizations that start now will migrate calmly. The ones that wait until 2030 will discover what panic looks like at enterprise scale.

What's Next

You now know why you need to migrate, what is broken, what replaces it, and who says you have to. The next question is intensely practical: how do you find all the cryptography in your environment? Chapter 5 walks through the cryptographic discovery process—building the inventory that every migration plan depends on.

Notes

The following sources support specific claims made in Chapter 4. Full bibliographic entries appear in the Bibliography.

1. For analysis of the US PQC regulatory framework hierarchy, see: "The Complete US Post-Quantum Cryptography (PQC) Regulatory Framework in 2026," postquantum.com (February 2026). This source provides detailed analysis of which mandates survive administration changes.

2. Quantum Computing Cybersecurity Preparedness Act, Pub. L. No. 117-349, signed December 21, 2022. As federal statute, its requirements cannot be rescinded by executive order.

3. Whether OMB issued the migration guidance required by the Preparedness Act by the August 2025 statutory deadline is unclear. An OMB draft memorandum circulated in July 2025 would direct agencies to fully migrate to PQC standards and require third-party vendors to disclose phased PQC transition timelines, but it has not been finalized as of March 2026.

4. The White House. National Security Memorandum on Promoting United States Leadership in Quantum Computing While Mitigating Risks to Vulnerable Cryptographic Systems (NSM-10), May 4, 2022.

5. Executive Order 14306, "Sustaining Select Efforts to Strengthen the Nation's Cybersecurity and Amending Executive Order 13694 and Executive Order 14144," signed June 6, 2025. Analysis based on comparison of EO 14144 (Biden, January 2025) with EO 14306 (Trump, June 2025).

6. NSA. “Commercial National Security Algorithm Suite 2.0 (CNSA 2.0) Algorithms.” PP-22-1338, Ver. 1.0, September 2022. FAQ updated to Ver. 2.1, December 2024. Timeline by system category from the CNSA 2.0 Algorithm Guidance document.

7. CNSA 2.0 approved algorithms: ML-KEM-1024 (key establishment), ML-DSA-87 (general signatures), LMS/XMSS per SP 800-208 (software/firmware signing), AES-256 (symmetric), SHA-384/512 (hashing). Note: SLH-DSA is NOT part of CNSA 2.0 and is not approved for NSS.

8. NIST IR 8547 (Initial Public Draft), “Transition to Post-Quantum Cryptography Standards.” November 12, 2024. Tables 2 and 4 list quantum-vulnerable algorithms with deprecation after 2030 and disallowance after 2035.

9. UK National Cyber Security Centre. “Timelines for Migration to Post-Quantum Cryptography.” Published March 2025. Three-phase roadmap: Phase 1 (to 2028), Phase 2 (2028–2031), Phase 3 (2031–2035).

10. NIS Cooperation Group. “Coordinated Implementation Roadmap for the Transition to Post-Quantum Cryptography.” Published early 2025. European Commission proposed directive amending NIS2 with explicit PQC requirements published January 2026.

11. G7 Cyber Expert Group. Financial Sector PQC Roadmap. Published January 13, 2026. Co-chaired by US Treasury and Bank of England. Targets critical financial systems for migration by 2030–2032.

12. Sector acquisition framework references: FAA AMS authoritative policy at the FAA Acquisition System Toolset (fast.faa.gov); DoDI 5000.02, “Operation of the Adaptive Acquisition Framework,” January 23, 2020, Change 1 (June 8, 2022); NASA NPR 7120.5F, “NASA Space Flight Program and Project Management Requirements” (current revision). See Appendix G for the full crosswalk mapping these frameworks to the book’s five-phase PQC migration model.

13. DoW CIO. Memorandum, “Preparing for Migration to Post Quantum Cryptography,” November 18, 2025. Signed by Katherine Arrington, Performing the Duties of the Chief Information Officer of the Department of War. https://dodcio.defense.gov/Portals/0/Documents/Library/PreparingForMigrationPQC.pdf

14. Per the November 18, 2025 memo, Component PQC migration leads must be reported to the Directorate within twenty days of the memo’s issuance and updated annually by September 30. Centralized requirements and approval-process documentation are maintained at https://cybersecurityks.osd.mil/DoDcs/pqc. Point of contact: osd.pentagon.dodcio.mesg.dcio-cs-pqc@mail.mil.

15. The DoW CIO memo’s authorization regime is “in addition to”—not a substitute for—Committee on National Security Systems Policy 15, “Use of Public Standards for Secure Information Sharing,” December 2024, and Chairman of the Joint Chiefs of Staff Instruction 6510.02, “Cryptographic Modernization Planning,” August 16, 2022.

Next: Chapter 5 — Know What You Have: Cryptographic Discovery

CHAPTER 5

Know What You Have: Cryptographic Discovery

Every PQC migration plan begins with the same question: “Where is cryptography in my environment?” The answer, invariably, is “more places than you think.”

Cryptography is embedded in every layer of modern IT infrastructure—TLS certificates on web servers and load balancers, SSH keys on Linux hosts, IPsec tunnels between sites, code signing certificates in CI/CD pipelines, S/MIME certificates in email clients, API tokens, database encryption, disk encryption, HSMs, smart cards, and dozens of applications that implement their own cryptographic functions. Most organizations have no comprehensive inventory of these assets. That’s the problem this chapter solves.

Without a cryptographic inventory, you’re migrating blind. You can’t prioritize what you can’t see, you can’t track progress against what you haven’t catalogued, and you can’t prove compliance with mandates that require you to “submit cryptographic inventories.”[1]

Why Discovery Is Harder Than It Sounds

If you’ve ever tried to audit certificates across an enterprise network, you know the pain. Cryptographic assets are scattered, undocumented, and often invisible to traditional IT management tools. The challenge has several dimensions:

- **Volume:** A typical enterprise has thousands to tens of thousands of certificates and keys across its environment. Federal agencies may have hundreds of thousands.
- **Diversity:** Cryptography exists in certificates, connection configurations, application code, hardware modules, firmware, IoT devices, and third-party SaaS services—each requiring different discovery techniques.
- **Opacity:** Many systems use cryptography without exposing it to administrators. An application may negotiate TLS internally without any external indication of which cipher suite or key exchange algorithm it’s using.
- **Sprawl:** Multi-cloud, hybrid, and edge architectures mean cryptographic assets span on-premises data centers, AWS/Azure/GCP regions, CDN edge nodes, and partner networks.
- **Fragmentation:** No single team owns cryptography. Network engineers own TLS termination. Security teams own certificates. Developers own code signing. IAM teams own authentication. PKI teams own the CA hierarchy. Each has partial visibility; none has the full picture.

> **PLAIN-LANGUAGE SIDEBAR** Think of cryptographic discovery like a building inspection for electrical wiring. You know there’s wiring in every wall—but you don’t know exactly where every wire runs, what gauge it is, or whether it meets current code. The only way to find out is to look—and to look systematically, because missing one segment could mean a fire.

Three Approaches to Cryptographic Discovery

NIST SP 1800-38B ("Quantum Readiness: Cryptographic Discovery") identifies a multi-faceted approach to discovery.[2] No single method catches everything. A comprehensive inventory requires combining at least three techniques:

Discovery Type	What It Finds	Limitations
Network Scanning	TLS/SSL versions, cipher suites, certificate details, and key exchange algorithms for any connection visible on the wire. Passive or active.	Sees only what crosses the network. Misses data-at-rest encryption, application-internal crypto, and east-west traffic in microsegmented environments.
Endpoint Scanning	Installed certificates, key stores, cryptographic libraries (OpenSSL, BoringSSL, NSS, Java KeyStore), SSH keys, disk encryption configurations.	Requires agent or authenticated access to each endpoint. Incomplete for unmanaged devices, IoT, and shadow IT.
Application Testing	Crypto functions embedded in application code, API calls, hardcoded keys, custom TLS configurations, and third-party library dependencies.	Requires SAST/DAST integration or manual code review. Doesn't scale easily across large application portfolios.

The key insight from NIST's work: **most of the data items required for PQC compliance (algorithm in use, key size, protocol version, certificate authority, expiration date) cannot yet be fully collected with automated tools alone.** NIST SP 1800-38B found that only three of the nine data items required by OMB M-23-02 could be collected automatically. The rest require manual effort or enrichment.[3]

What to Catalog: The Cryptographic Bill of Materials

Your cryptographic inventory—sometimes called a **Cryptographic Bill of Materials (CBOM)**—should capture the following for each asset:

- **Algorithm:** Which cryptographic algorithm is in use (RSA-2048, ECDHE-P256, AES-128-GCM, SHA-1, etc.)
- **Function:** What the algorithm does (key exchange, digital signature, bulk encryption, hashing)
- **Protocol context:** Which protocol the algorithm operates within (TLS 1.2, IPsec IKEv2, SSH 2.0, S/MIME)
- **System/application:** Which system or application uses this cryptographic configuration
- **Data classification:** The sensitivity of the data protected by this algorithm (ties back to the HNDL risk matrix from Chapter 2)
- **Owner:** The team or individual responsible for the system
- **Quantum risk status:** Broken by Shor's, weakened by Grover's, or safe (directly from the Chapter 2 vulnerability map)
- **Migration priority:** Based on data sensitivity, exposure to HNDL, and regulatory deadline

⚠ **MANDATE ALERT** OMB M-23-02 requires federal agencies to report nine specific data items for each cryptographic system. CISA's Automated Cryptography Discovery and Inventory (ACDI) strategy is building toward automated collection, but the tools are not yet mature enough to cover the full requirement. Plan for significant manual effort in the first inventory pass, then work toward automation for ongoing maintenance.

Don't Boil the Ocean: A Phased Discovery Approach

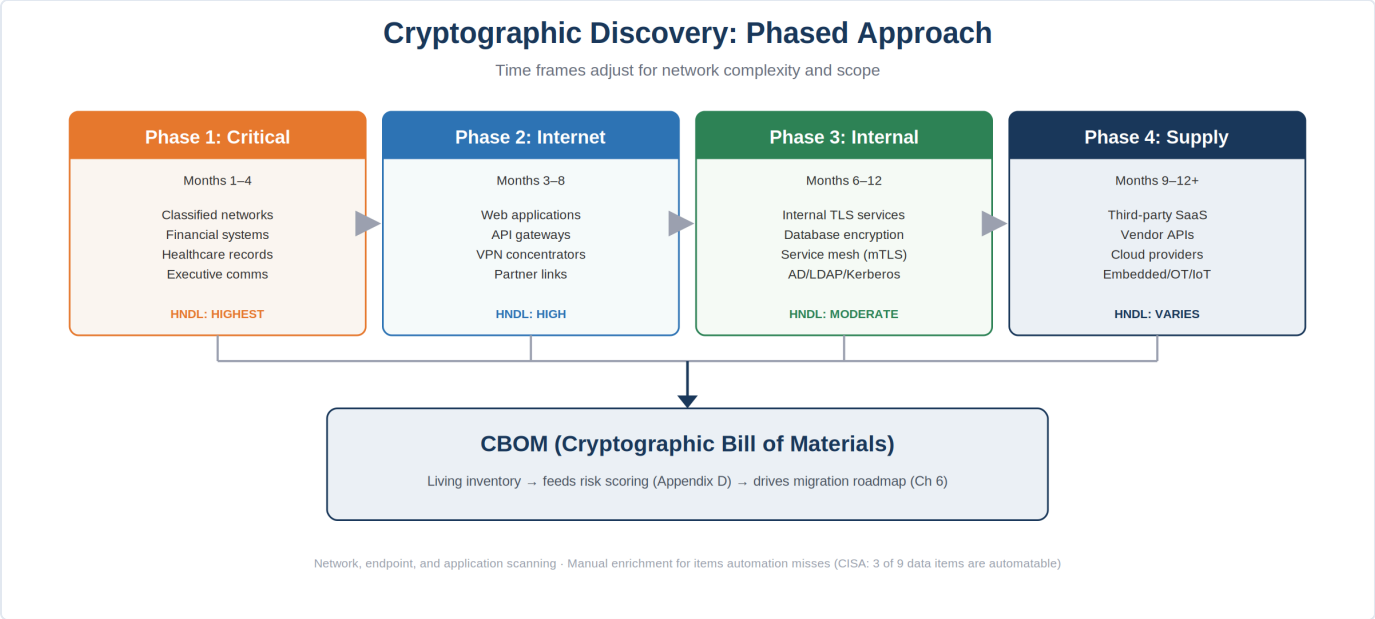

Figure 5.1 — Cryptographic Discovery: Four Phases

Attempting to discover every cryptographic asset across your entire environment simultaneously is a recipe for paralysis. Instead, take a phased approach driven by business criticality:

Phase 1: Critical Systems (Months 1–4)

Note: these time frames will adjust for network complexity and scope. A 20 person small business scales differently than a global 200k person enterprise.

Focus on systems that protect CRITICAL and HIGH-tier data (per Chapter 2's HNDL risk matrix): classified networks, financial transaction systems, healthcare records, executive communications, and any system processing data with a sensitivity lifetime exceeding 10 years. Start with network-facing systems where HNDL exposure is greatest.

Phase 2: Internet-Facing and Partner Connections (Months 3–8)

Expand to all internet-facing TLS endpoints (web applications, API gateways, CDN configurations), VPN concentrators, partner-to-partner encrypted links, and email infrastructure. These systems have the highest interception likelihood.

Phase 3: Internal Infrastructure (Months 6–12)

Cover internal certificate infrastructure, east-west TLS between microservices, database encryption, code signing and CI/CD pipelines, SSH key estates, and endpoint disk encryption. This phase is the longest because it touches the broadest surface area.

Phase 4: Sustain and Automate (Ongoing)

Transition from project-based discovery to **continuous inventory management.** Integrate discovery into CI/CD pipelines so new applications are inventoried at deployment. Embed cryptographic checks into security scanning (SAST/DAST). Make the CBOM a living document, not a one-time snapshot.

Inventory at Source: The Long-Term Strategy

Discovery tools are essential for the initial inventory, but they're not a sustainable long-term strategy on their own. The more mature approach is to **address inventory at source**—ensuring that new cryptographic assets are added to the inventory the moment they're implemented, rather than waiting for a periodic scan to find them.[4]

Practical ways to implement inventory at source:

- **CI/CD integration:** Add cryptographic checks to your deployment pipeline. When a new application is deployed, its TLS configuration, certificate chain, and cipher suite preferences are automatically catalogued.
- **Certificate lifecycle management:** If you're issuing certificates through an enterprise CA (or using a CLM platform like Venafi, Keyfactor, or AppViewX), every certificate issuance should automatically update the CBOM.
- **Infrastructure-as-Code:** If TLS configurations are defined in Terraform, Ansible, or similar tools, the inventory can be derived directly from the IaC repository.
- **Procurement requirements:** When acquiring new hardware or software, require vendors to disclose the cryptographic algorithms and library versions used. This becomes part of your supply chain risk management.

The goal is to shift discovery from being the primary method of inventory population to a validation and exception-finding mechanism. New assets are inventoried at source; discovery scans catch anything that slipped through the cracks.

The Opportunity: Cleaning Up Cryptographic Debt

Here's the silver lining in the PQC discovery process: **it forces you to confront cryptographic debt you've been carrying for years.**

When you start scanning your environment, you will find things that shouldn't be there—and they won't all be quantum-related. You'll find:

- Expired certificates nobody renewed (or knew about)

- SHA-1 signatures still in production (classically broken since 2017)
- TLS 1.0 and 1.1 connections that should have been retired years ago
- 3DES cipher suites still negotiated by legacy clients
- Self-signed certificates in production systems with no rotation schedule
- RSA-1024 keys that haven't been updated since they were deployed in 2008
- SSH keys that have never been rotated and are shared across teams

The PQC migration is an opportunity to clean house. Frame it that way for your leadership: "We're not just preparing for quantum—we're fixing the cryptographic hygiene issues we've been deferring for a decade." That framing turns a compliance exercise into a genuine security improvement, which is a much easier budget conversation.

> **F5 PERSPECTIVE** **Strategic Points of Control: How F5 Enables Cryptographic Discovery** The following section describes how F5 capabilities can support the cryptographic discovery process. This is vendor-specific guidance—the methodology described above applies regardless of your infrastructure stack. F5 devices—BIG-IP, SSL Orchestrator, NGINX—sit at strategic control points in the network where application traffic converges: between users and applications, between applications and APIs, between sites, and between cloud environments. This positioning provides a unique vantage point for cryptographic visibility that most organizations already have deployed but aren't fully leveraging for PQC readiness. **BIG-IP SSL Orchestrator (SSLO)** decrypts and re-encrypts TLS traffic at line speed using F5's full-proxy architecture. For every connection that passes through SSLO, the system sees the complete cryptographic handshake: the cipher suite negotiated, the key exchange algorithm used, the certificate presented, the TLS version, and the certificate chain. This means SSLO already possesses the raw data needed for network-level cryptographic inventory—the question is how to extract, catalog, and act on it. In a PQC discovery context, SSLO can identify: • Which connections still negotiate RSA, ECDHE, or DH key exchange (Shor's-vulnerable) • Which server certificates use RSA or ECDSA signatures (Shor's-vulnerable) • Which connections use AES-128 vs. AES-256 (Grover's exposure) • Which connections still use TLS 1.0/1.1 or weak cipher suites (cryptographic debt) • The volume and frequency of connections by cryptographic profile (prioritization data) **F5 Application Study Tool (AST) and F5 Insight** extend this visibility into operational dashboards. AST is an open-source tool built on OpenTelemetry, Prometheus, and Grafana that collects telemetry data from BIG-IP devices across your fleet. F5 Insight (announced March 2026 as part of the F5 ADSP platform) builds on AST's foundation to provide end-to-end observability with AI-assisted analysis. For PQC compliance, the combination of SSLO + AST/Insight enables a network-centric view of your cryptographic posture: which algorithms are in active use, at what volume, protecting what traffic categories, and—critically—which connections are still using quantum-vulnerable configurations. This data feeds directly into the CBOM and supports the risk-based prioritization framework described earlier in this chapter. **BIG-IP v21.1** (announced March 2026) adds NIST-compliant PQC cipher support with hybrid TLS cipher groups, allowing organizations to enable PQC protection while maintaining backward compatibility with classical configurations. BIG-IP Zero Trust Access (formerly APM) adds quantum-resistant TLS and SSL VPN tunneling. This means F5 isn't just helping you discover your quantum exposure—it's also providing the upgrade path for the infrastructure you're likely using to terminate and inspect that traffic.

Running a Discovery Pilot: Proof of Value

Before committing to a full enterprise discovery program, run a well-scoped pilot. This serves two purposes: it validates your toolset and methodology, and it produces concrete findings that justify the budget for the full inventory.[5]

Pilot Scope Recommendations

- **Select 2–3 representative environments** (e.g., one internet-facing web application environment, one VPN/IPsec infrastructure, one internal certificate domain)
- **Run all three discovery types** (network scan, endpoint scan, application review) on the pilot scope to compare coverage and gap areas
- **Allocate 4–6 weeks for the pilot,** including analysis and report generation
- **Document everything you find** that wasn't expected—this is your "cryptographic debt" evidence, and it's your most compelling argument for broader investment
- **Expect surprises.** NIST's NCCoE work found that automated discovery tools routinely reveal hundreds or thousands of cryptographic assets that organizations didn't know existed[6]

The pilot report becomes your proof of value—evidence that the quantum-vulnerable surface area is real, quantifiable, and larger than leadership assumed. That report is how you get the resources for Phases 1–4.

> **PLAIN-LANGUAGE SIDEBAR** In highly-regulated environments, pilot work needs vocabulary that auditors and mission owners recognize. Two patterns are worth naming before you start. **Concurrent shadow operation means** running the new PQC-capable path alongside the classical path in production, with the classical path as the safety net. Hybrid TLS 1.3 (X25519MLKEM768) is the canonical example: the handshake negotiates both a classical and a post-quantum shared secret, and the session key is derived from both. The PQC contribution "shadows" the classical exchange—if ML-KEM fails for any reason (library bug, implementation defect, side-channel discovery), the classical portion alone is sufficient to keep connections working. You gain production evidence about PQC behavior without gambling operational stability on it. **Regional rollout means** enabling PQC in one region, availability zone, or mission enclave first, instrumenting it thoroughly, then expanding outward. The pattern is feature-flag management applied to cryptography—what's new is the measurement burden: handshake latency distribution (p50/p95/p99), client failure rates, certificate-size fragmentation impact (Chapter 8), and HSM throughput under PQC load (Chapter 9). For federal and DoD environments, "region" often maps to a mission-specific enclave or security domain rather than geography. Both patterns carry forward into Phase 1 of the migration roadmap (Chapter 6) and are the operational foundation for the hybrid TLS work in Chapter 7.

What's Next

With your cryptographic inventory in hand, you now have the data needed to build a prioritized migration plan. Chapter 6 takes the discovery output and transforms it into a phased migration roadmap—complete with a recommended organizational structure (the Cryptographic Center of Excellence), risk-based prioritization, and the crypto-agility principles that ensure you're never locked into a single algorithm again.

Notes

The following sources support specific claims made in Chapter 5. Full bibliographic entries appear in the Bibliography.

1. OMB Memorandum M-23-02 (November 18, 2022) requires FCEB agencies to submit cryptographic inventory reports. The Quantum Computing Cybersecurity Preparedness Act mandates ongoing cryptographic inventories as a statutory obligation. NSM-10 requires annual submissions of quantum-vulnerable IT system inventories.

2. NIST SP 1800-38B (Preliminary Draft), "Migration to Post-Quantum Cryptography: Quantum Readiness—Cryptographic Discovery." Describes a multi-faceted discovery approach including network scanning, endpoint analysis, and application testing. Produced in collaboration with 47+ industry partners including AWS, IBM, Microsoft, Samsung SDS, and others.

3. CISA, "Strategy for Migrating to Automated Post-Quantum Cryptography Discovery and Inventory Tools." September 2024. Notes that only three of the nine M-23-02 data items can currently be collected with automated tools; the remaining six require manual collection or enrichment.

4. The "inventory at source" concept aligns with Gartner's PQC guidance (2026), which recommends that organizations "address inventory at source so that new assets are added to the inventory when implemented, and discovery becomes more relevant to exceptions rather than the primary method for inventory additions."

5. Gartner PQC guidance (2026) recommends "use a well-scoped discovery pilot as proof of value to aid in discovery toolset evaluation" and to "allocate at least 12 months for the cryptographic discovery of critical systems."

6. SafeLogic, "NIST Publishes Next Volume of PQC Migration Guidance" (2025). Notes that automated discovery tools in the NCCoE project routinely revealed hundreds to thousands of previously unknown cryptographic assets across participant environments.

7. F5, Inc. "BIG-IP SSL Orchestrator." Product documentation. SSLO provides high-performance decryption/re-encryption of inbound and outbound TLS traffic with policy-based steering and security service chaining. See: https://www.f5.com/products/big-ip-services/ssl-orchestrator

8. F5, Inc. "Application Study Tool (AST)." Open-source project on GitHub (f5devcentral/application-study-tool). Uses OpenTelemetry Collector with enhanced BIG-IP data receivers, Prometheus, and Grafana for fleet-wide telemetry and visualization.

9. F5, Inc. "F5 Strengthens Its Application Delivery and Security Platform." Press release, March 2026. Announces F5 Insight for ADSP, BIG-IP v21.1 with NIST-compliant PQC cipher support and hybrid TLS cipher groups, and quantum-resistant TLS/SSL VPN tunneling in BIG-IP Zero Trust Access.

Next: Chapter 6 — Building Your Migration Roadmap

CHAPTER 6

Building Your Migration Roadmap

You've catalogued the threat (Chapters 1–2), learned the replacements (Chapter 3), mapped the mandates (Chapter 4), and inventoried your exposure (Chapter 5). Now comes the question that separates planning from action: "How do we actually do this?"

This chapter provides the organizational structure, prioritization framework, and phased migration template you need to turn your cryptographic inventory into a funded, staffed, executable program. This is not theory—it's a migration playbook.

Step 1: Build the Team — The Cryptographic Center of Excellence

PQC migration is not a network project. It's not a security project. It's not a compliance project. It's all of them simultaneously—which means it dies in the gaps between teams unless you create a **cross-functional body with a clear mandate.**

We recommend establishing a **Cryptographic Center of Excellence (CCOE)**—a small, empowered group of strategic thinkers and complex problem solvers drawn from across the organization. The CCOE doesn't replace existing teams. It coordinates them, sets cryptographic policy, and owns the migration roadmap.[1]

Recommended CCOE Composition

Role / Domain	Why They're at the Table
Network Security	Owns TLS termination, IPsec VPNs, load balancer configurations, and the infrastructure where most visible PQC changes happen first.
Application Security	Understands how applications use cryptographic libraries, API authentication, and session management. Identifies application-layer migration dependencies.
Identity & Access Management	Owns PKI, certificate lifecycle, smart card / CAC/PIV authentication, and federation protocols (SAML, OIDC) that depend on digital signatures.
Infrastructure / Endpoint Security	Manages OS-level crypto (disk encryption, code signing verification, secure boot), endpoint configurations, and patch management.
Development / DevSecOps	Controls CI/CD pipelines, application code, cryptographic library selection, and the speed at which software can be updated for new algorithms.
Cryptography SME	Provides algorithm expertise, evaluates implementation correctness, and advises on parameter set selection. May be internal or external consultant.
Risk & Compliance	Maps PQC requirements to regulatory frameworks (FISMA, CMMC, FedRAMP, HIPAA, PCI-DSS). Tracks compliance posture and reports to leadership.
Procurement / Budget	Ensures PQC readiness criteria are embedded in vendor evaluations and acquisition processes. Manages the funding lifecycle for the migration program.

The CCOE should be small (8–12 people), empowered to set cryptographic policy, and have a direct reporting line to the CISO or CTO. Its mandate: ensure consistent, strategic, and measurable PQC migration across the

organization.

> **PLAIN-LANGUAGE SIDEBAR** If your organization doesn't have the internal expertise to staff a cryptography SME role, that's normal. Most don't. The CCOE can work with external consultants, your security vendor's engineering team, or industry bodies like the PKI Consortium for specialized guidance. The important thing is that the cross-functional structure exists—the expertise can be sourced.

Step 2: Design for Crypto-Agility

Before you start replacing algorithms, establish a design principle that will save you from doing this again in a decade: **crypto-agility.**

NIST defines crypto-agility as "the capabilities needed to replace and adapt cryptographic algorithms in protocols, applications, software, hardware, and infrastructures without interrupting the flow of a running system."[2] In December 2025, NIST published CSWP 39 ("Considerations for Achieving Crypto Agility: Strategies and Practices"), elevating crypto-agility from a nice-to-have to a formal strategic framework.

The core idea: **never hard-code cryptographic choices again.** The PQC migration should be the last time your organization does a panic-driven, multi-year cryptographic overhaul. After this transition, your architecture should support algorithm swaps as routine maintenance, not emergency projects.

Four Pillars of Crypto-Agility

- **Modularity:** Separate cryptographic algorithms from application logic. Use cryptographic APIs that abstract the algorithm choice, so the same API call can invoke classical or PQC algorithms without application changes.
- **Policy-driven configuration:** Cryptographic choices should be set by external policy—machine-readable configuration profiles—not compiled into software. When an algorithm is deprecated, an administrator updates a policy; developers don't rewrite code.[3]
- **Inventory and monitoring:** You can't be agile about what you can't see. The CBOM from Chapter 5 isn't a one-time deliverable—it's a living system that continuously tracks what cryptography is deployed where.
- **Testing and validation:** Build automated tests that verify cryptographic configurations against policy. When a new algorithm is approved or an old one deprecated, the test suite catches drift before auditors do.

CSWP 39 introduces a maturity model for crypto-agility, ranging from unstructured and reactive practices at the low end to fully adaptive programs integrated into enterprise risk management.[2] The PQC migration is your opportunity to climb that maturity curve. Don't just solve the quantum problem—build the capability to solve the next cryptographic problem, whatever it is.

Crypto-Agility in Five Layers

The four pillars above describe what crypto-agility looks like as a design discipline. But "be crypto-agile" is the kind of advice that's easy to nod at and hard to operationalize. What does it actually mean to build agility into an organization's cryptographic stack?

Gartner's PQC journey guide offers one of the cleanest decompositions we've found: crypto-agility shows up at five distinct layers, and each requires its own decisions, its own tools, and its own owners. Treating agility as a single objective tends to make it everyone's job and therefore nobody's. Treating it as five layers — each with concrete deliverables — makes it tractable.[7]

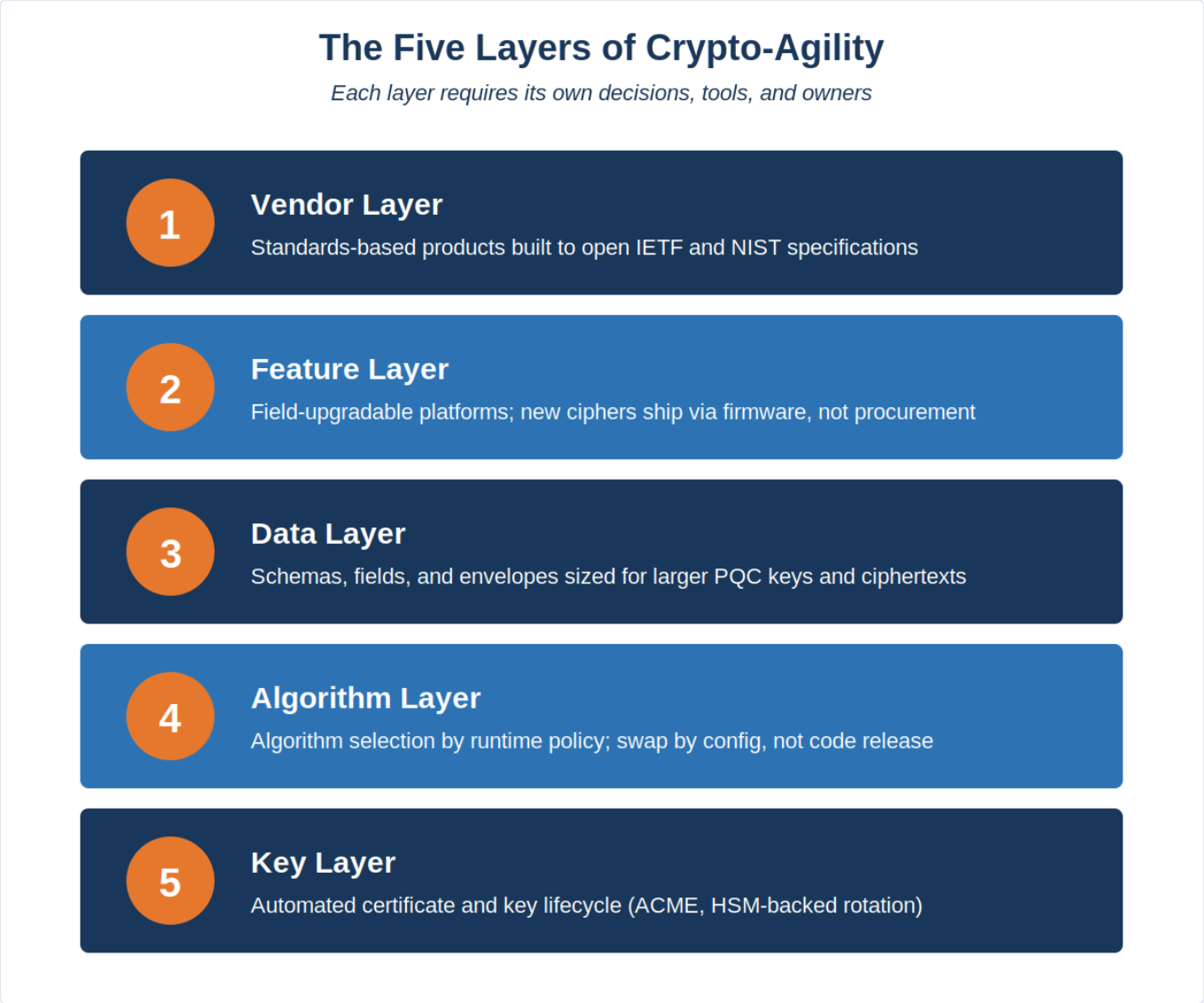

Figure 6.2 — The Five Layers of Crypto-Agility

Vendor layer. The vendor layer is about the products and platforms an organization brings in to host its cryptography. Agility here means buying from vendors who follow open standards rather than proprietary protocols — a standards-based product can interoperate with the rest of the ecosystem and can adopt new IETF or NIST algorithms as they ship; a proprietary one becomes a hostage. For PQC specifically, the minimum bar is FIPS-approved algorithms (ML-KEM, ML-DSA, SLH-DSA) and IETF-defined protocol profiles for TLS 1.3, IKEv2, and X.509. Any vendor selling "quantum-safe" cryptography that doesn't land cleanly in one of those buckets deserves a hard second look.

Feature layer. The feature layer is about whether the products already in production can be upgraded without being ripped out and replaced. Field-upgradable hardware and software is the difference between a cipher

migration that takes a configuration change and one that takes a procurement cycle. Ask every cryptographic-touching vendor what their PQC delivery mechanism looks like, and treat “wait for the next major version” as a yellow flag.

Data layer. The data layer is where things get unglamorous and important. Larger keys and ciphertexts mean larger fields in databases, longer columns in audit logs, larger signed envelopes in document formats, and bigger entries in directory schemas. If those schemas weren’t sized for it, PQC migration breaks them in subtle ways — truncated signatures that validate locally and fail downstream, or queries that silently drop rows when the certificate field overflows. Building data-layer agility means auditing every place a cryptographic artifact is stored or transmitted and making sure it can carry, at minimum, an ML-DSA-87 signature (approximately 4,627 bytes) or an ML-KEM-1024 ciphertext (approximately 1,568 bytes) with headroom for future algorithm growth.

Algorithm layer. The algorithm layer is the cleanest agility win and the easiest one to design in. The principle is straightforward: algorithm selection should be a runtime policy decision, not a compiled-in constant. A TLS terminator should accept a cipher group as configuration. A signing service should accept an algorithm identifier as a parameter. A KMS should let policy decide whether new keys are ML-DSA-65 or ML-DSA-87. When agility is configured rather than coded, swapping algorithms in response to a new cryptanalytic advisory becomes a change ticket, not a software release. NIST CSWP 39 formalizes this as “modularity,” and it is the single most important agility property a system can have.

Key layer. The key layer is about the operational machinery that issues, rotates, and retires keys and certificates. None of the layers above matter if cert rotation takes three weeks and a change-advisory-board approval. Agility at this layer is automation: ACME for certificate lifecycle, KMS-managed key rotation policies, HSM-backed signing operations that don’t require staring at a terminal. The CA/Browser Forum 47-day maximum certificate validity in March 2029 alone makes automated certificate lifecycle management non-optional; organizations that haven’t automated certificate operations by the end of 2027 will be trying to do that and PQC algorithm change at the same time.

Strong crypto-agility is the product of all five layers, not any one of them. A standards-based vendor running upgradable platforms that handle larger artifacts via policy-driven algorithm selection and automated certificate operations — that’s an organization that can absorb its next algorithm migration as routine work, not crisis response. The PQC migration is the first test of these layers for most organizations. It will not be the last.

F5 PERSPECTIVE **BIG-IP Maps to All Five Layers** The five layers map cleanly onto how the BIG-IP platform is architected. **Vendor layer.** ML-KEM and ML-DSA arrive through FIPS-validated OpenSSL modules and IETF-defined TLS 1.3 profiles — no proprietary cipher extensions, no parallel "F5-flavored" PQC. What you negotiate on the wire is what the standards bodies published. **Feature layer.** Customers move from classical to hybrid PQC via firmware upgrade. v17.5.1 introduced X25519MLKEM768 hybrid key exchange; v21.1 expanded the supported PQC cipher set. No platform refresh, no new licensing, no procurement cycle. **Data layer.** The certificate chain math in Chapter 8 (ML-DSA-65 chains running 15–20 KB on the wire) translates into concrete BIG-IP capacity planning. Memory per connection scales, SSL profile sizing matters more than it did with ECDSA, and TLS Certificate Compression (RFC 8879) becomes a profile setting worth turning on by default. **Algorithm layer.** The SSL cipher group is a configuration object, not a code path. Fleet-wide algorithm changes ship via iControl REST, AS3 declarations, or BIG-IQ — an algorithm rollback in response to a future cryptanalytic advisory is a config push, not a software release. **Key layer.** BIG-IP v21.1 added native ACME v2 client support on the TLS terminator, eliminating the need for external Certificate Lifecycle Management glue for automated renewal. Combined with the CA/Browser Forum 47-day validity deadline in March 2029, this is the layer where v21.1 carries the most operational weight.

⚠ **MANDATE ALERT** **Eight Things Your Crypto Policy Must Now Address** Updating internal cryptographic policy is one of the cheapest deliverables of a postquantum program — and one of the highest-leverage. Every architecture decision, vendor procurement, and engineering pattern that follows will reference the policy. At minimum, it should address: **Deprecation timelines** for RSA, DH, ECC, and other quantum-vulnerable algorithms, aligned to NIST IR 8547 (deprecated after 2030, disallowed after 2035) unless sector mandates pull dates earlier. **PQC adoption timelines and conditions,** including which algorithms are approved, at which parameter sets, and whether hybrid deployment is required during the transition window. **Minimum parameter sets per security category** — FIPS Category 3 (ML-KEM-768 / ML-DSA-65) as the default; Category 5 (ML-KEM-1024 / ML-DSA-87) for long-lived signing keys and 25-plus-year data. **Hybrid versus pure PQC strategy** — where each pattern applies, and when the transition from hybrid to pure PQC is scheduled. **Position on QKD and adjacent techniques** (photonic-layer protection, fully homomorphic encryption, multiparty computation). The defensible default: not for general-purpose cryptography; revisit if NSA/NIST guidance changes. **Vendor readiness deadlines** with explicit consequences for vendors that miss them (re-procurement, exception process, contract triggers). **Cryptographic inventory mandate** — new systems register their cryptographic dependencies in the CBOM at deployment, not after. **Crypto-agility requirements** — algorithm selection as configuration, not hard-coded constants. NIST CSWP 39 supplies the reference framework.

Step 3: Prioritize Based on Risk, Not Compliance Dates

With your CBOM in hand and your CCOE assembled, the next question is: what do we migrate first?

The temptation is to prioritize by compliance deadline—"CNSA 2.0 says networking by 2026, so we start there." That's not wrong, but it's incomplete. The better framework prioritizes by **risk exposure**, which accounts for both compliance and the actual damage a quantum adversary could inflict:

Priority	Criteria	Examples	Action
P0	HNDL-exposed + long-lived data + internet-facing	TLS key exchange on internet-facing services protecting classified, financial, or medical data	**Migrate now. Deploy hybrid key exchange immediately.**

Priority	Criteria	Examples	Action
P1	HNDL-exposed + moderate data lifetime + internet-facing	General web TLS, VPN tunnels, partner-to-partner links, email encryption	Begin migration within 6 months. Hybrid mode.
P2	Authentication and signatures on long-lived artifacts	Code signing, firmware signing, CA certificates, legal/compliance evidence	Plan migration within 12 months. CNSA 2.0 firmware signing is urgent.
P3	Internal infrastructure + moderate data sensitivity	Internal TLS between microservices, east-west traffic, internal PKI, SSH keys	Migrate within NIST timeline. Plan for 2028–2032 execution.
P4	Low-sensitivity + short-lived data + low interception risk	Ephemeral session tokens, internal API keys, transient development environments	Upgrade during normal refresh cycles. Low urgency.

Notice that key exchange (confidentiality) consistently outranks signatures (authentication) in priority. That's because the HNDL threat applies to key exchange today—captured traffic can be decrypted retroactively. Signatures, by contrast, only need to be quantum-resistant at the time they're verified. This is why Chrome, NIST, and most migration frameworks recommend **key exchange first, signatures second.**[4]

Step 4: Assess HSM and Infrastructure Readiness

Before you can execute the migration, you need to know whether your infrastructure can actually support the new algorithms. The most common blocker is **HSM readiness**—Hardware Security Modules sit at the root of trust for most certificate and signing operations, and if your HSM can't handle PQC algorithms, your PKI migration stalls before it starts.[5]

Five HSM Planning Questions

- **Algorithm support:** Does the HSM firmware expose the PQC algorithms you need (ML-KEM, ML-DSA, LMS/XMSS)? Which parameter sets are available?
- **API and connector mapping:** How do PQC key types map into PKCS#11, vendor APIs, and the application connectors your CA, signing platform, and identity systems use?
- **Backup and HA semantics:** Do new PQC key types change how cloning, backup, restore, or active-active designs behave? Stateful hash-based signatures (LMS/XMSS) have unique state management requirements.
- **Performance and object size:** Larger PQC keys and signatures change throughput, storage, CSR processing, and certificate issuance rates. Benchmark before committing to a production timeline.
- **Validation and compliance timing:** Algorithm availability in a product is not the same as FIPS 140-3 validation. Verify the exact firmware version, validation state, and expected certification timeline for your environment.

HSM vendor readiness is improving rapidly but unevenly. Thales Luna HSMs support ML-DSA through firmware 7.9.0+. Entrust nShield 5 has NIST CAVP-validated support for ML-DSA, ML-KEM, and SLH-DSA with FIPS 140-3 certification work following.[6] The planning lesson: verify the exact firmware, SDK, connector, and validation path before committing a PKI or signing architecture to a date.

Step 5: Execute in Phases

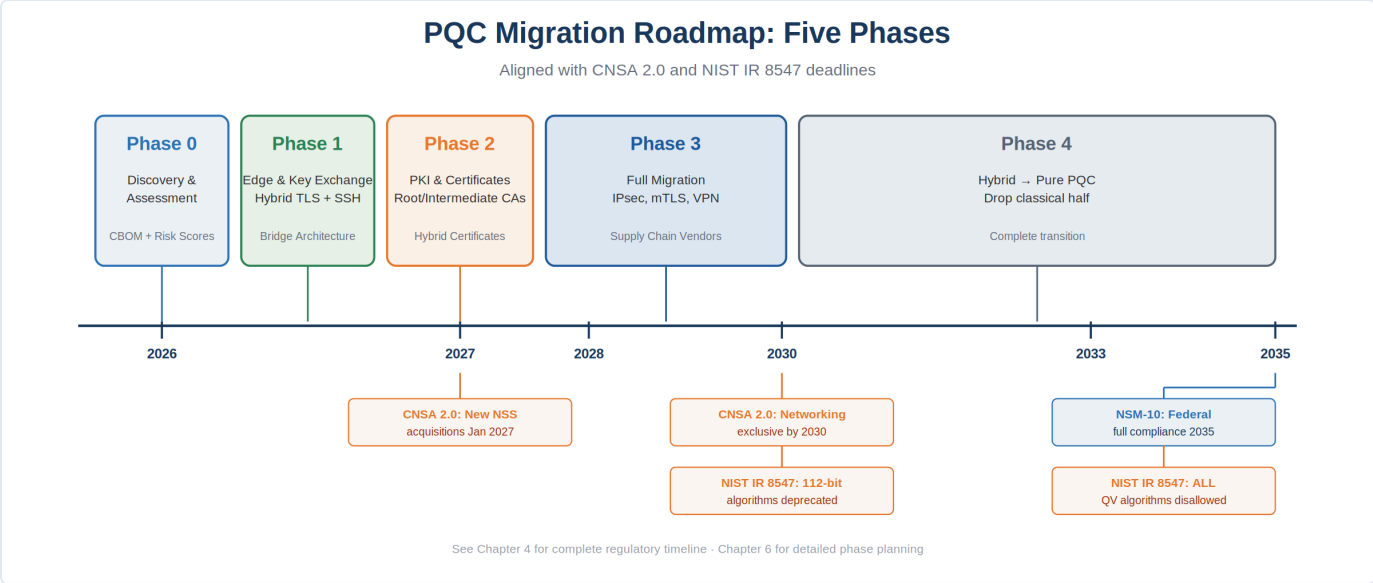

Figure 6.1 — PQC Migration Roadmap: Five Phases (2026–2035)

The most credible migration plan is phased: edge first, origins next, trust infrastructure after that, and supply chain signing alongside. Here's the template:

Phase	Timeline	Key Actions
Phase 0: Organize	Months 1–3	Establish CCOE. Complete CBOM for P0 systems. Assess HSM readiness. Define crypto-agility policy. Set budget and executive sponsorship.
Phase 1: Edge First	Months 3–9	Deploy hybrid key exchange (ML-KEM + X25519) on internet-facing TLS termination points. Pilot 2–3 services. Measure client compatibility, performance, and operational impact. Expand to all P0 and P1 services.
Phase 2: Trust Infrastructure	Months 6–18	Migrate code signing and firmware signing to PQC (CNSA 2.0 priority). Begin PKI migration: issue PQC certificates from internal CAs. Update HSMs. Test hybrid certificate chains. Migrate VPN/IPsec key exchange.
Phase 3: Broaden	Months 12–30	Migrate P2 and P3 systems. Expand PQC to internal TLS, SSH, email encryption, and database encryption. Replace quantum-vulnerable certificates across the enterprise. Address partner and supply chain dependencies.
Phase 4: Complete and Sustain	Months 24–36+	Retire all quantum-vulnerable algorithms. Transition from hybrid to pure PQC where appropriate. Embed crypto-agility into ongoing operations (Chapter 9). Decommission legacy cryptographic configurations. Validate compliance posture against NIST 2035 deadline.

> ⚠ **MANDATE ALERT** These timelines are templates, not mandates. Your actual pace depends on your regulatory environment, infrastructure complexity, and risk tolerance. For CNSA 2.0 organizations, Phase 1 should already be in progress. For commercial organizations following the NIST timeline, the UK NCSC's Phase 1 target (crypto inventory and plan complete by 2028) is a reasonable benchmark.

F5 PERSPECTIVE **Edge-First Migration with F5 BIG-IP** Phase 1 of the migration roadmap—hardening internet-facing TLS key exchange—maps directly to F5's deployment model. BIG-IP already sits at the TLS offload point for most F5 customers, making it the fastest place to prove PQC progress. The operational pattern: enable hybrid TLS 1.3 (X25519MLKEM768) on the browser-facing side of BIG-IP while preserving classical TLS on the backend origin connections. This hardens the internet-exposed leg against HNDL risk immediately, without requiring every origin server, application framework, or backend library to be upgraded first. This "bridge architecture" is often the right operational choice, but it should be described honestly: it improves the front door first. It does not make the full application path post-quantum safe if the backend is still classical. The program still has to expand into PKI, certificates, HSMs, code signing, device identity, and supply chain trust. BIG-IP is the starting point, not the whole program. Chapter 7 covers the hybrid deployment patterns in detail.

Selling the Roadmap: Budget and Executive Sponsorship

A migration roadmap without budget is a wish list. Here's how to frame the investment for leadership:

- **Frame it as risk reduction, not compliance:** "We're reducing the window during which captured data can be retroactively decrypted" is a more compelling message than "NIST says we have to."
- **Quantify the crypto debt cleanup:** Your discovery pilot (Chapter 5) found SHA-1 in production, expired certificates, and TLS 1.0 connections. Those are security risks today, not just quantum risks. The PQC budget fixes both.
- **Use the SHA-1 precedent:** The SHA-1 to SHA-2 migration took 12+ years and cost organizations billions in aggregate. The PQC migration is larger and more complex. Early investment reduces total cost.
- **Show the regulatory trajectory:** The Master Timeline from Chapter 4 demonstrates that deadlines are converging across NIST, CNSA 2.0, UK NCSC, and EU NIS2. This isn't one agency's opinion—it's global consensus.
- **Start small, prove value:** Phase 0 and the early Phase 1 pilot can be funded from existing security budgets. The pilot results (discovery findings + performance benchmarks + client compatibility data) justify the larger Phase 2–4 investment.

PLAIN-LANGUAGE SIDEBAR **What Does PQC Migration Actually Cost?** The U.S. federal government estimated total government-wide PQC migration costs at $7.1 billion between 2025 and 2035, and requires agencies to update cost estimates annually. While your organization won't spend billions, the cost categories are the same: **Personnel and expertise** — staff time for discovery, planning, testing, and execution. This is typically the largest cost. Expect 2–5 FTEs dedicated to the CCOE for 3–5 years in a mid-size enterprise. **Hardware replacement** — HSMs, network appliances, and embedded devices that cannot be firmware-upgraded to support PQC may need replacement. This is the wild card—cost varies dramatically by environment. **Software and tooling** — crypto discovery tools, certificate lifecycle management platforms, testing infrastructure, and vendor upgrades. **Potential downtime and rollback** — budget for complications. The AIVD/TNO PQC Migration Handbook emphasizes that unforeseen issues during execution are common and a robust rollback procedure is essential. The PQC migration is also an opportunity: the crypto debt cleanup (finding SHA-1, expired certs, TLS 1.0) delivers security value today, not just in a post-quantum future. Frame the investment as infrastructure modernization, not just compliance.

Building the Financial Case

The arguments above frame the pitch. Securing the actual funding is a separate exercise — one that depends less on the strength of the pitch and more on how the program integrates with the organization's financial planning cycle. Funding for a multi-year cryptographic program is not a line item your finance team has seen before; the financial pattern has to be built from scratch.[7]

A useful working frame: the postquantum program is a multi-year capital and operating investment that competes with every other multi-year initiative in the organization. Treat finance as a partner, not a gatekeeper. The earlier they understand the cryptographic risk and the migration shape, the more likely they are to defend the budget when other priorities crowd it.

Partner with finance early

Bring the CFO's office into the conversation during the planning phase, not at the budget-cycle deadline. Walk them through the HNDL risk in plain language. Walk them through the multi-year deliverable structure. Walk them through the cost of doing nothing — not as a scare tactic, but as the alternative scenario any sound investment decision has to compare against. The goal is shared ownership of the program's financial trajectory, not a one-shot funding ask.

Plan for reforecasting, not a single budget request

Multi-year cryptographic migrations rarely land on their first estimate. New NIST guidance, vendor delivery slips, discovered scope, and cryptanalytic news will all shift the plan. The financial model should be built to accommodate quarterly or semi-annual reforecasts, with the program lead and a finance partner running the cycle jointly. Organizations that pitch one large number up front and then return for "additional" funding when the scope shifts erode credibility fast. Organizations that pitch a phased model with explicit reforecast checkpoints build it.

Frame costs in both tangible and intangible categories

Tangible costs are the easy ones: staff time, tooling licenses, professional services, HSM upgrades, lab and test infrastructure, ongoing certificate operations. These are the line items finance will recognize. Intangible costs need surfacing too: migration-related downtime, opportunity cost of engineering attention pulled from other work, the regression risk of running classical and hybrid in parallel for months. Counting only the tangible costs produces a number that's too low, and the program runs out of runway in year two.

Mirror the budget to the roadmap

The first funded deliverable should be a scoping exercise — discovery pilot, vendor assessment, initial CBOM — that produces the data needed to size the rest. This is also the easiest funding ask, because the deliverable is small, the timeline is short, and the output is a defensible plan. Subsequent phases — edge hybrid deployment, internal mTLS, signing infrastructure, legacy bridge architecture — each carry their own scope and cost estimate. The roadmap and the budget become the same document.

Invest in the right mix of platform and custom tooling

A budget that funds only commercial CLM, discovery, and TLS platforms misses the integration work that holds them together. A budget that funds only custom-built tooling misses the maturity and support of vendor platforms. The right ratio depends on the organization's engineering capacity and risk tolerance, but the question itself deserves explicit attention in the financial model. "Buy what's mature; build what's specific" is a defensible starting principle.

Tie metrics to dollars where possible

Finance partners respond to numbers. Where the program can quantify progress — percent of internet-facing endpoints on hybrid TLS, percent of P0 systems migrated, mean time to deploy a new cipher policy — those numbers should feed the financial reporting. They demonstrate execution, justify continued funding, and give finance a defensible answer when other budget owners ask why cryptographic migration is consuming this much capital this many years in a row.

What's Next

You have the team (CCOE), the design principle (crypto-agility), the priority matrix (risk-based), the infrastructure assessment (HSM readiness), and the phased plan (edge first). The next chapter dives into the specific technical pattern that dominates Phase 1: hybrid mode—running classical and post-quantum cryptography side by side during the transition. Chapter 7 covers hybrid TLS, hybrid certificates, and the bridge architecture in detail.

Notes

The following sources support specific claims made in Chapter 6. Full bibliographic entries appear in the Bibliography.

1. The Cryptographic Center of Excellence (CCOE) concept aligns with Gartner's 2026 PQC guidance, which recommends creating a "cross-functional cryptographic center of excellence with clear mandates to ensure con-

sistent and strategic quantum threat remediation across the organization." The recommended composition draws from Gartner's suggested domains: data security, network security, infrastructure, endpoint, application security, development, IAM, cryptography, risk/compliance, procurement, and budgeting.

2. NIST CSWP 39 (Final), "Considerations for Achieving Crypto Agility: Strategies and Practices." Published December 2025. Defines crypto-agility, proposes a maturity model, and describes technical levers including modularity, policy-mechanism separation, inventory, and testing.

3. CSWP 39 describes "technology-specific, machine-consumable configuration profiles" as a mechanism for enforcing cryptographic policy across systems without hard-coding algorithm choices into software.

4. Chrome's PQC strategy explicitly prioritizes key exchange over authentication due to the HNDL risk asymmetry. See: Chromium Blog, "Advancing Our Amazing Bet on Asymmetric Cryptography," May 2024. NIST IR 8547 similarly notes that application-specific guidance may require earlier migration for key establishment to mitigate HNDL.

5. HSM readiness planning framework adapted from F5, Inc. internal PQC field guidance (2025). The five planning questions address the operational dimensions most frequently encountered in customer PQC migration discussions.

6. Thales documents ML-DSA support through Luna HSM Firmware 7.9.0+, with operational caveats for stateful hash-based signatures (LMS-HSS) including backup and HA limitations. Entrust announced NIST CAVP-validated support for ML-DSA, ML-KEM, and SLH-DSA in nShield 5 firmware in 2025, with FIPS 140-3 certification work following. Source: vendor product documentation.

7. Sarah Almond and Mark Horvath, "A Journey Guide to Postquantum Readiness," Gartner Research G00843746, 13 March 2026. The five-layer crypto-agility decomposition (vendor, feature, data, algorithm, key) in Step 2 is paraphrased from this source, as is the financial planning guidance in "Building the Financial Case." The Mandate Alert's eight policy requirements also draw on this source.

Next: Chapter 7 — Hybrid Mode: Bridging Classical and Quantum-Safe

CHAPTER 7

Hybrid Mode: Bridging Classical and Quantum-Safe

In an ideal world, you'd flip a switch and every system in your environment would instantly use post-quantum algorithms. In the real world, migration happens gradually—and during that transition, classical and post-quantum cryptography need to coexist. That coexistence is called **hybrid mode**, and it's the dominant deployment pattern for PQC today.

This chapter explains what hybrid mode is, why it matters, how it works across TLS, IPsec, and SSH, and where the approach has limitations. It also addresses a question you'll inevitably encounter: "What about Quantum Key Distribution?"

Why Hybrid? The Belt-and-Suspenders Argument

The post-quantum algorithms in FIPS 203, 204, and 205 have been rigorously evaluated through an eight-year international competition. But they are younger than the classical algorithms they replace. RSA has been scrutinized for over 40 years. ML-KEM has been scrutinized for roughly 8. The cryptographic community has high confidence in the new algorithms—but not 40 years of confidence.

Hybrid mode provides a hedge: **combine a classical algorithm with a PQC algorithm so that the system remains secure as long as at least one of them holds.** If ML-KEM is someday broken by a novel attack, the classical X25519 component still protects the session. If a quantum computer arrives and breaks X25519, the ML-KEM component protects it. You need to break both to compromise the connection.[1]

> **PLAIN-LANGUAGE SIDEBAR** Think of hybrid mode like a deadbolt paired with a smart lock on your front door. If someone picks the deadbolt, the smart lock still holds. If someone hacks the smart lock, the deadbolt still holds. An attacker has to defeat both to get in. That's the security guarantee of hybrid cryptography.

NIST IR 8547 explicitly supports hybrid implementations during the transition period.[2] ENISA recommends hybrid schemes for EU organizations. The UK NCSC endorses hybrid key exchange. And the real-world deployment numbers speak for themselves: as of September 2025, approximately 43% of human-generated HTTPS connections to Cloudflare used hybrid PQC key exchange.[3]

Hybrid TLS: Already in Your Browser

If you're reading this chapter in Chrome, Edge, Brave, or another Chromium-based browser, there's a good chance your current connection is already using hybrid PQC key exchange—and you didn't do a thing to enable it.

The dominant hybrid TLS key exchange is **X25519MLKEM768**, which combines the classical X25519 elliptic curve key agreement with ML-KEM-768 post-quantum key encapsulation. The IETF has formalized this in

draft-ietf-tls-ecdhe-mlkem, specifying three hybrid groups:[4]

Hybrid Group	Components	Client Key Share Size	Security Level
X25519MLKEM768	X25519 + ML-KEM-768	1,216 bytes	Level 3 (AES-192)
SecP256r1MLKEM768	P-256 + ML-KEM-768	1,249 bytes	Level 3 (FIPS)
SecP384r1MLKEM1024	P-384 + ML-KEM-1024	1,665 bytes	Level 5 (AES-256)
X25519 alone (classical)	X25519 only	32 bytes	— (broken by Shor's)

The overhead is modest: X25519MLKEM768 adds approximately 1.1 KB to the client's key share compared to classical X25519. In practice, this adds only 1–2 milliseconds to the TLS handshake—imperceptible to users. Multiple studies and production deployments have confirmed that the performance impact of hybrid key exchange is negligible on modern networks.[5]

Who's Already Deployed

- **Google Chrome:** Enabled X25519MLKEM768 by default for TLS 1.3 connections. Previously used the pre-standard X25519Kyber768Draft00, now migrating to the final standard.
- **Cloudflare:** 43% of human HTTPS traffic using hybrid PQC as of September 2025. Scanning origins to enable hybrid edge-to-origin connections automatically.[3]
- **AWS:** Hybrid TLS support in s2n-tls and AWS services. Contributors to the IETF hybrid TLS draft.
- **Apple:** Secured iMessage with PQ3 protocol (PQC key exchange) since iOS 17.4. Safari hybrid TLS in progress.
- **Signal:** Integrated X25519 + Kyber hybrid key exchange into the Signal Protocol.

The point: hybrid PQC isn't experimental. It's production-grade infrastructure that billions of connections use daily. The question for your organization isn't whether hybrid works—it's when you'll enable it on your own infrastructure.

The Bridge Architecture: Front-Side PQC, Back-Side Classical

For most enterprises, the fastest path to PQC progress is not upgrading every application, library, and origin server simultaneously. It's upgrading the edge—the TLS termination point where internet traffic enters your environment.

The **bridge architecture** works like this: your TLS termination device (load balancer, reverse proxy, ADC) negotiates hybrid TLS 1.3 with modern clients on the front side while maintaining classical TLS 1.2 or TLS 1.3 connections to backend origins. The internet-exposed leg is PQC-hardened against HNDL; the internal leg remains classical until origins are upgraded. In federal and enterprise architecture vocabulary, this pattern is also called a **cryptographic proxy layer**—a dedicated enforcement point that performs cryptographic upgrade on behalf of downstream systems that aren't yet PQC-capable. Both terms describe the same architecture; "bridge" emphasizes the temporal aspect (carrying systems across the migration), while "cryptographic proxy layer" emphasizes the structural aspect (a distinct layer in the trust architecture).[6]

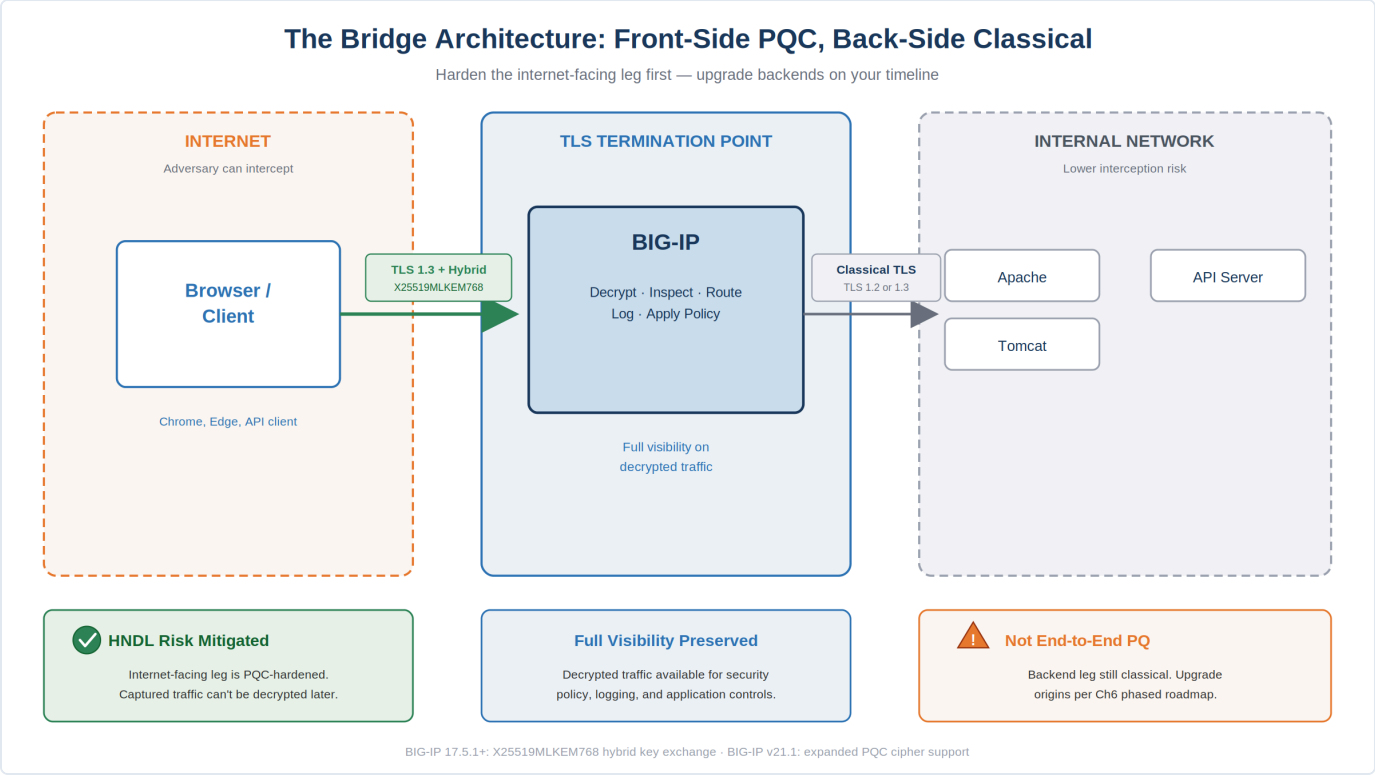

Figure 7.1 — The Bridge Architecture: Front-Side PQC, Back-Side Classical

Leg	Protocol State	What It Means
Browser → Edge Device	TLS 1.3 + hybrid key exchange (X25519MLKEM768)	Front door is PQC-hardened. HNDL risk mitigated for the internet-facing leg.
Inside Edge Device	Traffic terminated and decrypted	Policy enforcement, inspection, logging, and routing continue unchanged.
Edge Device → Origin	Classical TLS 1.2 or TLS 1.3 (separate session)	Backend is NOT post-quantum hardened. Useful bridge, not end-to-end PQ.

F5 PERSPECTIVE **BIG-IP as the PQC Bridge** This bridge architecture maps directly to BIG-IP's deployment model. BIG-IP 17.5.1 added support for X25519MLKEM768 hybrid key exchange in TLS 1.3 on both client and server sides. BIG-IP v21.1 expands PQC cipher support with hybrid TLS cipher groups and adds quantum-resistant TLS/SSL VPN tunneling through BIG-IP Zero Trust Access. The operational value: a small number of internet-facing VIPs can be upgraded to hybrid TLS before hundreds of origin servers are touched. Application teams gain runway to upgrade Apache, Java, OpenSSL, PKI workflows, and HSM dependencies in a controlled sequence. BIG-IP preserves its existing visibility—decrypted traffic remains available for routing, security policy, logging, and application controls. **An honest assessment:** this architecture hardens the front door first. It does not make the full application path post-quantum safe if the backend is still classical. The backend leg, while less exposed to external interception, still requires migration on the timeline established in Chapter 6. BIG-IP is the starting point, not the whole program.

Hybrid IPsec: Pre-Shared Keys as a Stopgap

IPsec environments—particularly in DoD and federal networks—face a different hybrid challenge. Full PQC integration into IKEv2 is still maturing through IETF drafts, and many VPN devices don't yet support ML-KEM in their key exchange.

The interim solution is **Post-Quantum Pre-Shared Keys (PPKs)**, specified in RFC 8784. PPKs add a quantum-resistant pre-shared secret to the IKEv2 key derivation process—effectively layering a symmetric (quantum-safe) secret on top of the existing classical key exchange. Even if an adversary captures the IKE handshake and later breaks the DH/ECDH component with a quantum computer, the PPK ensures the derived session keys are still protected.[7]

PPKs are a bridge, not a destination. The long-term solution is native ML-KEM integration in IKEv2, which the IETF and NSA are actively developing through the CNSA 2.0 IPsec profile.[8] But for organizations that need quantum-resistant VPN tunnels today, PPKs are the fastest path available and are already supported by multiple VPN vendors.

Hybrid SSH: Already the Default

SSH may be the simplest hybrid success story. OpenSSH introduced hybrid post-quantum key exchange earlier than most protocols, and it's now the default.

- **OpenSSH 9.0 (April 2022):** Introduced sntrup761x25519-sha512 as the default key exchange. This combined a lattice-based algorithm (NTRU Prime) with X25519.
- **OpenSSH 10.0 (April 2025):** Switched the default to mlkem768x25519-sha256, aligning with NIST's ML-KEM standard.[9]

If your servers are running a current OpenSSH version, your SSH key exchanges are already quantum-safe in hybrid mode. Authentication (host and user keys) still uses classical algorithms and will need migration to PQC signatures—but the key exchange leg is handled.

Hybrid Certificates: The Harder Problem

Hybrid key exchange is solved. Hybrid authentication—specifically, hybrid certificates—is significantly harder.

A **hybrid certificate** carries two public keys and two signatures: one classical (e.g., ECDSA) and one post-quantum (e.g., ML-DSA). Old clients that don't understand PQC can validate the classical signature and ignore the PQC component. New clients can validate both. This ensures backward compatibility during the transition.

The challenge: hybrid certificates are even larger than pure PQC certificates. A certificate with both an ECDSA and ML-DSA-65 signature carries the overhead of both—roughly 5+ KB for a single certificate, compared to ~1 KB for today's ECDSA certificates. In a three-certificate chain, the overhead becomes significant and can trigger TLS handshake fragmentation. Chapter 8 covers this in detail.[10]

For now, the industry approach is: deploy hybrid key exchange first (it's the HNDL priority) and defer hybrid certificate migration until the size and interoperability challenges are resolved. Chrome, NIST, and the IETF

are all aligned on this sequencing.

What About Quantum Key Distribution?

In customer conversations about PQC, someone will inevitably ask: "Why don't we just use Quantum Key Distribution instead?" It's a fair question. QKD uses the physics of quantum mechanics—not mathematical trapdoors—to distribute encryption keys, and its security is theoretically guaranteed by the laws of nature rather than computational hardness. That sounds like the ultimate solution.

It's not. At least not for most organizations. Here's why.

QKD requires dedicated physical infrastructure —typically fiber optic links or satellite channels—between every pair of communicating parties. It cannot operate over the existing internet. It cannot protect a connection between a mobile device and a web server. It cannot secure email. It doesn't scale to the millions of connections per second that modern applications demand.[11]

QKD provides key distribution only. It does not authenticate. It cannot verify that the other party is who they claim to be. To use QKD securely, you still need classical or post-quantum digital signatures for authentication—which means you need PQC anyway.[12]

QKD has limited range. Current fiber-based QKD systems work over distances of roughly 100–200 km before requiring "trusted nodes" (relay points that must be physically secured). Quantum repeaters that could extend the range are still in early research stages.

Major security agencies recommend against QKD for most use cases. The NSA does not support QKD for protecting National Security Systems. The UK NCSC states: "PQC is the best mitigation to the threat to cryptography from quantum computers" and will not endorse QKD for government or military applications. ANSSI (France), BSI (Germany), and NLNCSA (Netherlands) have all taken similar positions.[13] The November 18, 2025 DoW CIO memorandum *Preparing for Migration to Post Quantum Cryptography* made this position binding for the Department of War: DoW Components are prohibited from testing, evaluating, piloting, using, or procuring QKD—or any solution combining QKD with other cryptographic key establishment—for confidentiality, authenticity, integrity, key distribution, or randomness generation, absent specific exception from the DoW CIO PQC Directorate. See Chapter 4 for the full memo treatment.[14]

> **PLAIN-LANGUAGE SIDEBAR** QKD is a real technology with genuine strengths, and it has a role in specialized, high-security, point-to-point links where dedicated fiber infrastructure already exists and the cost is justified. China has deployed a working QKD network of approximately 5,000 km and a QKD satellite. The EU's EuroQCI initiative is building an EU-wide QKD network. These are significant investments. But for the vast majority of organizations—protecting web applications, VPN tunnels, API traffic, email, and cloud workloads—PQC is the answer. QKD and PQC can complement each other in specialized environments, but PQC is the practical, standards-based, deployable-today solution for enterprise cryptographic migration.

From Hybrid to Pure PQC: When to Drop the Classical Half

Hybrid mode is a bridge, not a destination. Eventually—as confidence in PQC algorithms matures and classical algorithms are disallowed by NIST—organizations will transition from hybrid to pure PQC. The timing depends on your risk posture:

- **2026–2030:** Deploy hybrid everywhere. This is the belt-and-suspenders phase. Classical algorithms are still permitted; PQC algorithms are still accumulating cryptanalytic scrutiny. Hybrid gives you quantum safety without betting everything on the new math.
- **2030–2033:** Begin transitioning high-confidence systems to pure PQC. By this point, ML-KEM and ML-DSA will have had 6+ years of post-standardization scrutiny. NIST will have deprecated classical algorithms at the 112-bit level. New systems should default to PQC-only.
- **2033–2035:** Complete the transition. NIST disallows all quantum-vulnerable algorithms. Hybrid mode becomes unnecessary because the classical component no longer adds value—and may introduce unnecessary complexity and bandwidth overhead.

The crypto-agility principles from Chapter 6 ensure you can make this transition smoothly: if your architecture is modular and policy-driven, switching from hybrid to pure PQC is a configuration change, not a redesign.

What's Next

This chapter covered the deployment patterns for running classical and PQC side by side. But how do these patterns actually play out at the protocol level? What happens to TLS handshake sizes when PQC certificates enter the picture? How does IPsec IKEv2 change when ML-KEM replaces DH? What does a PQC SSH session look like on the wire?

Chapter 8 takes you inside each protocol—TLS, IPsec, SSH, and PKI—with the technical detail your engineering teams need to plan and execute the migration.

Notes

The following sources support specific claims made in Chapter 7. Full bibliographic entries appear in the Bibliography.

1. The hybrid security guarantee—secure as long as at least one component algorithm holds—is formalized in IETF draft-ietf-tls-hybrid-design. NIST IR 8547 (Section 4.1) describes hybrid approaches as combining "a classical algorithm and a PQC algorithm into a composite mechanism, intended to be secure as long as at least one of the component algorithms is secure."

2. NIST IR 8547 (Initial Public Draft), "Transition to Post-Quantum Cryptography Standards." November 2024. Explicitly supports hybrid implementations during the transition period.

3. Cloudflare Blog, "Automatically Secure: How We Upgraded 6,000,000 Domains." September 2025. Reports approximately 43% of human-generated connections using hybrid PQC key exchange as of mid-September 2025.

4. IETF draft-ietf-tls-ecdhe-mlkem (draft-04, February 2026). Specifies three hybrid groups for TLS 1.3: X25519MLKEM768 (0x11EC), SecP256r1MLKEM768, and SecP384r1MLKEM1024. Authored by Kwiatkowski (PQShield), Kampanakis (AWS), Westerbaan (Cloudflare), and Stebila (University of Waterloo).

5. NIST SP 1800-38C (Preliminary Draft): Quantum Readiness—Testing Draft Standards. Performance benchmarking showed hybrid ML-KEM + ECDH key exchange added only 1–2 milliseconds to TLS handshake latency in most configurations. Cloudflare's production data confirms negligible user-facing impact.

6. Bridge architecture concept described in F5, Inc. internal PQC field guidance (2025). BIG-IP 17.5.1 supports X25519MLKEM768 hybrid key exchange in TLS 1.3 on both client and server sides. BIG-IP v21.1 expands PQC cipher support.

7. RFC 8784, "Mixing Preshared Keys in the Internet Key Exchange Protocol Version 2 (IKEv2) for Post-Quantum Security." Specifies how to incorporate a post-quantum pre-shared key into IKEv2 key derivation.

8. NSA. draft-guthrie-cnsa2-ipsec-profile: CNSA Suite 2.0 Profile for IPsec. Specifies ML-KEM-1024 for key establishment in IPsec for National Security Systems.

9. OpenSSH 10.0 Release Notes (April 2025). Default key exchange changed to mlkem768x25519-sha256. See: IETF draft-ietf-sshm-mlkem-hybrid-kex for the specification.

10. Hybrid certificate sizes and TLS handshake fragmentation are covered in detail in Chapter 8. The PKI Consortium's PQC working group and IETF are developing dual-key certificate formats. Chrome has explicitly stated that certificate migration requires alternative approaches (Merkle Tree Certificates, trust expressions) due to size constraints.

11. NSA Cybersecurity Advisory, "Quantum Key Distribution (QKD) and Quantum Cryptography." States that NSA "does not support the use of QKD to protect communications in National Security Systems" and recommends PQC as "a more cost effective and easily maintained solution."

12. UK NCSC, "Quantum Networking Technologies." Updated 2025. States: "PQC is the best mitigation to the threat to cryptography from quantum computers" and "NCSC will not support the use of QKD for government or military applications." Notes that QKD does not provide authentication.

13. RAND Corporation, "U.S.-Allied Militaries Must Prepare for the Quantum Threat to Cryptography." June 2025. Notes that cyber/comms security agencies of UK, France, Germany, Netherlands, Sweden, and Czech Republic have all stated a clear preference for PQC over QKD. NSA prohibits QKD for NSS.

14. DoW CIO. Memorandum, "Preparing for Migration to Post Quantum Cryptography," November 18, 2025. Attachment, paragraph 2(a), prohibits DoW Components from testing, evaluating, piloting, using, or procuring QKD; QKD combined with other cryptographic key establishment; quantum communications or networking; non-local quantum randomness generation; or non-FIPS random number generation for confidentiality, authenticity, integrity, key distribution, or randomness generation, absent specific exception by the DoW CIO PQC Directorate. https://dodcio.defense.gov/Portals/0/Documents/Library/PreparingForMigrationPQC.pdf

Next: Chapter 8 — Protocol Deep Dives: TLS, IPsec, SSH, and PKI

CHAPTER 8

Protocol Deep Dives: TLS, IPsec, SSH, and PKI

This is the engineering chapter. The previous seven chapters built the case for why migration matters, what algorithms replace the vulnerable ones, and how to plan the program. This chapter goes inside the protocols themselves—byte by byte where it matters—to show exactly what changes when post-quantum cryptography enters the picture.

We'll focus on the areas that have the greatest operational impact: the TLS certificate size problem (which may be the single biggest deployment challenge in the entire PQC transition), the DNSSEC fragmentation cascade, IPsec IKEv2 key exchange changes, SSH authentication migration, and PKI chain restructuring.

The Certificate Size Problem: Why PQC Authentication Is Hard

Chapter 7 explained that hybrid key exchange is solved—X25519MLKEM768 adds roughly 1.1 KB to the client's key share and the performance impact is negligible. But key exchange is only half the TLS handshake. The other half is authentication—the certificate chain the server sends to prove its identity. That's where PQC creates a genuine engineering crisis.

The Math: Classical vs. PQC Authentication Data

A typical TLS 1.3 handshake today transmits approximately **1,248 bytes** of authentication data: five signatures and two public keys across the certificate chain and Certificate Transparency SCTs (Signed Certificate Timestamps). This fits easily inside the initial TCP flight.[1]

Replacing these with ML-DSA changes the picture dramatically. Here's the byte-by-byte accounting for a standard three-certificate chain (root CA, intermediate CA, leaf server certificate):[2]

Component	ECDSA P-256	ML-DSA-44	ML-DSA-65
Public key (leaf)	64 bytes	1,312 bytes	1,952 bytes
Public key (intermediate)	64 bytes	1,312 bytes	1,952 bytes
Signature × 3 (chain)	64 × 3 = 192 bytes	2,420 × 3 = 7,260 bytes	3,309 × 3 = 9,927 bytes
SCT signatures × 2	64 × 2 = 128 bytes	2,420 × 2 = 4,840 bytes	3,309 × 2 = 6,618 bytes
TLS handshake signature	64 bytes	2,420 bytes	3,309 bytes
X.509 metadata/extensions	~740 bytes	~740 bytes	~740 bytes
TOTAL AUTH DATA	**~1,248 bytes**	**~17,884 bytes**	**~25,138 bytes**

That's roughly a **14× increase with ML-DSA-44 and a 20× increase with ML-DSA-65** compared to today's classical certificates. The authentication data that currently fits in a kilobyte now consumes 17–25 KB.

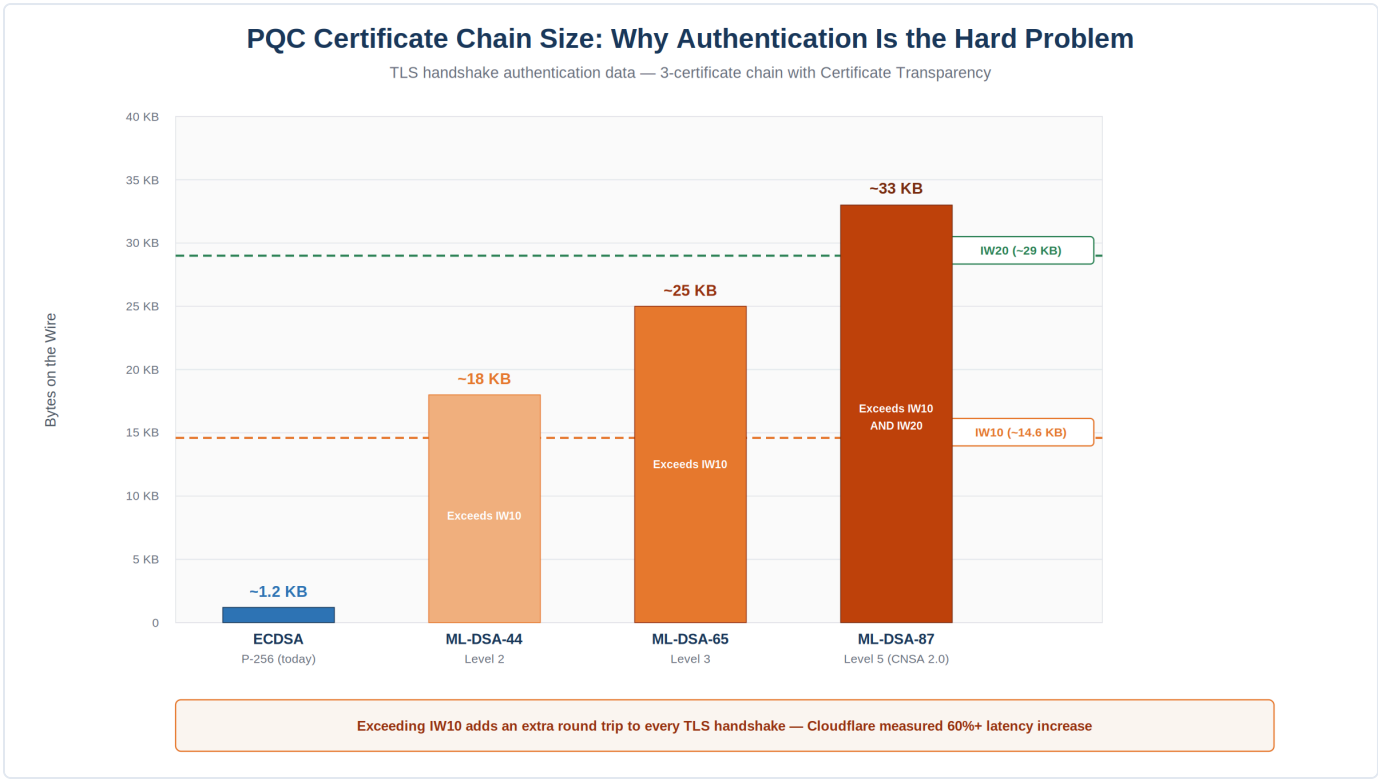

Figure 8.1 — PQC Certificate Chain Sizes vs. TCP Congestion Window Thresholds

> **PLAIN-LANGUAGE SIDEBAR** Imagine shipping a letter versus shipping a small package. The letter fits in any mailbox. The package might not—you might need to ring the doorbell, wait for someone to come to the door, and hand it over personally. That extra step is what happens when PQC certificates exceed the network's initial delivery window: the server has to pause, wait for an acknowledgment, and then continue. That pause adds a full round trip to every new connection.

The TCP Congestion Window Problem

When a TCP connection opens, the server doesn't flood the network with data. It starts with a limited **initial congestion window (initcwnd)**—the maximum amount of data it can send before waiting for the first acknowledgment from the client.

RFC 6928 standardized this at **IW10: 10 segments × 1,460 bytes = approximately 14.6 KB**. Many production servers, CDNs, and cloud load balancers now run IW20 (~29 KB), but IW10 remains the default on most Linux systems and many enterprise appliances.[3]

Here's the collision: a classical TLS 1.3 server response (ServerHello + certificate chain + key share + Finished) typically totals 4–6 KB—well within IW10. With PQC certificates:

- **ML-DSA-44 chain (~17 KB):** Exceeds IW10. Requires an extra round trip. Fits within IW20.
- **ML-DSA-65 chain (~25 KB):** Exceeds IW10 significantly. Marginal even for IW20 when combined with application data.
- **ML-DSA-87 chain (~33 KB):** Exceeds even IW20. Multiple extra round trips on default configurations.

Cloudflare's testing measured the real-world impact: adding approximately 9 KB to TLS handshakes caused roughly a 15% slowdown. Crossing the 10 KB threshold triggered an extra round trip that slowed handshakes by over 60%.[4] At scale—millions of new connections per second—that extra round trip adds measurable latency to every first page load, every API call, and every mobile app launch.

Google's Viability Threshold

Google's Chrome team has published a candid assessment of what's deployable:[5]

- **Adding ~2 KB to TLS handshakes:** "Very painful, but plausible."
- **Adding ~7 KB:** "Implausible unless a cryptographically relevant quantum computer is tangibly imminent."
- **No standardized PQC signature scheme** can stay under 7 KB for a full TLS certificate chain with Certificate Transparency. ML-DSA-65 pushes past 20 KB.

This is why Chrome is not simply dropping PQC signatures into the existing X.509 certificate infrastructure. The math doesn't work. Instead, Google has announced a fundamentally different architecture.

Merkle Tree Certificates: Google's Architectural Answer

In February 2026, Google announced **Merkle Tree Certificates (MTCs)**—a new certificate format that shrinks quantum-resistant TLS authentication data from roughly 14,700 bytes down to as little as 736 bytes. That's potentially smaller than today's classical certificate chains.[6]

The core insight: instead of every certificate carrying its own large post-quantum signature, a Certification Authority signs a single cryptographic commitment—a "Tree Head"—representing many certificates organized in a Merkle tree. Browsers receive compact inclusion proofs rather than full signature chains. The heavy ML-DSA signatures are applied once per batch of certificates, not once per individual certificate.

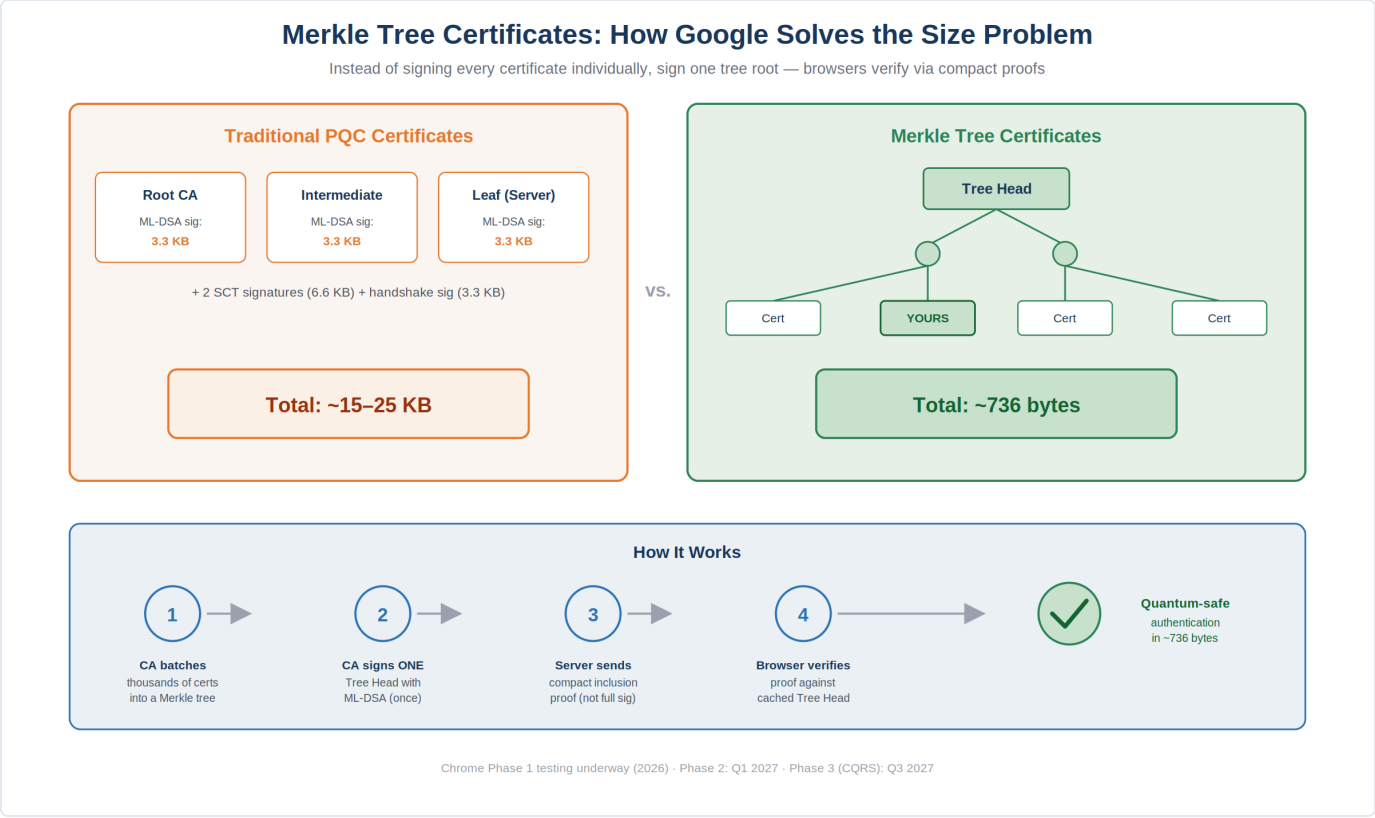

Figure 8.2 — Merkle Tree Certificates: Traditional PQC (~25 KB) vs. MTCs (~736 bytes)

MTCs also integrate Certificate Transparency directly into the issuance model. Because certificates must be included in a public tree, transparency becomes inherent—eliminating the separate SCT overhead that currently adds two extra signatures to every handshake.

> **PLAIN-LANGUAGE SIDEBAR** Think of today's certificate system like a notary who hand-signs every single document individually. Each signature is big and bulky, and every person who needs proof has to carry the full signed original. Merkle Tree Certificates work more like a notary who signs a single master ledger containing thousands of documents. Instead of carrying around the full signed original, you just carry a small receipt that proves your document is in the ledger. The notary's signature is just as trustworthy—but the receipt fits in your pocket.

The Rollout Timeline

Phase	Timeline	What Happens
Phase 1	2026 (underway)	Live feasibility study with Cloudflare. ~1,000 TLS certificates enrolled. Cloudflare operates as a "mock MTCA." Every MTC connection backed by traditional X.509 as fail-safe.
Phase 2	Q1 2027 (target)	Invite existing CT Log operators to bootstrap public MTC infrastructure. Only operators with a "usable" Chrome CT log before Feb 1, 2026 eligible.
Phase 3	Q3 2027 (target)	Launch Chrome Quantum-resistant Root Store (CQRS)—a separate trust store supporting only MTCs. Operates alongside existing Chrome Root Program.

The standardization work is proceeding through the IETF's newly formed **PLANTS working group**, jointly developed by Google and Cloudflare. This is the most significant structural change to the Web PKI since Certificate Transparency itself.[7]

> **MANDATE ALERT** MTCs are a Chrome/Cloudflare initiative—not yet a universal standard. Non-browser TLS clients (API consumers, IoT devices, mobile apps, server-to-server traffic) will still need to handle traditional PQC certificates for the foreseeable future. Your migration plan must account for both paths: MTCs for web-facing traffic and traditional PQC certificates for everything else.

Mitigations for Traditional PQC Certificate Chains

While MTCs are the long-term architectural answer for web browsers, there are near-term techniques to reduce the impact of large PQC certificates in traditional deployments:

Increase the initial congestion window. Raising from IW10 to IW20 accommodates approximately 29 KB of server response in the first flight—enough for ML-DSA-44 and most ML-DSA-65 chains to complete in a single round trip. On Linux, this is a sysctl or ip route change.[8]

TLS certificate compression. RFC 8879 defines certificate compression for TLS 1.3. Compression can reduce on-the-wire certificate chain sizes by 40–60%, potentially keeping PQC chains within IW10. Both client and server must support it, but adoption is growing in modern browsers and web servers.[9]

Intermediate certificate suppression. If browsers pre-load known intermediate CA certificates, servers don't need to send them during the handshake. Firefox already pre-loads over 1,400 intermediate certificates. Suppressing the intermediate saves 4–5 KB per handshake—a substantial reduction when every kilobyte matters.

Use ML-DSA-44 where Level 3 isn't required. ML-DSA-44 provides NIST Security Level 2 (equivalent to AES-128) with 1,312-byte public keys and 2,420-byte signatures—roughly 35% smaller than ML-DSA-65. For web-facing TLS where certificate lifetimes are short and CNSA 2.0 compliance isn't mandated, Level 2 may be sufficient.

FN-DSA (FIPS 206) for compact signatures. Falcon produces 666-byte signatures at Level 1—3.6× smaller than ML-DSA-44. A full FN-DSA chain adds only 5–8 KB. The trade-off: Falcon's signing algorithm requires precise floating-point arithmetic that makes constant-time implementation difficult, so it's best suited for infrequent CA-level signing rather than high-volume leaf certificate issuance.[10]

DNSSEC: The UDP Fragmentation Cascade

If TLS has a certificate size problem, DNSSEC has a certificate size crisis. DNS operates primarily over UDP, where the constraints are far tighter than TCP.

The recommended maximum DNS message size to avoid IP fragmentation is **1,232 bytes** (based on IPv6's minimum MTU of 1,280 bytes minus 48 bytes for headers). Classical DNSSEC signatures from ECDSA P-256 (64 bytes) or RSA-2048 (256 bytes) fit comfortably.[11]

Post-quantum signatures do not. Even the smallest NIST-standardized PQC signature—FN-DSA-512 at 666 bytes per signature—exceeds the 1,232-byte limit when a DNSSEC response contains two or three signatures (as NSEC/NSEC3 denial-of-existence responses do). ML-DSA-44 signatures at 2,420 bytes each make the problem far worse.[12]

When a DNSSEC response exceeds the UDP limit, one of two things happens:

- **IP fragmentation:** The response is split across multiple UDP packets. Approximately 10% of resolvers fail to reassemble IP fragments correctly, and many firewalls and middleboxes drop them entirely.
- **TCP fallback:** The server sets the truncation (TC) bit, and the resolver retries the query over TCP. This adds a three-way handshake plus at least one additional round trip—roughly doubling DNS resolution time for affected queries.

Research presented at IETF hackathons and the NIST PQC conferences has shown that even Falcon-512 can trigger TCP fallback in some DNSSEC scenarios. ML-DSA variants consistently force fallback.[13]

What the Community Is Exploring

DNSSEC's PQC migration is further behind than TLS. The IETF has established a dedicated mailing list (pq-dnssec@ietf.org) and hosted multiple PQC DNSSEC hackathons. Several approaches are under active investigation:

- **QNAME-based fragmentation (QBF):** Application-layer fragmentation that splits large responses into manageable DNS-native chunks, resolving queries in roughly half the time of standard TCP fallback.
- **SLH-DSA in Merkle Tree Ladder (MTL) mode:** Amortizes the large SLH-DSA signature across many DNS records, using compact inclusion proofs for individual queries—conceptually similar to MTCs for TLS.
- **Smaller signature algorithms:** NIST's additional signature call includes candidates like MAYO, Hawk, and SNOVA that may offer better size profiles for DNS. These are still under evaluation.

The DNS root zone's Key Signing Key (KSK) rollover—the most consequential DNSSEC event—is expected around 2028–2029. Whether that rollover will incorporate PQC algorithms or remain classical is an open question with significant implications for the entire DNS hierarchy.[14]

IPsec IKEv2: ML-KEM Integration

Chapter 7 introduced Post-Quantum Pre-Shared Keys (PPKs) as the immediate stopgap for IPsec environments. The long-term destination is native ML-KEM integration in IKEv2, which replaces the classical Diffie-Hellman key exchange with post-quantum key encapsulation.

The CNSA 2.0 IPsec profile specifies **ML-KEM-1024** for key establishment in National Security Systems. The protocol changes are relatively contained compared to TLS: IKEv2 already supports pluggable key exchange mechanisms through its Transform Type 4 (Diffie-Hellman Group) negotiation. Replacing a classical DH group with ML-KEM-1024 follows the same negotiation flow—the primary difference is message size.[15]

ML-KEM-1024 produces a 1,568-byte public key and 1,568-byte ciphertext—substantially larger than the 256-byte DH group 14 or 32-byte X25519 shares used in classical IPsec. For typical site-to-site VPN tunnels with

long-lived SAs, this per-SA overhead is manageable. For deployments with thousands of dynamic tunnels (large SD-WAN fabrics, hub-and-spoke architectures), the aggregate key exchange bandwidth becomes a capacity planning consideration.

Authentication in IKEv2 also requires PQC migration. When ML-DSA certificates replace RSA or ECDSA certificates for IKE authentication, the same certificate size challenges from TLS apply—amplified in mutual TLS scenarios where both sides present certificate chains.[16]

SSH: The Simplest Migration Path

SSH continues to be the protocol with the smoothest PQC transition, as we previewed in Chapter 7.

Key exchange is already PQC-ready. OpenSSH 10.0 (April 2025) defaults to mlkem768x25519-sha256. The hybrid exchange adds roughly 2.3 KB to the key exchange—noticeable in theory, but in practice, SSH sessions are long-lived and the one-time handshake overhead is amortized over the session's lifetime.[17]

Host key authentication is the remaining migration task. SSH host keys are currently Ed25519 or RSA. Replacing them with ML-DSA host keys means larger SSH server identification payloads, but SSH doesn't have TLS's certificate chain overhead—there's no intermediate CA hierarchy. A single ML-DSA-65 host key adds approximately 5.3 KB (1,952-byte public key + 3,309-byte signature), which is manageable.

User authentication via PQC keys follows the same pattern. If your environment uses SSH certificates (rather than bare public keys), the certificate sizes will mirror the ML-DSA figures above—but again, without the multi-level chain amplification that makes TLS certificates so challenging.

Secure Email: S/MIME and PGP

For organizations that handle classified or sensitive communications—particularly in DoD, intelligence, and federal civilian agencies—S/MIME is the primary mechanism for signed and encrypted email. PGP (and its open standard, OpenPGP) serves a similar role in some environments.

Both protocols face the same PQC challenges as TLS and IPsec: key exchange algorithms (RSA, ECDH) must be replaced with ML-KEM, and signature algorithms (RSA, ECDSA) must be replaced with ML-DSA. The IETF has active drafts for both:

- **S/MIME:** The IETF LAMPS working group is developing composite certificate formats that bundle classical and PQC algorithms for Cryptographic Message Syntax (CMS). Draft standards for ML-KEM and ML-DSA in S/MIME are in progress.
- **OpenPGP:** RFC 9580 (the updated OpenPGP specification, published July 2024) includes provisions for PQC algorithm identifiers. The crypto-refresh working group has been preparing the groundwork for ML-KEM and ML-DSA integration.

PLAIN-LANGUAGE SIDEBAR The certificate size challenges from TLS apply directly to encrypted email. Every S/MIME signed message carries the sender's certificate chain. With PQC certificates, each signed email becomes significantly larger. For organizations processing millions of signed messages daily, the storage and bandwidth implications are material—and email archival systems designed for today's certificate sizes will need capacity planning updates.

PKI Chain Migration: The Long Pole in the Tent

The Public Key Infrastructure underpins everything above—TLS, IPsec, SSH certificates, code signing, email (S/MIME), document signing, and device identity. Migrating PKI is the deepest, most cross-cutting element of the PQC transition.

Phased PKI Migration

The practical migration sequence, informed by the NIST NCCoE's guidance and the CA/Browser Forum's evolving requirements:[18]

- **Phase 1 — Root and Intermediate CAs:** Issue new root certificates with PQC algorithms (ML-DSA-87 for roots needing CNSA 2.0, ML-DSA-65 for general purpose). Distribute these through trust store updates. This is the slowest step—root distribution takes years.
- **Phase 2 — Leaf certificates:** Begin issuing leaf server certificates with PQC algorithms. Initially, issue hybrid certificates (dual ECDSA + ML-DSA) for backward compatibility. Transition to pure PQC as client support matures.
- **Phase 3 — Client certificates:** Migrate mutual TLS (mTLS) client certificates to PQC. This is especially impactful in Zero Trust environments where every client connection authenticates with a certificate.
- **Phase 4 — Non-web PKI:** Code signing, S/MIME email certificates, document signing, device identity certificates. Each has its own ecosystem, tooling, and migration challenges.

The PQC Root Key Ceremony

Generating a new PQC root private key is not a routine task. Public and private CAs have operated under formal key ceremony practices for decades—tied to WebTrust for Certification Authorities audit requirements, CA/Browser Forum Baseline Requirements, and NIST SP 800-57 Part 2 key management guidance—to ensure that the act of creating the root key pair is witnessed, scripted, and auditable. PQC migration does not change this discipline; it extends it. The same ceremonial controls apply, with a few PQC-specific additions around algorithm selection, HSM firmware verification, and—for stateful hash-based signatures—state management.[19]

The section that follows describes the ceremony controls that organizations issuing PQC root or intermediate CA keys should plan for. It is not a replacement for your CA's Certificate Policy (CP) and Certification Practice Statement (CPS); those documents describe what your specific organization commits to. The following checklist is the common baseline shared across public CAs, government PKIs, and large private CAs.

Pre-Ceremony Checklist

The most common reason key ceremonies fail audit is inadequate preparation. Before convening witnesses and unsealing the HSM:

- **Algorithm and parameter selection:** Confirm the specific PQC algorithm, parameter set, and hash function (e.g., ML-DSA-87 for CNSA 2.0 roots, ML-DSA-65 for general purpose, LMS or XMSS for code signing). Document the rationale in the ceremony script.
- **HSM firmware verification:** Verify the exact HSM firmware version supports the chosen algorithm and is at a FIPS 140-3 validation state acceptable to your audit scheme. Record the firmware hash or version string in the ceremony log.
- **CP/CPS alignment:** Ensure the ceremony script implements what your CP/CPS describes. Discrepancies between script and published CPS are audit findings.
- **Physical and logical access:** Reserve the ceremony room. Verify smart-card / M-of-N token inventory. Confirm tamper-evident bags, seal numbers, and evidence-of-integrity mechanisms are on hand.
- **Witness identification:** Designate and identify all ceremony participants and their roles (see below). All participants should have completed background checks consistent with your CP/CPS.
- **Ceremony script review:** Walk the script end-to-end with all roles present at least 24 hours in advance. The live ceremony is not the time to discover missing steps.

Multi-Person Control

Root key operations require multi-person control. Two related but distinct disciplines apply:

- **Split knowledge:** No single person has enough information to operate the root key alone. In practice, this means M-of-N smart-card quorums (commonly 3-of-5 or 5-of-7 for tier-1 public roots) holding HSM partition authentication material.
- **Dual control:** Every operational step on the root HSM requires at least two people present, each with independent authentication. Neither can proceed without the other.

PQC does not change these controls, but it does change what the quorum is authorizing. A ceremony to generate ML-DSA-87 keys produces fundamentally different artifacts than one generating RSA-4096: key sizes differ (ML-DSA-87 public keys are 2,592 bytes versus 512 bytes for RSA-4096), signature sizes differ (ML-DSA-87 signatures are 4,627 bytes versus 512 bytes), and—for stateful hash-based algorithms—the quorum is authorizing the creation of a state-managed key, not a stateless one. Witnesses should understand what they are attesting to.

Script-Driven Operations

Every action during the ceremony follows a pre-approved, reviewed, and signed-off script. The script specifies exact commands, expected outputs, and decision points. Any deviation triggers a documented exception process, and material deviations halt the ceremony. Three practical requirements:

- **Command-level specificity:** The script captures the exact HSM command syntax (keygen algorithm parameters, key label, quorum requirements). Generic instructions like “generate the root key” are insufficient for audit.
- **Expected-output capture:** For each command, the script includes the expected output or success criterion. The ceremony scribe compares actual output against expected and flags discrepancies in real time.
- **Version control:** The script is a versioned artifact. The exact version number used during the ceremony is recorded in the ceremony log and retained with the audit evidence.

Tamper-Evident Logging

Every ceremony produces a contemporaneous log—the written record that the auditor will review. The log should capture:

- **Participants and roles:** Names, roles, and ID verification method for every person in the room.
- **Timestamps:** Start and end time for each script step. Use wall-clock time from a trusted source.
- **HSM artifacts:** Key label, algorithm parameters, public key fingerprint (post-generation), firmware version, and any error messages.
- **Physical evidence:** Tamper-evident bag numbers, seal numbers, smart-card identifiers, and chain-of-custody transfers.
- **Signatures:** All participants sign each page of the log at ceremony close. Electronic signatures are acceptable where permitted by CP/CPS.

Video recording of the ceremony is common for public CAs and is usually required by WebTrust for tier-1 root ceremonies. The recording becomes part of the audit evidence.

Witness Roles

A formal ceremony distinguishes multiple roles. The minimum set for most PKI hierarchies:

- **Ceremony Administrator (CA Officer):** Executes the script. Authenticates to the HSM. Operates the ceremony laptop or console.
- **Internal Witness:** Observes every step. Is not a smart-card holder. Independently verifies script adherence.
- **External Witness / Auditor:** For publicly trusted CAs under WebTrust, a Qualified Auditor attends and issues an opinion that the ceremony was conducted per the CA’s stated procedure. For private PKIs, this role may be an internal audit function or a trusted third party.
- **Scribe:** Maintains the contemporaneous log. Should not be the same person as the Ceremony Administrator.
- **Smart-card / Quorum Holders:** M-of-N token holders required to authenticate key operations. Physically present and identified in the log.

- **Security Officer:** Controls physical access to the ceremony room. Maintains custody of tamper-evident materials before and after the ceremony.

Post-Ceremony Verification

The ceremony is not complete when the key is generated. Post-ceremony verification confirms that the key operates as intended and that all artifacts are sealed and stored correctly:

- **Public key fingerprint verification:** Compute the fingerprint of the newly generated public key using two independent tools. Compare against the fingerprint captured in the log. Discrepancy is a ceremony failure requiring reconvening.
- **Test signature and verification:** Sign a known test vector with the new root key. Verify the signature with the extracted public key. Record both the test vector and the verification result in the log.
- **Stateful hash-based signature state initialization:** If using LMS or XMSS for code signing, verify that the stateful signing counter is correctly initialized, that state backup and replication mechanisms are tested, and that the HSM enforces single-use of each signature state. Reusing an LMS/XMSS state is a catastrophic failure mode.
- **Artifact sealing:** Place all removable materials (smart cards, backup HSM cartridges, printed logs) into tamper-evident bags. Record seal numbers in the log. Transfer to secure storage under dual control.
- **Ceremony log finalization:** Close and sign the ceremony log. Distribute copies per CP/CPS retention policy. The log is the durable record; the ceremony cannot be reconstructed from memory.
- **Audit package assembly:** Assemble script version, ceremony log, video (if applicable), HSM firmware verification evidence, and public key fingerprint for delivery to the Qualified Auditor.

PQC ceremonies differ in detail from classical ceremonies—larger keys, different HSM command syntax, stateful signature state management, new algorithm parameter sets to validate. They do not differ in discipline. The same multi-person control, the same scripted execution, the same tamper-evident logging, the same witness roles apply. Organizations that have run classical root ceremonies successfully have most of the operational muscle they need; what changes is the content of the ceremony, not its structure.

The mTLS Amplification Effect

Most TLS performance studies focus on server authentication—the server sends its certificate chain to the client. In mTLS environments (common in Zero Trust, service mesh, and API gateway architectures), the client also sends a certificate chain. With PQC, the handshake is doubly impacted: a server chain of ~17 KB plus a client chain of ~17 KB means the handshake could exceed 34 KB of authentication data—well beyond IW20.[20]

For organizations running mTLS at scale (every microservice authenticating to every other microservice), this is a critical capacity planning consideration that most PQC migration guides overlook.

Shorter Certificate Lifetimes Compound the Problem

The CA/Browser Forum's Ballot SC-081v3 sets a schedule that shrinks publicly trusted TLS certificate validity: 200 days starting March 2026, 100 days by March 2027, and 47 days by March 2029.[21] Shorter lifetimes mean more frequent issuance, which means paying the PQC overhead tax more often. The combination of larger cer-

tificates and higher issuance velocity is what makes a simple drop-in replacement unsustainable at internet scale—and why architectural solutions like MTCs are necessary.

> **F5 PERSPECTIVE** **BIG-IP Capacity Planning for PQC Certificates** BIG-IP devices that terminate TLS must account for the larger PQC certificate chains in their memory and throughput planning. Key considerations: **Memory per connection:** Each active TLS session stores the peer's certificate chain in memory during the handshake. With ML-DSA-65 certificates, this increases from roughly 4 KB to 20+ KB per session. At 100,000 concurrent connections, that's an additional 1.5+ GB of memory dedicated to certificate storage alone. **Bandwidth:** A BIG-IP serving 10 million TLS connections per day with ML-DSA-65 certificates transmits approximately 250 GB more certificate data daily than with classical certs. For most enterprise deployments, this is well within infrastructure capacity—but it's a line item in capacity planning, not invisible. **Initial congestion window:** BIG-IP supports configurable TCP profiles, including the initial congestion window size. Increasing initcwnd from 10 to 20 on internet-facing virtual servers may be the single highest-impact configuration change for PQC readiness—a one-line profile modification that eliminates the extra round trip for most PQC certificate chains. **TLS certificate compression:** As BIG-IP adds support for RFC 8879 TLS certificate compression, enabling it alongside PQC certificates will be a critical optimization. Monitor F5's release notes for availability.

Zero Trust and IAM in a Post-Quantum World

Zero Trust Architecture is predicated on a simple premise: verify every request, and let no network location confer implicit trust. NIST SP 800-207 codifies this premise through seven tenets, chief among them that all communication is secured regardless of network location, and that resource authentication and authorization are dynamic and strictly enforced. CISA's Zero Trust Maturity Model v2.0 and OMB M-22-09 have since made Zero Trust the expected operating model for federal agencies. The premise is sound. The challenge for a PQC migration is that every Zero Trust control depends on cryptographic identity—certificates, signatures, tokens, attestations—and every one of those controls inherits the quantum-vulnerability of its underlying algorithms.[22]

Zero Trust does not introduce new cryptographic problems. It amplifies the ones already described in this chapter. Where a perimeter model authenticates at the edge and trusts the interior, Zero Trust authenticates at every hop. Every service-to-service call is mTLS. Every API request is signed. Every device presents a certificate. Every user session is re-evaluated. The cryptographic surface area is larger by one or two orders of magnitude, which means the PQC migration cost is larger by the same factor.

Where the PQC Pressure Lands

- **Workload-to-workload mTLS.** Service mesh architectures (Istio, Linkerd, Consul Connect) and SPIFFE-based workload identity frameworks issue short-lived certificates to every workload. With PQC, each handshake pays the certificate-size cost (Chapter 8 certificate size section) on both sides. The mTLS amplification effect is not a corner case in Zero Trust; it is the common case.
- **Signed tokens (OIDC, OAuth 2.0, SAML).** Identity providers sign access tokens, ID tokens, and SAML assertions. Those signatures must transition to ML-DSA for long-term verifiability, especially where tokens are archived or replayed against audit logs. JWTs with RSA/ECDSA signatures remain forgeable to an adversary with a CRQC, even after the session has ended.

- **Identity provider (IdP) signing keys.** An IdP's signing key is a single point of compromise with enormous cryptographic blast radius: compromising it forges every identity assertion the IdP issues. IdP root keys are prime candidates for early PQC migration, and their migration follows the same ceremony discipline described earlier in this chapter.
- **ZTNA brokers and the cryptographic proxy layer.** ZTNA platforms terminate TLS at a broker that evaluates policy before connecting the client to a resource. This is the cryptographic proxy layer pattern (Chapter 7). The broker is the right place to enable hybrid PQC first: one upgrade point protects many downstream resources. F5 BIG-IP Zero Trust Access, added in v21.1, is one commercial example of this pattern.
- **Device and workload attestation.** Device posture checks, TPM attestations, and workload identity proofs all produce signed claims that relying parties verify. Those signatures must become PQC before the signed-claim lifetime exceeds the attacker's time-to-CRQC. For devices with multi-year deployment lifecycles (industrial control, medical, embedded), this is already urgent.

Zero Trust does not change the PQC migration work; it changes the scope. An organization that has built a Zero Trust Architecture has already committed to strong cryptographic identity everywhere, which means that same organization is committed to replacing every instance of quantum-vulnerable cryptography in that identity fabric. The good news: the migration maps cleanly onto existing Zero Trust investments. Policy Decision Points and Policy Enforcement Points are natural upgrade targets; cryptographic proxy layers consolidate the upgrade into fewer places; and workload identity frameworks like SPIFFE were designed with crypto-agility in mind. The Zero Trust roadmap and the PQC roadmap should not be two programs. They should be one.

Protocol Migration Summary

Protocol	Key Exchange Status	Authentication Status	Biggest Challenge
TLS 1.3	Solved — X25519MLKEM768 deployed at scale	In progress — MTCs in Phase 1 testing	Certificate size exceeds TCP initcwnd; MTCs needed for web scale
IPsec	PPK stopgap deployed; native ML-KEM in CNSA 2.0 profile	ML-DSA certs for IKE auth; mTLS amplification	Large-scale SD-WAN/hub-spoke key exchange bandwidth
SSH	Solved — mlkem768x25519 default in OpenSSH 10.0	Host/user key migration to ML-DSA pending	Minimal — no cert chain overhead; single key per host
DNSSEC	N/A (signatures only)	Research phase — MTL mode, QBF, new algorithms	UDP 1,232-byte limit; even Falcon-512 triggers fallback
PKI	N/A	Root/intermediate CA migration beginning	Root trust store distribution takes years; mTLS doubles overhead

What's Next

You now understand what changes at the protocol level and where the pain points are. Chapter 9 shifts to the operational reality: once PQC is deployed, how do you monitor it, manage certificate rotation at scale, handle performance regressions, train your team, and ensure long-lived signed artifacts remain trustworthy in a post-quantum world?

Notes

The following sources support specific claims made in Chapter 8. Full bibliographic entries appear in the Bibliography.

1. NIST PQC Conference presentation, Andrew Regenscheid & Bill Newhouse (December 2024). TLS WebPKI authentication data breakdown: server certificate (1 public key + 1 signature + 2 SCT signatures), intermediate CA certificate (1 public key + 1 signature), TLS handshake (1 signature). Classical total: ~1,248 bytes. ML-DSA-44 total: ~14,724 bytes.

2. NIST FIPS 204 (ML-DSA) specifies signature and public key sizes. ML-DSA-44: 1,312-byte public key, 2,420-byte signature. ML-DSA-65: 1,952-byte public key, 3,309-byte signature. ML-DSA-87: 2,592-byte public key, 4,627-byte signature. X.509 metadata overhead varies by certificate; ~740 bytes is a representative figure including extensions.

3. RFC 6928, "Increasing TCP's Initial Window," standardized IW10 (10 segments). Many production systems, CDNs, and cloud providers now use IW20 or higher. Default Linux initcwnd remains 10 as of kernel 6.x.

4. Cloudflare infrastructure testing and NIST 5th PQC Standardization Conference paper, "The Impact of Data-Heavy Post-Quantum TLS 1.3." Testing showed ~15% slowdown at +9 KB, 60%+ slowdown when crossing the 10 KB threshold due to extra round trip from congestion control interaction.

5. Google Chrome team analysis cited in multiple sources including the MTC announcement (February 2026). Google's threshold: +2 KB is "very painful but plausible"; +7 KB is "implausible unless a CRQC is tangibly imminent." No standardized PQC signature scheme meets the 7 KB threshold for a full chain with CT.

6. Google Security Blog, "Cultivating a Robust and Efficient Quantum-Safe HTTPS," February 2026. Merkle Tree Certificates (MTCs) reduce authentication data from ~14,700 bytes to as low as 736 bytes. Developed jointly with Cloudflare; standardization through IETF PLANTS working group.

7. IETF PLANTS (Post-quantum Lightweight Authentication for Network TLS Security) working group formed to standardize MTCs. Phase 1 testing underway with ~1,000 certificates enrolled (Cloudflare as mock MTCA). Phase 2 targets Q1 2027; Phase 3 (Chrome Quantum-resistant Root Store) targets Q3 2027.

8. Increasing TCP initcwnd from IW10 to IW20 accommodates ~29 KB in the first server flight. On Linux: ip route change default via initcwnd 20 initrwnd 20. This is a well-understood optimization already deployed by many CDNs and cloud providers.

9. RFC 8879, "TLS Certificate Compression." Defines zlib, Brotli, and Zstandard compression for TLS 1.3 certificate messages. Can reduce PQC certificate chain sizes by 40–60%. Requires both client and server support.

10. NIST FIPS 206 (draft), FN-DSA (Falcon). FN-DSA-512 produces 666-byte signatures—3.6× smaller than ML-DSA-44's 2,420 bytes. Full FN-DSA chain: ~5–8 KB. Cloudflare analysis notes Falcon's floating-point signing makes constant-time implementation extremely difficult; better suited for CA-level signing than high-volume leaf issuance.

11. IETF DNS Flag Day 2020 recommendation: EDNS buffer size of 1,232 bytes avoids fragmentation on nearly all current networks. Based on IPv6 minimum MTU of 1,280 bytes minus 48 bytes for IPv6 and UDP headers.

12. IETF draft-fregly-research-agenda-for-pqc-dnssec-02. Even FN-DSA-512 (666-byte signatures) exceeds the 1,232-byte limit with two or three signatures in NSEC/NSEC3 responses. ML-DSA signatures (2,420–4,627 bytes) make UDP transport of DNSSEC responses effectively impossible.

13. NIST 6th PQC Standardization Conference, VeriSign/NIST presentation: "Post-Quantum Diversity for DNSSEC: Routine Performance, Resilient Fallback." PQC DNSSEC hackathon at IETF 123 (July 2025) tested implementations in BIND, NSD, and CoreDNS with ML-DSA, FN-DSA, SLH-DSA, MAYO, SQIsign, Hawk, and SNOVA.

14. DNS root zone KSK rollover expected ~2028–2029 per VeriSign/NIST presentations. Whether PQC algorithms will be incorporated into root zone DNSSEC operations is an open question under active discussion at IETF and ICANN.

15. NSA, draft-guthrie-cnsa2-ipsec-profile: CNSA Suite 2.0 Profile for IPsec. Specifies ML-KEM-1024 for key establishment. ML-KEM-1024: 1,568-byte public key, 1,568-byte ciphertext.

16. NIST PQC Conference presentation on mTLS impact: "The Impact of ML-KEM and ML-DSA on mTLS Connection Time-To-Last-Byte." In mTLS (mutual TLS), both client and server send certificate chains, doubling the authentication data overhead in both directions.

17. OpenSSH 10.0 Release Notes (April 2025). Default key exchange: mlkem768x25519-sha256. Adds ~2.3 KB to key exchange compared to classical X25519.

18. NIST NCCoE SP 1800-38 (Preliminary Draft): Migration to Post-Quantum Cryptography. PKI migration guidance covers root CA establishment, intermediate CA issuance, leaf certificate transition, and non-web PKI considerations.

19. PQC root key ceremony controls draw from three canonical sources: NIST SP 800-57 Part 2 Rev 1 (Recommendation for Key Management: Best Practices for Key Management Organizations, May 2019); WebTrust Principles and Criteria for Certification Authorities v2.2.2 (CPA Canada / AICPA); and the CA/Browser Forum Baseline Requirements, Section 8 (Audit), which requires that a Qualified Auditor opine on the CA's key ceremony during key and certificate generation and on the controls used to ensure the integrity and confidentiality of the key pair. Publicly trusted CAs are subject to annual WebTrust audit; private and government CAs typically follow the same practices through CP/CPS commitments.

20. mTLS amplification: a server chain (~17 KB with ML-DSA-44) plus client chain (~17 KB) totals ~34 KB of authentication data in a single handshake—exceeding IW20 (~29 KB). This is especially relevant in Zero Trust architectures and service mesh deployments.

21. CA/Browser Forum Ballot SC-081v3. Publicly trusted TLS certificate maximum validity: 398 days (current), 200 days (March 2026), 100 days (March 2027), 47 days (March 2029). Domain validation data reuse tightens to 10 days by end of timeline.

22. Zero Trust foundational references: NIST SP 800-207, "Zero Trust Architecture" (August 2020), defines the seven tenets and the Policy Engine / Policy Administrator / Policy Enforcement Point component model. NIST SP 800-207A, "A Zero Trust Architecture Model for Access Control in Cloud-Native Applications in Multi-Cloud Environments" (September 2023), extends the model for workload-identity patterns including SPIFFE. CISA Zero Trust Maturity Model v2.0 (April 2023) organizes maturity across five pillars (Identity,

Devices, Networks, Applications & Workloads, Data). OMB Memorandum M-22-09 (January 2022) establishes the federal Zero Trust strategy for executive branch agencies.

Next: Chapter 9 — Day-2 Operations: Monitoring, Rotation, and Long-Term Assurance

CHAPTER 9

Day-2 Operations: Monitoring, Rotation, and Long-Term Assurance

Deploying post-quantum cryptography is a milestone, not a finish line. Once PQC is live in your environment—hybrid TLS on the edge, updated SSH key exchanges, new certificate chains in the pipeline—a new set of operational challenges begins. This chapter covers what happens after the migration: how you monitor PQC in production, manage certificates that are 10× larger, protect long-lived signed artifacts, keep your vendor ecosystem aligned, and build the institutional knowledge to sustain the program.

Protecting Long-Lived Signed Artifacts

Most of the PQC migration discussion focuses on data in transit—TLS sessions, VPN tunnels, SSH connections. But some of the most consequential cryptographic artifacts in your environment aren't protecting live traffic. They're protecting evidence: signed firmware images, software bills of materials (SBOMs), audit logs, legal contracts, code signing certificates, and regulatory filings that must remain verifiable for years or decades.

These artifacts face a threat that live TLS connections do not: **harvest-now, forge-later**. An adversary who captures a classically signed firmware image today could, with a future quantum computer, forge an altered version with a valid signature—retroactively compromising the trust chain. This isn't theoretical; it's the signature-side analog of the HNDL attack we described in Chapter 1.[1]

What Needs Protection

Artifact Type	Typical Retention	Quantum Risk
Firmware images	10–20+ years (embedded/OT)	Forged firmware accepted as authentic; supply chain compromise
Code signing certificates	5–10 years	Malicious code signed with forged certificate trusted by endpoints
SBOMs / build manifests	Lifetime of deployed software	Tampered SBOM hides vulnerable or malicious components
Audit logs / compliance records	7–25 years (regulatory)	Forged or altered audit trail; repudiation of signed actions
Legal contracts / e-signatures	Decades	Signatory repudiates contract; forged amendments accepted as valid
Timestamping authority records	Decades	Forged timestamps alter the provable sequence of events

Migration Patterns for Legacy Evidence

Researchers have formalized three practical patterns for protecting existing signed artifacts:[2]

Pattern 1: Hybrid signatures for new artifacts. Starting now, sign all new firmware, SBOMs, audit records, and code releases with both a classical signature and a PQC signature (ML-DSA). If the classical signature is later broken, the PQC signature preserves integrity. This is the CNSA 2.0 approach for software and firmware signing, with a "prefer PQC by 2025" target.[3]

Pattern 2: Re-signing legacy artifacts. For existing signed artifacts that must remain verifiable beyond Q-Day, re-sign them with a PQC key inside a trusted environment (HSM or TEE). This retroactively extends the evidentiary lifetime of legacy records. The trust assumption: the original artifacts were unmodified at the time of re-signing.

Pattern 3: Merkle root anchoring. For large batches of legacy records, compute a Merkle tree over the batch and sign only the root with a PQC signature. Individual records are verified via compact inclusion proofs against the signed root. This amortizes the cost of PQC signatures across thousands of records—a practical approach for audit log archives.

> **PLAIN-LANGUAGE SIDEBAR** Think of re-signing legacy artifacts like a notary re-stamping old documents with a new, tamper-proof seal. The original signatures are still there—they prove the document was authentic when it was first signed. The new PQC seal proves it hasn't been altered since, even if someone eventually finds a way to forge the original stamp.

Certificate Lifecycle Management at Scale

PQC certificates are larger, expire more frequently (per the CA/Browser Forum's tightening validity schedule), and involve new algorithms that your existing tooling may not fully support. The certificate lifecycle—issuance, distribution, installation, monitoring, renewal, and revocation—becomes more demanding across every step.

Key Operational Changes

Storage and memory. Certificate stores on endpoints, load balancers, and HSMs will consume significantly more space. A PQC certificate chain that once fit in 3–4 KB now requires 17–25 KB. At scale, this affects memory allocation per TLS session and certificate cache sizing.

Renewal velocity. With publicly trusted certificates shrinking to 47-day maximum validity by 2029, automated certificate management becomes non-optional. Manual renewal processes will collapse under the volume.[4] The tooling landscape has traditionally concentrated in three places: standalone ACME clients (certbot, acme.sh), Kubernetes controllers (cert-manager), and enterprise CLM platforms (Venafi, Keyfactor, AppViewX). A newer pattern worth tracking is native ACME support inside the TLS enforcement point itself. Apache HTTP Server has supported it since 2.4.30; Caddy and Traefik treat automatic certificate management as default behavior; NGINX released the ngx_http_acme_module in August 2025.[5] This shift matters because it removes a class of external orchestration: the server or reverse proxy that terminates TLS also handles its own certificate renewal, with no intermediate automation tier to maintain. Expect native ACME support to expand across enterprise ADC platforms as 47-day validity approaches.

Revocation checking. OCSP responses and CRLs that carry PQC signatures will also be larger. Stapling OCSP responses (where the server pre-fetches and attaches the OCSP response to the TLS handshake) becomes

even more important to avoid per-client OCSP lookup overhead.

HSM compatibility. The five HSM readiness questions from Chapter 6 remain critical in Day-2 operations. Ongoing firmware updates from HSM vendors (Thales, Entrust, Marvell/Cavium) will add PQC algorithm support incrementally. Track vendor release notes and plan HSM firmware upgrades into your maintenance windows.[6]

> **MANDATE ALERT** Microsoft announced general availability of PQC APIs (ML-KEM and ML-DSA) in Windows Server 2025 and Windows 11 (24H2/25H2) via CNG, with Active Directory Certificate Services (ADCS) PQC support targeted for early 2026. If you run a Microsoft PKI, this is your on-ramp for issuing PQC certificates from your enterprise CA. AWS KMS and Google Cloud KMS both support ML-DSA for digital signatures. These services can sign firmware, SBOMs, and other artifacts with PQC today—no HSM upgrade required.

Performance Monitoring and Regression Detection

PQC introduces measurable performance changes. In most cases, they're small enough to be invisible to end users (1–2 ms on a TLS handshake). But in edge cases—high-latency networks, mobile connections, mTLS-heavy service meshes, or misconfigured initial congestion windows—the impact can compound.

What to Monitor

Metric	What to Watch For	Action Threshold
TLS handshake latency (p50/p95/p99)	Increase after enabling PQC certificates or hybrid key exchange	p95 increase >50 ms suggests congestion window or cert size issue
TLS handshake failure rate	Middlebox or client incompatibility with larger handshakes	Any increase >0.1% warrants investigation of specific client/path
Certificate chain size on the wire	Unexpected growth (e.g., dual-signed hybrid certs larger than expected)	Chains exceeding 20 KB on IW10 systems need initcwnd tuning
SSH key exchange duration	Baseline change after mlkem768x25519 becomes default	Typically <5 ms increase; larger suggests network or library issue
VPN tunnel establishment time	IKEv2 rekeying with ML-KEM or PPK overhead	Monitor per-tunnel and aggregate; flag rekeying storms
Memory consumption per connection	Larger cert chains stored in session memory	Correlate with connection count; plan for 5–10× cert memory growth

The key principle: **baseline before you migrate**. Capture your current TLS handshake latency distribution, failure rates, and memory profiles before enabling PQC. Without a clean baseline, you can't distinguish PQC-induced regressions from unrelated changes.[7]

Vendor and Supply Chain Readiness

Your PQC migration is only as strong as your weakest vendor. If a third-party SaaS provider, payment processor, or API gateway still uses RSA-2048, your data transiting that interface remains quantum-vulnerable regardless of your internal readiness.[8]

The Vendor PQC Readiness Conversation

For each critical vendor and supplier, your procurement and security teams should be asking:

- **Algorithm support:** Do your products support ML-KEM and ML-DSA? Which versions? Is the support FIPS-validated or pending validation?
- **Timeline:** What is your published PQC migration roadmap? When will PQC be available in production releases?
- **Hybrid support:** Can your product operate in hybrid mode during the transition, or is it PQC-only?
- **Crypto-agility:** If a PQC algorithm is broken or deprecated, how quickly can you swap to an alternative? Is algorithm selection configurable by the customer, or does it require a vendor release?
- **CBOM disclosure:** Can you provide a Cryptographic Bill of Materials documenting which algorithms, key sizes, and protocols your product uses?

The USDA has already embedded explicit PQC procurement language in its acquisition regulations—requiring that products in CISA-listed categories support PQC.[9] Other federal agencies will follow. For vendors selling to the public sector, PQC readiness is quickly becoming a contract requirement, not a differentiator.

Building Institutional Knowledge

PQC is not a one-time project that can be handed to a contractor and forgotten. It's a permanent shift in the cryptographic foundation of your infrastructure. The people who maintain your systems need to understand what changed and why.

Who Needs Training and What They Need to Know

Role	Core Knowledge	Depth
Network / infrastructure engineers	Hybrid TLS configuration, initcwnd tuning, cert compression, IPsec PPK setup, SSH key migration	Hands-on configuration and troubleshooting
PKI / identity team	PQC certificate issuance, CA hierarchy changes, HSM firmware updates, ACME automation, hybrid certificate formats	Deep operational expertise
Security operations (SOC)	Recognizing PQC-related handshake failures, cipher suite anomalies, and algorithm downgrade attacks	Detection and triage
Application developers	Library updates (OpenSSL 3.5+, BoringSSL, Windows CNG), API changes for PQC key generation, and testing strategies	Integration and testing
Leadership / CISO	Risk posture, compliance timeline, budget implications, vendor readiness assessment	Strategic awareness and decision authority

The CCOE (Cryptographic Center of Excellence) model from Chapter 6 provides the organizational structure; training fills it with capability. Consider tabletop exercises that simulate a PQC deployment failure—a certificate chain that breaks a critical application, a middlebox that drops hybrid handshakes, or an HSM that doesn't support ML-DSA yet. These exercises build muscle memory before the pressure of production.

The Crypto-Agility Feedback Loop

Chapter 6 introduced crypto-agility as an architectural principle. In Day-2 operations, crypto-agility becomes a continuous process: a feedback loop that keeps your cryptographic posture aligned with evolving standards, emerging threats, and real-world performance data.

The loop has four steps:

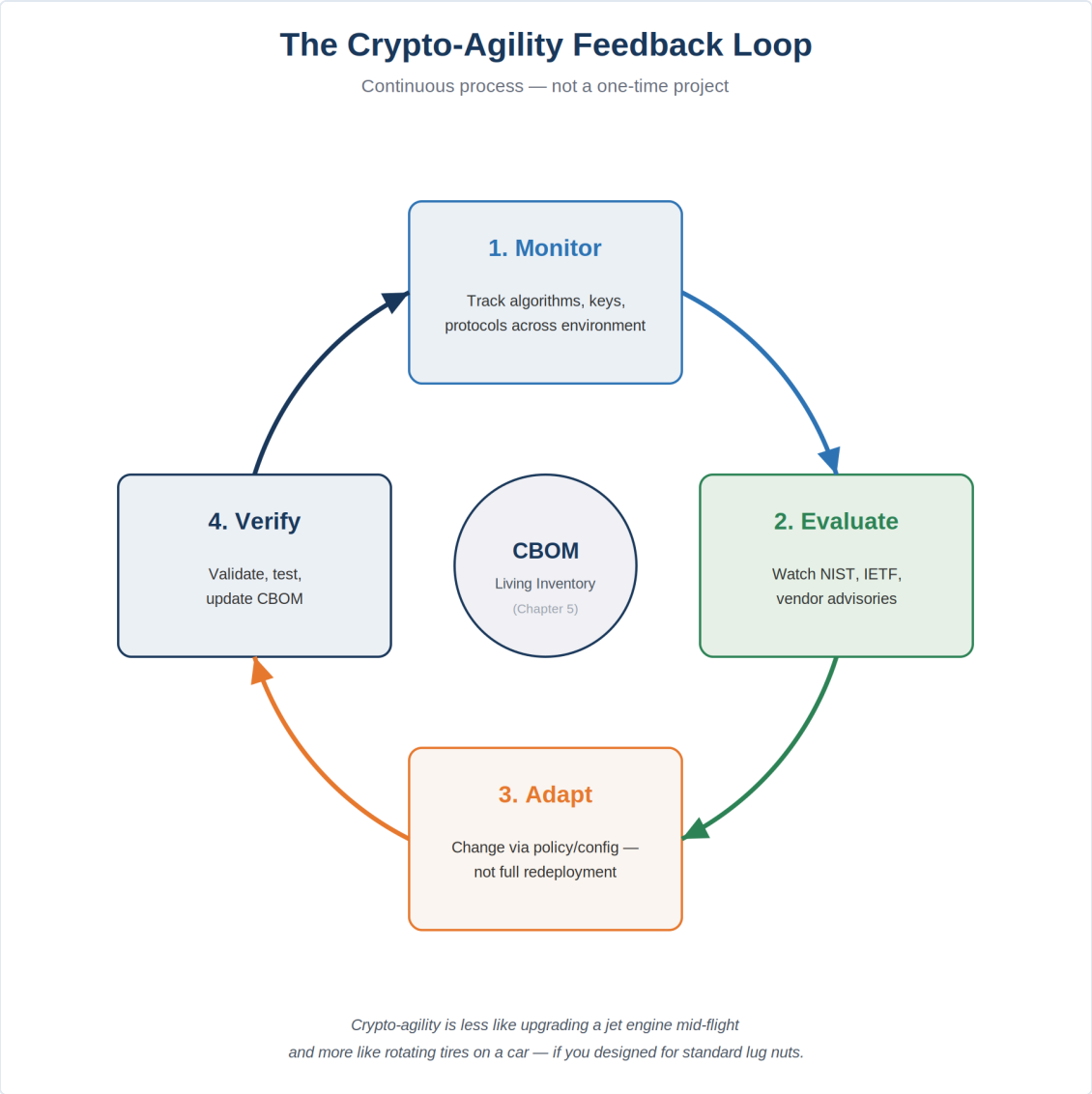

Figure 9.1 — The Crypto-Agility Feedback Loop

1. Monitor. Continuously track which algorithms, key sizes, and protocols are in use across your environment. Your CBOM from Chapter 5 is a living document, not a one-time deliverable. Update it as systems change.[10]

2. Evaluate. Watch for NIST advisories, IETF draft updates, and cryptanalytic research. If a new attack weakens ML-KEM or ML-DSA, your team needs to assess the impact within days, not months. Subscribe to the NIST PQC mailing list, IETF TLS working group updates, and your vendors' security advisories.

3. Adapt. When a change is needed—a new algorithm, a deprecated parameter, a configuration update—your crypto-agile architecture should allow it through policy and configuration changes rather than full application redeployments. This is the payoff for the modular design principles established in Chapter 6.

4. Verify. After any change, validate that the new configuration is operating correctly: handshakes complete, performance is within bounds, interoperability is maintained. Then update your CBOM and close the loop.

> **PLAIN-LANGUAGE SIDEBAR** Crypto-agility in practice is less like upgrading a jet engine mid-flight and more like rotating tires on a car. If you designed the system with standard lug nuts (modular interfaces), the swap is straightforward maintenance. If every tire is welded on (hard-coded algorithms), every change is a crisis. The architectural decisions you make during migration determine which experience your team has for the next decade.
> **F5 PERSPECTIVE Day-2 PQC Operations with BIG-IP** BIG-IP's role in Day-2 PQC operations extends naturally from its position as the TLS termination and visibility point: **Observability:** BIG-IP telemetry (via Application Study Tool, F5 Insight, or iRules logging) can surface per-VIP cipher suite negotiation, handshake latency distributions, and certificate chain sizes. This data feeds directly into the monitoring table above—your PQC performance baseline lives on BIG-IP. **Algorithm policy enforcement:** SSL/TLS profiles on BIG-IP control which cipher suites and key exchange groups are offered to clients. Updating the allowed set to include (or require) X25519MLKEM768 is a profile change, not a code deployment. When the time comes to drop classical-only key exchange, it's the same profile change in reverse. **Certificate rotation:** As PQC certificates enter production, BIG-IP's certificate management capabilities handle the larger chain sizes. For automated rotation today, F5 publishes Kojot ACME, an open-source ACMEv2 client utility that runs on BIG-IP and supports HTTP-01 and DNS-01 validation, wildcard certificates, HSM/FIPS key preservation, HA deployments, and OCSP monitoring. On the NGINX side of F5's portfolio, the ngx_http_acme_module ships natively in NGINX Open Source and NGINX Plus, enabling certificate issuance and renewal directly through NGINX configuration directives without external clients. Readers should expect F5's native ACME footprint across the platform to continue expanding as 47-day certificate validity approaches. **Crypto-agility at the edge:** BIG-IP is the crypto-agility enforcement point for your internet edge. When NIST publishes an advisory, when a new IETF draft changes a code point, or when your CCOE decides to adjust PQC policy, the change is implemented on a handful of BIG-IP profiles rather than hundreds of application servers.

Closing the Loop

This book has followed a deliberate arc: why the quantum threat demands action (Chapters 1–2), what replaces the vulnerable algorithms (Chapters 3–4), and how to discover, plan, deploy, and operate the migration (Chapters 5–9). Along the way, we've been honest about what's solved (hybrid key exchange), what's in progress (certificate authentication), and what's genuinely hard (DNSSEC, long-lived evidence, and the sheer organizational challenge of touching every cryptographic system in an enterprise).

The PQC migration is the most significant cryptographic transition since the move from DES to AES—and arguably larger, because it touches public-key infrastructure in ways the symmetric transition never did. But it's also a transition with clear standards, strong community momentum, and years of preparation time for organizations that start now.

The Appendices that follow provide quick-reference tools for your team: a glossary, an algorithm cheat sheet, a compliance checklist, and a vendor PQC readiness assessment template. Keep this book within arm's reach.

The migration is a multi-year journey, and the operational practices in this chapter will be your daily companion long after the deployment celebrations are over.

Notes

The following sources support specific claims made in Chapter 9. Full bibliographic entries appear in the Bibliography.

1. The "harvest-now, forge-later" concept for digital signatures is analogous to the "harvest-now, decrypt-later" threat for encryption. An adversary captures classically signed artifacts today and uses a future quantum computer to forge signatures, retroactively compromising integrity and non-repudiation. See: "Post-Quantum-Resilient Audit Evidence for Long-Lived Regulated Systems," arXiv:2512.00110 (February 2026).

2. Three migration patterns for legacy evidence: hybrid signatures for new artifacts, re-signing legacy artifacts with PQC keys, and Merkle root anchoring for batch re-authentication. Formalized in arXiv:2512.00110 with security proofs (Q-Audit Integrity, Q-Non-Equivocation, Q-Binding) and benchmarks on commodity hardware.

3. CNSA 2.0 timeline: software and firmware signing should prefer PQC by 2025, with exclusive PQC required by 2030. NSA guidance explicitly encourages dual-signing firmware immediately. NIST SP 800-208 approves stateful hash-based signatures (LMS, XMSS) specifically for code signing and secure boot use cases.

4. CA/Browser Forum Ballot SC-081v3. Certificate maximum validity shrinks to 200 days (March 2026), 100 days (March 2027), 47 days (March 2029). ACME-based automation becomes operationally essential at these renewal velocities.

5. The ACME protocol is specified in RFC 8555 (Barnes, R., Hoffman-Andrews, J., McCarney, D., Kasten, J. "Automatic Certificate Management Environment (ACME)." IETF, March 2019). ACME v2 API was released in March 2018 and standardized in RFC 8555. Native ACME support in web servers and reverse proxies: Apache HTTP Server since 2.4.30; Caddy and Traefik as default behavior; NGINX ngx_http_acme_module preview released August 12, 2025 (see NGINX Community Blog, "NGINX Introduces Native Support for ACME Protocol"). F5 provides an ACMEv2 client for BIG-IP via the open-source Kojot ACME project on GitHub (f5devcentral/kojot-acme, MIT licensed), supporting HTTP-01, DNS-01, wildcard certificates, HSM/FIPS key preservation, HA, and OCSP.

6. HSM readiness questions from Chapter 6: (1) Does the HSM support ML-KEM/ML-DSA? (2) Does the API map to your PKI's calling conventions? (3) Can the HSM back up and replicate PQC keys in HA configurations? (4) What are the performance benchmarks for PQC operations? (5) Is the implementation FIPS 140-3 validated? Thales, Entrust, and Marvell/Cavium are in various stages of PQC support rollout.

7. Performance baselining: capture TLS handshake latency at p50, p95, and p99 before enabling PQC. Use existing APM tools (Prometheus/Grafana, F5 AST/Insight, Datadog, New Relic) to establish clean baselines. Post-migration, compare same metrics on same paths to isolate PQC-induced changes from unrelated variables.

8. Supply chain PQC risk: a third-party provider using RSA-2048 creates a quantum-vulnerable link regardless of your internal migration status. CISA and NIST NCCoE guidance recommends extending PQC assessments to

third-party vendors. Multiple sources (DigiCert, Palo Alto Networks) emphasize that “your PQC migration is only as strong as your weakest vendor.”

9. USDA Acquisition Regulation (AGAR), revised September 2025, contains explicit PQC procurement language: products in CISA-listed categories must support PQC. Additional agencies are expected to adopt similar language through agency-specific acquisition rules. See: postquantum.com, “The Complete US Post-Quantum Cryptography Regulatory Framework in 2026.”

10. NIST CSWP 39 (Cybersecurity White Paper): “Crypto-Agility Considerations for Migrating to Post-Quantum Cryptographic Algorithms.” Recommends maintaining a living cryptographic inventory (CBOM) as part of continuous security monitoring. Integration with NIST CSF v2.0 functions: Identify, Protect, Detect, Respond, Recover.

Next: Appendices — Glossary, Algorithm Cheat Sheet, Compliance Checklist, Vendor PQC Readiness Template, and Bibliography

Appendix D

Quantum Risk Scoring Methodology

This appendix provides a reusable, quantitative risk scoring methodology for assessing the quantum threat to specific systems and applications. It is adapted from the TNO/AIVD quantum risk methodology and designed to plug into your organization's existing risk management framework (NIST RMF, ISO 27005, or agency-specific processes).

The quantum risk score combines three factors: **algorithm weakness** (how vulnerable is the cryptography?), **impact severity** (what happens if it's broken?), and **migration difficulty** (how hard is it to fix?). Each is scored independently, then combined into an overall 0–4 risk score.

Factor 1: Algorithm Weakness Score (0–3)

Score	Classification	Examples
0	Quantum-safe. No migration needed.	AES-256, SHA-384, SHA-512, ML-KEM, ML-DSA, SLH-DSA
1	Theoretically weakened by quantum, but not practically broken. Future attention needed.	AES-128 (Grover's halves effective strength), SHA-256 (collision resistance reduced but still adequate), 3DES
2	Broken by quantum computer running Shor's algorithm. Migration required within NIST timeline.	RSA-2048+, ECDSA P-256/P-384, ECDH, DH, DSA — all quantum-vulnerable asymmetric algorithms
3	Broken by quantum AND already weakened classically. Immediate migration regardless of quantum timeline.	RSA-1024, SHA-1 signatures, DES, RC4, MD5 — deprecated algorithms still found in legacy systems

Factor 2: Impact Severity Score (0–4)

Assess the impact if the cryptographic protection on this system were completely compromised. Consider both the security function (confidentiality, integrity, authentication) and the data sensitivity.

Score	Impact Level	Description
0	Negligible	Public data. No confidentiality, integrity, or authentication requirement.
1	Low	Internal data with limited sensitivity. Breach causes minor operational disruption. Short data lifespan (<2 years).
2	Moderate	Business-sensitive data. Breach causes financial loss, reputational damage, or regulatory penalty. Data lifespan 2–10 years.
3	High	Classified data, PII at scale, critical infrastructure control systems, or high-value IP. Data lifespan 10–25 years. HNDL risk is significant.
4	Critical	National security data, weapons systems, intelligence sources and methods, long-lived infrastructure (satellites, OT/ICS). Data lifespan 25+ years. HNDL risk is severe and immediate.

Factor 3: Migration Difficulty Score (0–4)

Estimate the time, effort, and complexity required to migrate this system to PQC. Higher scores indicate harder migrations that need earlier starts.

Score	Difficulty	Description
0	Trivial	Already quantum-safe or requires only a configuration change (e.g., enabling hybrid TLS on a modern load balancer).
1	Low	Software update or library upgrade. Vendor provides PQC-ready release. Standard testing and rollout cycle.
2	Moderate	Requires coordinated changes across multiple systems (e.g., PKI hierarchy + application servers + clients). Interoperability testing needed.
3	High	Custom protocols, embedded/OT systems, vendor dependencies with no PQC roadmap, or HSM replacements required. Multi-year effort.
4	Extreme	Cannot be migrated in place. Requires full system replacement, contract renegotiation, or physical hardware swap (satellites, deployed military systems, legacy SCADA).

Computing the Overall Quantum Risk Score (0–4)

The overall quantum risk score is derived by combining the three factors. This is not a simple average—algorithm weakness gates the assessment, while impact and difficulty determine urgency:

Step 1: If Algorithm Weakness = 0, the system is quantum-safe. Overall Risk = 0 regardless of other factors. No further assessment needed.

Step 2: If Algorithm Weakness = 3, the system has classical vulnerabilities. Overall Risk = 4 (Critical). Migrate immediately regardless of quantum timeline.

Step 3: For Algorithm Weakness = 1 or 2, compute:

Overall Risk = min(4, round((Impact + Difficulty) / 2))

Systems with Algorithm Weakness = 1 (symmetric only) generally score lower because the threat is theoretical. Multiply the computed score by 0.5 and round up for these systems.

Risk Score Action Guide

Score	Priority	Recommended Action
0	None	System is quantum-safe. Document in CBOM and monitor for future changes.
1	Low (P4)	Monitor. Include in long-term migration plan. No immediate action required.
2	Medium (P2–P3)	Plan migration within 2–4 years. Include in Phase 2–3 of roadmap (Chapter 6). Begin vendor engagement.
3	High (P1)	Migrate within 1–2 years. Prioritize in Phase 1. Enable hybrid mode immediately where possible.
4	Critical (P0)	Migrate immediately. Escalate to CCOE and executive leadership. This system is actively at risk from classical and/or quantum threats.

Worked Examples

The following three scenarios illustrate the scoring methodology in practice. Your CBOM will produce dozens to hundreds of such assessments; these examples establish the reasoning pattern.

Example 1: Internet-facing web application, RSA-2048 TLS, protecting PII

A public-facing HR portal terminates TLS with an RSA-2048 certificate and ECDHE key exchange. The portal handles employee records subject to GDPR and state privacy laws, with a 7-year retention requirement.

Algorithm Weakness = 2. Asymmetric cryptography (RSA signatures, ECDHE key exchange) is Shor's-vulnerable.

Impact Severity = 4. PII exposure triggers regulatory notification, customer trust damage, and multi-year legal exposure. Internet-facing means HNDL interception is assumed.

Migration Difficulty = 2. Standard TLS 1.3 migration path: enable hybrid X25519MLKEM768 at the terminator. Application stack unchanged.

Overall Risk = min(4, round((4 + 2) / 2)) = **3**, but HNDL exposure on PII with 7+ year sensitivity lifetime pulls this to P0 per Chapter 6's priority matrix. Enable hybrid TLS immediately.

Example 2: Internal east-west TLS 1.2 between microservices, low-sensitivity operational data

A service mesh uses TLS 1.2 with ECDHE between internal microservices handling non-sensitive operational telemetry. Traffic is isolated to the internal VPC; no customer data transits this leg.

Algorithm Weakness = 2. Same asymmetric exposure as Example 1.

Impact Severity = 2. Non-sensitive data, short retention, limited interception surface.

Migration Difficulty = 3. Service mesh coordination across dozens of microservices and sidecar proxies; coordination cost dominates the effort.

Overall Risk = min(4, round((2 + 3) / 2)) = **3**. Place in P2–P3 per priority matrix. Schedule within Phase 3 of the migration roadmap; coordinate upgrades with normal service mesh refresh cycles.

Example 3: Legacy code-signing, SHA-1 with RSA-2048, embedded firmware

An industrial control system vendor signs firmware updates with a SHA-1 + RSA-2048 signing chain. The certificates were issued in 2018 and the signing infrastructure predates the organization's migration to SHA-2.

Algorithm Weakness = 3. SHA-1 is classically broken (collision attacks demonstrated). This gates the overall score regardless of the other factors.

Impact Severity and Migration Difficulty are not computed; Step 2 of the scoring methodology sets Overall Risk = **4 (Critical)**. Migrate immediately. This scenario illustrates the scoring framework's gating rule: systems with classical vulnerabilities jump to Critical independent of quantum timeline considerations. PQC planning does not displace classical hygiene.

Record the scores for each system in your CBOM (Chapter 5) alongside the cryptographic inventory. The risk scores feed directly into the priority matrix in Chapter 6 and help justify budget allocation to leadership.

Appendix E

APPENDICES & REFERENCE

PQC Migration Maturity Assessment

This self-assessment tool helps organizations evaluate their current readiness for the PQC migration across eight key dimensions. It is adapted from the TNO/AIVD PQC growth model and designed for use by your CCOE (Chapter 6) as a quarterly or semi-annual checkpoint.

For each dimension, identify the stage (1–5) that best describes your organization's current state. The assessment highlights which areas need attention and provides a structured framework for tracking progress over time.

How to Use This Scorecard

- **Assess each dimension independently.** Your organization may be at Stage 4 in Awareness but Stage 1 in PQC Availability—that's normal.
- **Focus on the lowest-scoring dimensions first.** These are your bottlenecks. Improvements in one area often unblock progress in others.
- **Reassess quarterly.** The PQC landscape is evolving rapidly. What was "not available" six months ago may now be generally available.

Dimension 1: Awareness & Leadership

Stage	Description
1	No awareness of quantum threat within leadership or technical teams.
2	Individual contributors aware; no executive sponsorship or budget allocation.
3	CISO/CTO briefed. Quantum threat included in risk register. Initial budget discussion underway.
4	Executive sponsor assigned. Dedicated budget approved. PQC migration in organizational roadmap.
5	Board-level reporting. PQC integrated into enterprise risk management. Regular progress reviews.

Dimension 2: Governance & Organization

Stage	Description
1	No dedicated team or ownership for cryptographic migration.
2	Ad hoc responsibility assigned to existing security team as side duty.
3	CCOE established (Chapter 6) with defined roles. Migration plan drafted.
4	CCOE operating with cross-functional representation. Phased roadmap approved and in execution.
5	CCOE integrated into permanent security operations. Crypto-agility policies enforced across the org.

Dimension 3: Cryptographic Discovery & Inventory

Stage	Description
1	No inventory of cryptographic assets. Unknown what algorithms are in use.
2	Partial manual inventory of high-priority systems. No CBOM format established.
3	Automated discovery running on priority systems. CBOM format defined. Quantum risk scores assigned.
4	Comprehensive CBOM covering all systems. Continuously updated. Integrated with risk management.
5	Real-time cryptographic observability. CBOM auto-updated from network telemetry. Drift detection alerts.

Dimensions 4–8: Quick-Reference Stages

For the remaining five dimensions, assess your organization against the same 1–5 scale:

Dimension	Stage 1–2	Stage 3	Stage 4	Stage 5
4. Policies & Compliance	No PQC policy. Crypto policies outdated.	PQC policy drafted. Regulatory gaps identified.	PQC procurement requirements in contracts.	Full compliance with NIST/CNSA 2.0 timelines.
5. PQC Availability	No PQC-capable products in environment.	Vendor PQC roadmaps collected. Lab testing begun.	PQC-capable products deployed in pilot/hybrid.	PQC standard across production. Pure PQC for new systems.
6. Hybrid Deployment	No hybrid crypto deployed.	Hybrid TLS enabled on edge/pilot systems.	Hybrid across all internet-facing and priority internal systems.	Transitioning from hybrid to pure PQC per NIST timeline.
7. Crypto-Agility	Hard-coded algorithms. No abstraction layer.	Crypto-agility policy defined. New systems designed for agility.	Algorithm changes via config, not code. CBOM feedback loop active.	Proven agility: algorithm swap completed in <30 days across environment.
8. Knowledge & Skills	No PQC training. Reliance on external consultants.	Core team trained. Tabletop exercises conducted.	Role-specific training (Ch9 matrix) complete. Internal PQC expertise.	Organization self-sufficient. Contributing to community/standards.

Interpreting Your Score

Sum your scores across all 8 dimensions (max 40). This gives an overall maturity snapshot:

- **8–15 (Early Stage):** You're at the beginning. Focus on awareness, establishing the CCOE, and starting cryptographic discovery. Chapters 1–5 are your priority reading.
- **16–24 (In Progress):** You have a foundation. Focus on completing discovery, deploying hybrid mode, and engaging vendors. Chapters 6–7 are your next steps.
- **25–32 (Advanced):** You're executing. Focus on protocol deep dives (Chapter 8), Day-2 operations (Chapter 9), and closing gaps in the lowest-scoring dimensions.
- **33–40 (Mature):** You're a leader. Focus on transitioning from hybrid to pure PQC, contributing to standards, and mentoring your supply chain partners.

Track scores over time. The goal is steady progress across all dimensions, not perfection in any single one. A balanced approach reduces the risk of being caught with a critical gap when a quantum milestone arrives.

> **PLAIN-LANGUAGE SIDEBAR** This scorecard is a conversation starter, not a compliance checklist. Print it, bring it to your next security review, and have each stakeholder independently score the eight dimensions. Where scores differ by 2 or more stages, you've found a gap in shared understanding—and that's the most valuable outcome of the exercise.

Appendix A

Glossary

Quick-reference definitions for terms used throughout this book. Terms are listed alphabetically.

AES (Advanced Encryption Standard) Symmetric block cipher standardized by NIST. AES-128/192/256 refers to the key length in bits. Considered quantum-safe at 256-bit key lengths.

Algorithm Weakness Score A 0–3 rating of how vulnerable a cryptographic algorithm is to quantum attack (Appendix D).

AMS (Acquisition Management System) FAA lifecycle framework for planning, analyzing, acquiring, deploying, and sustaining systems. Six phases per FAA FAST policy: Service Analysis & Strategic Planning, Concept & Requirements Definition, Initial Investment Analysis, Final Investment Analysis, Solution Implementation, and In-Service Management. Disposal and service life extension are managed within In-Service Management rather than as a distinct phase. See Appendix G for PQC migration crosswalk.

ATO (Authorization to Operate) Formal management decision authorizing an information system to operate at an acceptable level of residual risk (NIST SP 800-37 Rev 2). Typically three-year validity for federal systems, with continuous authorization models increasingly encouraged.

Bridge Architecture Deployment pattern where the TLS termination point (e.g., BIG-IP) negotiates hybrid PQC on the front side and classical TLS to backends. Hardens the internet-facing leg first.

CA/Browser Forum Industry consortium that sets baseline requirements for publicly trusted TLS certificates. Ballot SC-081v3 sets certificate validity shrinking to 47 days by 2029.

CBOM (Cryptographic Bill of Materials) A structured inventory documenting every cryptographic algorithm, key, certificate, and protocol in use across an organization's systems.

CCOE (Cryptographic Center of Excellence) A cross-functional team (Chapter 6) responsible for steering the PQC migration across network, PKI, application, and governance domains.

CNSA 2.0 (Commercial National Security Algorithm Suite 2.0) NSA's updated algorithm guidance for National Security Systems. Specifies ML-KEM-1024 and ML-DSA-87. Full compliance required by 2030–2035.

CNSSI 4009 Committee on National Security Systems Instruction 4009, the CNSS Glossary. Authoritative source for federal information security terminology across U.S. National Security Systems.

CRQC (Cryptographically Relevant Quantum Computer) A quantum computer powerful enough to run Shor's algorithm against deployed cryptographic key sizes. Does not yet exist; timeline estimates range from 10–20+ years.

Crypto-Agility The architectural capability to swap, update, or replace cryptographic algorithms without redesigning applications or protocols. A core design principle for PQC migration.

Cryptographic Proxy Layer Structural term for the Bridge Architecture (Chapter 7)—a dedicated enforcement point (TLS terminator, ADC, reverse proxy) that performs cryptographic upgrade on behalf of down-

stream systems that are not yet PQC-capable. Commonly used in federal and enterprise architecture documentation as a synonym for "bridge architecture."

DoW CIO PQC Directorate Department of War organization established by the November 18, 2025 DoW CIO memorandum *Preparing for Migration to Post Quantum Cryptography*. Issues two new authorizations every DoW Component must obtain before any PQC engagement: **cryptographic intake approval** (before testing, evaluating, piloting, investing in, or acquiring any PQC-enabling or PQC-related technology) and **cryptographic deployment approval** (before deployment, informed by NIST, NSA, and IC certification outcomes). These approvals layer on top of, not replace, FIPS 140-3, NIAP Common Criteria, and NSA CSfC. Led by Dr. Britta Hale. Chapter 4.

EO 14144 / EO 14306 U.S. Executive Orders directing federal agencies to inventory quantum-vulnerable systems and begin PQC migration. EO 14306 rescinded several prior orders but preserved PQC mandates.

FIPS 203 (ML-KEM) NIST standard for Module-Lattice-Based Key Encapsulation Mechanism. Replaces ECDH/DH for key exchange. Three parameter sets: ML-KEM-512/768/1024.

FIPS 204 (ML-DSA) NIST standard for Module-Lattice-Based Digital Signature Algorithm. Replaces RSA/ECDSA for signatures. Three parameter sets: ML-DSA-44/65/87.

FIPS 205 (SLH-DSA) NIST standard for Stateless Hash-Based Digital Signature Algorithm (formerly SPHINCS+). Conservative backup to ML-DSA, using only hash functions. Larger signatures.

FIPS 206 (FN-DSA) NIST draft standard for FFT-over-NTRU-Lattice Digital Signature Algorithm (formerly Falcon). Compact 666-byte signatures at Level 1; complex floating-point implementation.

FISMA (Federal Information Security Modernization Act) 2014 law updating the 2002 Federal Information Security Management Act. Requires federal agencies to implement information security programs; mandates RMF compliance through OMB oversight.

Grover's Algorithm Quantum search algorithm that provides quadratic speedup against symmetric cryptography. Halves effective key strength: AES-128 → 64-bit equivalent. AES-256 remains safe.

Harvest-Now, Decrypt-Later (HNDL) Attack strategy where adversaries capture encrypted traffic today for decryption by a future quantum computer. The primary driver of urgency for PQC key exchange migration.

HQC (Hamming Quasi-Cyclic) Code-based backup KEM selected by NIST in March 2025. Provides algorithmic diversity from lattice-based ML-KEM. Standard expected ~2027.

HSM (Hardware Security Module) Dedicated hardware device for secure key generation, storage, and cryptographic operations. PQC migration requires HSM firmware/hardware supporting ML-KEM/ML-DSA.

Hybrid Mode Running a classical algorithm alongside a PQC algorithm so that the system is secure as long as at least one holds. Example: X25519MLKEM768 for TLS key exchange.

IW10 / IW20 TCP initial congestion window set to 10 or 20 segments. IW10 (≈14.6 KB) is the default; IW20 (≈29 KB) accommodates most PQC certificate chains in a single flight.

Merkle Tree Certificates (MTCs) Google/Cloudflare initiative replacing per-certificate PQC signatures with compact Merkle inclusion proofs. Reduces TLS authentication data from ~15 KB to ~736 bytes.

ML-DSA (Module-Lattice Digital Signature Algorithm) See FIPS 204. The primary PQC signature algorithm. ML-DSA-65 is the general-purpose recommendation.

ML-KEM (Module-Lattice Key Encapsulation Mechanism) See FIPS 203. The primary PQC key exchange algorithm. ML-KEM-768 is the general-purpose recommendation.

mTLS (Mutual TLS) TLS configuration where both client and server authenticate with certificates. PQC doubles the certificate size overhead (both directions send PQC chains).

NIST IR 8547 NIST guidance on transitioning to PQC. Deprecates 112-bit classical algorithms after 2030; disallows all quantum-vulnerable public-key algorithms after 2035.

NSM-10 (National Security Memorandum 10) 2022 White House directive requiring federal agencies to migrate to PQC "as much as is feasible by 2035."

POA&M (Plan of Action and Milestones) Document identifying tasks needed to remediate known security weaknesses, milestones for completion, and resource requirements. Core output of the RMF Monitor step (NIST SP 800-37 Rev 2).

PPK (Post-Quantum Pre-Shared Key) RFC 8784 mechanism for layering a quantum-resistant symmetric secret onto IKEv2 IPsec key derivation as an interim PQC measure.

Q-Day The hypothetical date when a CRQC first breaks deployed public-key cryptography. Not a single event—different algorithms may fall at different times.

QKD (Quantum Key Distribution) Hardware-based key distribution using quantum physics. NSA, NCSC, ANSSI, and BSI recommend PQC over QKD for most use cases due to cost, range, and scalability limitations.

RMF (Risk Management Framework) NIST SP 800-37 Rev 2 seven-step process—Prepare, Categorize, Select, Implement, Assess, Authorize, and Monitor—for managing security and privacy risk across federal information systems.

SCRM (Supply Chain Risk Management) Discipline for identifying and mitigating risks introduced by suppliers, subcontractors, and third-party components (NIST SP 800-161 Rev 1). Extended in Chapter 9 to address PQC vendor readiness.

Shor's Algorithm Quantum algorithm that efficiently factors large integers and computes discrete logarithms, breaking RSA, ECDSA, ECDH, DH, and DSA.

SSP (System Security Plan) Document describing how required security controls are implemented or planned for an information system (NIST SP 800-18, SP 800-37 Rev 2). Core component of the authorization package submitted for ATO.

X25519MLKEM768 The dominant hybrid TLS key exchange combining classical X25519 with ML-KEM-768. Client key share: 1,216 bytes. Specified in IETF draft-ietf-tls-ecdhe-mlkem.

Appendix B

APPENDICES & REFERENCE

Algorithm Cheat Sheet

One-page reference for algorithm selection. Sizes are approximate and include DER/X.509 encoding overhead where applicable.

Key Exchange / Encapsulation (replacing ECDH, DH, RSA key transport)

Algorithm	Public Key	Ciphertext	Security Level	Notes
ML-KEM-512	800 bytes	768 bytes	Level 1 (AES-128)	Fastest/smallest. Not recommended for high-value.
ML-KEM-768	1,184 bytes	1,088 bytes	Level 3 (AES-192)	**★ RECOMMENDED default for most use cases.**
ML-KEM-1024	1,568 bytes	1,568 bytes	Level 5 (AES-256)	Required for CNSA 2.0 / NSS.
HQC (expected ~2027)	~2,249 bytes	~4,481 bytes	Level 1/3	Code-based backup KEM. Algorithmic diversity from ML-KEM.

Digital Signatures (replacing RSA, ECDSA, EdDSA, DSA)

Algorithm	Public Key	Signature	Security Level	Notes
ML-DSA-44	1,312 bytes	2,420 bytes	Level 2 (AES-128)	Smallest ML-DSA. Suitable where Level 3 not required.
ML-DSA-65	1,952 bytes	3,309 bytes	Level 3 (AES-192)	**★ RECOMMENDED default for general-purpose signing.**
ML-DSA-87	2,592 bytes	4,627 bytes	Level 5 (AES-256)	Required for CNSA 2.0 / NSS root CAs.
SLH-DSA-SHA2-128s	32 bytes	7,856 bytes	Level 1 (AES-128)	Hash-based. Conservative backup. Very large signatures.
FN-DSA-512 (draft)	897 bytes	666 bytes	Level 1 (AES-128)	Most compact signatures. Complex implementation (floating-point). Best for CA-level signing.
FN-DSA-1024 (draft)	1,793 bytes	1,280 bytes	Level 5 (AES-256)	Level 5 Falcon variant. Same implementation complexity caveats.

The 80/20 Rule

For 80% of enterprise migration scenarios, two algorithms cover your needs: **ML-KEM-768 for key exchange + ML-DSA-65 for signatures.** Start there. Optimize later.

Appendix C

APPENDICES & REFERENCE

PQC Compliance Checklist

A consolidated timeline and action checklist for PQC-related mandates. Check off items as your organization completes them.

Immediate Actions (Now)

☐	Action	Reference
☐	**Cryptographic asset discovery**	Inventory all algorithms, keys, certs, and protocols. Produce CBOM. (Ch5, NIST SP 1800-38B)
☐	**Quantum risk assessment**	Score all systems using Appendix D methodology. Identify P0 systems.
☐	**Establish CCOE**	Cross-functional team per Ch6 model. Assign executive sponsor.
☐	**Update cryptographic policies**	Incorporate PQC requirements into procurement, development, and security policies.
☐	**Enable hybrid TLS key exchange**	X25519MLKEM768 on internet-facing load balancers/CDN. (Ch7 bridge architecture)
☐	**Verify SSH key exchange**	Confirm OpenSSH 10.0+ default (mlkem768x25519). Update if needed.
☐	**Begin dual-signing firmware**	Sign new firmware/SBOMs with both classical + ML-DSA. (CNSA 2.0 “prefer by 2025”)

2026–2027 Actions

☐	Action	Reference
☐	**Assess HSM PQC readiness**	Five questions from Ch6. Plan firmware upgrades or replacements.
☐	**Pilot PQC certificates**	Test ML-DSA certificates in non-production. Measure handshake performance. (Ch8)
☐	**Engage vendors on PQC roadmaps**	Collect PQC timelines from all critical vendors. (Appendix F template)
☐	**Evaluate Merkle Tree Certificates**	Track Chrome/Cloudflare MTC pilot (Phase 1–2). Plan for CQRS if web-facing.
☐	**Increase TCP initcwnd to 20**	On internet-facing VIPs/load balancers to accommodate PQC cert chains. (Ch8)
☐	**Deploy IPsec PPKs**	RFC 8784 post-quantum pre-shared keys on priority VPN tunnels. (Ch7)
☐	**Automate certificate lifecycle**	ACME or vendor CLM for 200-day cert validity (CA/B Forum, March 2026).

2028–2030 Actions

☐	Action	Reference
☐	**Complete PKI hierarchy migration**	New root/intermediate CAs with ML-DSA. Begin issuing PQC leaf certificates.
☐	**Migrate IPsec to native ML-KEM**	Replace PPK stopgap with CNSA 2.0 IPsec profile (ML-KEM-1024).
☐	**Re-sign legacy evidence**	Re-sign critical audit logs, contracts, and firmware archives with PQC. (Ch9 Pattern 2/3)

☐	Action	Reference
☐	**Automate 47-day cert renewal**	Prepare for CA/B Forum 47-day maximum validity by March 2029.
☐	**Transition high-confidence systems to pure PQC**	Drop classical-only where ML-KEM/ML-DSA have 6+ years post-standardization scrutiny.

2030–2035 Actions

☐	Action	Reference
☐	**NIST deprecation deadline (2030)**	All 112-bit classical algorithms deprecated. No new deployments. (NIST IR 8547)
☐	**CNSA 2.0 full compliance (2030)**	NSS: exclusive use of ML-KEM-1024 / ML-DSA-87 for networking.
☐	**Complete hybrid → pure PQC transition**	Remove classical component from hybrid deployments where no longer needed.
☐	**NIST disallow deadline (2035)**	All quantum-vulnerable public-key algorithms disallowed. NSM-10 full compliance.
☐	**Validate crypto-agility**	Confirm ability to swap algorithms within 30 days across the environment. (Appendix E Dim 7)

Federal Sector Additions

Federal agencies and federal service integrators should apply the following items in addition to the timeline-based actions above. These align PQC migration activities with the existing Risk Management Framework (NIST SP 800-37 Rev 2) rather than creating a parallel compliance track.

☐	Action	Reference
☐	**Include PQC in System Security Plans (SSPs)**	Document PQC control selection, implementation timeline, and residual risk per NIST SP 800-37 Rev 2.
☐	**Track PQC migration in POA&Ms**	Record quantum-vulnerable systems and their migration milestones in the Plan of Action and Milestones for continuous monitoring.
☐	**Submit annual quantum-vulnerable IT system inventory**	Required by NSM-10 for federal agencies. (Ch 5 Note 1)
☐	**M-23-02 cryptographic inventory reporting**	FCEB agencies submit annual cryptographic inventory to CISA per OMB Memorandum M-23-02. (Ch 5 Note 1)
☐	**FedRAMP continuous monitoring for cloud offerings**	Include PQC posture and migration progress in FedRAMP ConMon deliverables.
☐	**Reauthorize systems after PQC migration**	Treat hybrid/PQC deployment as a significant change triggering ATO reauthorization per NIST SP 800-37 Rev 2 Monitor step.
☐	**Embed PQC in procurement language**	Apply PQC readiness requirements to acquisition clauses. USDA/AGAR provides a model. (Ch 9 Note 9)

Appendix F

APPENDICES & REFERENCE

Vendor PQC Readiness Assessment Template

Use this template when evaluating vendors, suppliers, and third-party service providers. Distribute to procurement, security architecture, and CCOE team members.

Vendor Information

Vendor Name:	
Product / Service:	
Assessment Date:	
Assessed By:	
Contract Renewal Date:	

Algorithm & Protocol Support

Capability	Supported	Planned	No Plans
ML-KEM-768 / ML-KEM-1024 key exchange	☐	☐	☐
ML-DSA-65 / ML-DSA-87 digital signatures	☐	☐	☐
SLH-DSA (hash-based backup signatures)	☐	☐	☐
X25519MLKEM768 hybrid TLS 1.3	☐	☐	☐
TLS 1.3 certificate compression (RFC 8879)	☐	☐	☐
Hybrid certificates (dual classical + PQC)	☐	☐	☐
IPsec IKEv2 with ML-KEM or PPK (RFC 8784)	☐	☐	☐
SSH mlkem768x25519 key exchange	☐	☐	☐
FIPS 140-3 validation for PQC algorithms	☐	☐	☐
CBOM (Cryptographic Bill of Materials) disclosure	☐	☐	☐

Migration Readiness Questions

Question	Vendor Response
What is your published PQC migration roadmap and timeline?	
Can customers configure algorithm selection, or does it require a vendor release?	
How quickly can you swap to an alternative algorithm if one is deprecated?	
Do you support hybrid mode (classical + PQC simultaneously)?	
What is the expected performance impact of enabling PQC?	

Question	Vendor Response
Are your PQC implementations based on NIST-validated libraries?	
Can you provide a CBOM for your product?	
What HSM vendors/firmware versions are supported for PQC operations?	
What is your testing/certification timeline for FIPS 140-3 PQC validation?	
How does your product handle PQC certificate chain sizes (>15 KB)?	

Overall Assessment

PQC Readiness Rating:	**□ Ready □ In Progress □ Not Ready □ No Plans**
Risk to Our Migration:	□ Low □ Medium □ High □ Critical
Recommended Action:	
Follow-up Date:	

Appendix G

APPENDICES & REFERENCE

Federal Framework Crosswalk

Federal agencies operate under multiple overlapping cybersecurity and acquisition frameworks. This appendix maps the book's five-phase PQC migration model (Chapter 6) against the four frameworks most commonly encountered by federal and DoD readers: the NIST Risk Management Framework, the FAA Acquisition Management System, FedRAMP, and the DoD Risk Management Framework. The intent is not to replace these frameworks' own guidance but to show PQC program managers where their migration work maps onto existing compliance artifacts—SSPs, POA&Ms, ConMon submissions, Investment Analysis reports, and ATO packages.

Use this crosswalk when building a PQC program charter, responding to RFIs or audit inquiries, or aligning budget requests with existing framework deliverables. Every row points to work your organization likely already performs; what changes is the cryptographic content of that work.

NIST Risk Management Framework (SP 800-37 Rev 2)

The NIST RMF is a seven-step process—Prepare, Categorize, Select, Implement, Assess, Authorize, Monitor—applied at organizational, mission/business, and system tiers. The framework is mandatory for federal civilian agencies under FISMA and is adopted by reference in DoD and Intelligence Community RMF implementations. PQC migration maps directly onto existing RMF artifacts without introducing a parallel compliance track.

Book Phase	Framework Activity	PQC Migration Activity
Phase 0: Organize	**Prepare (org + system level)**	Establish CCOE. Identify mission-critical systems requiring PQC migration. Update risk management strategy to include quantum threat. Align PQC scope with existing authorization boundaries.
Phase 0: Organize	**Categorize**	Review existing FIPS 199 / CNSSI 1253 categorizations. Systems handling long-lifetime sensitive data warrant higher categorization for HNDL risk.
Phase 1: Edge First	**Select & Implement**	Tailor existing control baselines (NIST SP 800-53 SC-8, SC-12, SC-13, SC-17) to include PQC. Implement hybrid TLS on internet-facing TLS terminators. Update SSPs to reflect PQC additions.
Phase 2: Trust Infrastructure	**Select & Implement**	Extend control implementation to PKI (code signing, firmware signing), HSMs, VPN/IPsec. Document PQC implementation in SSP.
Phase 3: Broaden	**Assess**	Assess PQC control effectiveness. Update SAR with PQC assessment results. Track deviations and risks in POA&M.
Phase 3: Broaden	**Authorize**	Determine whether PQC-capable deployment constitutes a significant change triggering ATO reauthorization. Update ATO package.
Phase 4: Complete and Sustain	**Monitor**	Incorporate PQC posture into continuous monitoring. Track vendor PQC readiness, algorithm deprecation milestones (NIST IR 8547), and emerging side-channel findings in POA&M.

FAA Acquisition Management System (AMS)

The FAA AMS is the lifecycle acquisition framework governing FAA capital investments, including National Airspace System (NAS) infrastructure. Authoritative guidance resides at fast.faa.gov (FAA Acquisition System Toolset). The AMS comprises six lifecycle phases with distinct decision points overseen by the Joint Resources Council (JRC). Security work integrates via the Information Systems Security Engineering (ISSE) process, which applies NIST SP 800-53 controls to AMS deliverables such as the Preliminary and Final Requirements documents.

Book Phase	Framework Activity	PQC Migration Activity
Phase 0: Organize	**Service Analysis & Strategic Planning**	Identify services with long-lifetime data or safety-critical cryptographic dependencies. Include PQC readiness in strategic planning.
Phase 0: Organize	**Concept & Requirements Definition**	Develop PQC-aware Concept of Operations. Document cryptographic requirements that support hybrid TLS, ML-DSA signing, and ML-KEM key exchange in preliminary requirements documents (pPR).
Phase 1: Edge First / Phase 2	**Initial Investment Analysis**	Include PQC capability in alternatives analysis. Develop Basis of Estimates (BOE) for PQC-capable components. Tailor NIST SP 800-53 controls to the acquisition.
Phase 1: Edge First / Phase 2	**Final Investment Analysis**	Finalize security test plans including PQC verification. Update SIR, SOW, and CDRL with PQC requirements. Obtain stakeholder sign-off on PQC scope.
Phase 2: Trust Infrastructure	**Solution Implementation**	Execute DT/OT/IOA for PQC-enabled systems. Verify hybrid TLS operation in the NAS environment. Address any PQC-induced performance regressions before In-Service Decision.
Phase 3 / Phase 4	**In-Service Management**	Include PQC posture in ongoing SCAP reporting. Plan technology refresh cycles around PQC milestones (2030 NIST deprecation, CNSA 2.0 exclusive use deadlines). Re-certify when significant PQC changes occur.

FedRAMP (Federal Risk and Authorization Management Program)

FedRAMP provides government-wide security assessment and authorization for cloud service offerings (CSOs) used by federal agencies. Cloud service providers (CSPs) achieve authorization via agency sponsorship or program authorization pathways. Current authorizations use the Rev 5 baselines; the FedRAMP 20x modernization initiative (announced March 2025) introduces automation-driven continuous reporting and Key Security Indicators. Core deliverables remain the System Security Plan, Plan of Action and Milestones, and monthly continuous monitoring submissions.

Book Phase	Framework Activity	PQC Migration Activity
Phase 0: Organize	**Authorization Boundary Definition**	Identify cloud service offerings within FedRAMP boundary that rely on quantum-vulnerable cryptography. Document cryptographic modules in SSP per FRR203.
Phase 1: Edge First	**Control Implementation (Rev 5)**	Deploy hybrid TLS on CSO-facing endpoints. Implement PQC-capable cryptographic modules aligned with FedRAMP Cryptographic Modules Guidance. Update SSP and boundary documentation.
Phase 2: Trust Infrastructure	**Annual Assessment / 3PAO**	Include PQC controls in annual assessment scope. Capture PQC evidence in Integrated Inventory Workbook (IIW). Update continuous monitoring submissions.

Book Phase	Framework Activity	PQC Migration Activity
Phase 3: Broaden	**Significant Change Request**	Major PQC deployments (new cryptographic modules, PKI migration, cipher suite changes) trigger SCR workflow. Document per FedRAMP ConMon Playbook significant-change process.
Phase 4: Complete and Sustain	**Continuous Monitoring (ConMon)**	Monthly ConMon submissions reflect PQC posture. POA&M tracks remaining quantum-vulnerable systems with target remediation dates. Prepare for FedRAMP 20x automation-driven evidence model.

DoD Risk Management Framework (DoDI 8510.01)

DoDI 8510.01, reissued July 19, 2022 as "Risk Management Framework for DoD Systems," adopts the NIST SP 800-37 Rev 2 RMF process while layering DoD-specific governance. Categorization uses CNSSI 1253 rather than FIPS 199 for National Security Systems. The framework emphasizes cybersecurity reciprocity—the reuse of authorization evidence across Components to reduce redundant testing. PQC migration for DoD Components aligns with the CNSA 2.0 timeline (exclusive PQC use for NSS by 2030–2035).

Book Phase	Framework Activity	PQC Migration Activity
Phase 0: Organize	**Prepare / Tier 1–2**	OSD-level PQC policy aligns with CNSA 2.0 timeline. DoD Component CIOs integrate PQC into cybersecurity strategy. RMF TAG guidance referenced for PQC implementation.
Phase 0: Organize	**Categorize (CNSSI 1253)**	Review NSS categorizations. Systems processing Top Secret, Secret, or long-lifetime classified data prioritized for PQC migration. Align with CNSA 2.0 exclusivity requirements.
Phase 1 / Phase 2	**Select & Implement**	Select PQC controls per NIST SP 800-53 with CNSSI 1253 overlays. Deploy CNSA 2.0-compliant implementations: ML-KEM-1024 and ML-DSA-87 for NSS networking. Document in SSP.
Phase 2: Trust Infrastructure	**Assess**	Assess PQC control implementation. Leverage DoD Cybersecurity Reciprocity where possible to reduce redundant testing. Document findings for the Receiving AO.
Phase 3: Broaden	**Authorize**	AO makes risk-based authorization decision for PQC-enabled system. Reciprocity framework enables cross-Component reuse of PQC authorization evidence.
Phase 4: Complete and Sustain	**Monitor**	Continuous monitoring per DoDI 8530.01. Track CNSA 2.0 milestones (NSS exclusive PQC use by 2030–2035). Update ISRMC (DoD Risk Executive Function) on enterprise PQC posture.

Cross-Framework Observations

Three patterns recur across all four frameworks. First, the SSP (or its framework equivalent) is always the anchor document—PQC controls must be documented there regardless of which framework governs the system. Second, the POA&M is always the tracking mechanism for incomplete PQC migration; remaining quantum-vulnerable systems should be recorded there with target remediation milestones. Third, authorization decisions (ATO, In-Service Decision, FedRAMP Authorization) are significant-change events when PQC deployment substantially alters the system's cryptographic posture. Program managers should plan for these decision gates in the timeline.

The book's five-phase migration model (Chapter 6) deliberately does not mirror any one framework's step structure. This separation is intentional: PQC migration spans multiple systems, each of which may be at a different point in its own RMF/AMS/FedRAMP/DoD RMF cycle. The book's phases describe the cryptographic work; the framework steps describe how that work is authorized and sustained within federal compliance structures. Use both views together.

APPENDICES & REFERENCE

Bibliography

Master reference list for all sources cited in chapter endnotes. Organized by category. ~95 entries covering all 128 endnotes across Chapters 1–9.

Foundational Physics and Quantum Mechanics

Planck, M. “Über das Gesetz der Energieverteilung im Normalspektrum.” Annalen der Physik 309 (1901): 553–563.

Einstein, A. “Über einen die Erzeugung und Verwandlung des Lichtes betreffenden heuristischen Gesichtspunkt.” Annalen der Physik 322, no. 6 (1905): 132–148.

Bohr, N. “On the Constitution of Atoms and Molecules.” Philosophical Magazine 26 (1913): 1–25.

de Broglie, L. “Recherches sur la théorie des quanta.” PhD thesis, University of Paris, 1924.

Heisenberg, W. “Über quantentheoretische Umdeutung kinematischer und mechanischer Beziehungen.” Zeitschrift für Physik 33 (1925): 879–893.

Schrödinger, E. “Quantisierung als Eigenwertproblem.” Annalen der Physik (1926). Wave mechanics formulation.

Born, M. “Zur Quantenmechanik der Stoßvorgänge.” Zeitschrift für Physik 37 (1926): 863–867. Probabilistic interpretation.

Heisenberg, W. “Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik.” Zeitschrift für Physik 43 (1927): 172–198. Uncertainty principle.

Schrödinger, E. “Die gegenwärtige Situation in der Quantenmechanik.” Naturwissenschaften 23 (1935): 807–812, 823–828, 844–849. Cat thought experiment.

Einstein, A., Podolsky, B., and Rosen, N. “Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?” Physical Review 47, no. 10 (1935): 777–780. EPR paradox.

Bell, J.S. “On the Einstein Podolsky Rosen Paradox.” Physics 1, no. 3 (1964): 195–200.

Aspect, A., Dalibard, J., and Roger, G. “Experimental Realization of EPR-Bohm Gedankenexperiment.” Physical Review Letters 49 (1982): 1804–1807.

Tonomura, A. et al. “Demonstration of single-electron buildup of an interference pattern.” American Journal of Physics 57 (1989): 117–120.

Pais, A. Subtle is the Lord: The Science and the Life of Albert Einstein. Oxford University Press, 1982.

Nielsen, M.A. and Chuang, I.L. Quantum Computation and Quantum Information. Cambridge University Press, 10th Anniversary Edition, 2010.

Mermin, N.D. Quantum Computer Science: An Introduction. Cambridge University Press, 2007.

Quantum Biology

Engel, G.S. et al. “Evidence for wavelike energy transfer through quantum coherence in photosynthetic systems.” Nature 446 (2007): 782–786.

Panitchayangkoon, G. et al. “Long-lived quantum coherence in photosynthetic complexes at physiological temperature.” PNAS 107, no. 29 (2010): 12766–12770.

Cao, J. et al. “Quantum biology revisited.” Science Advances 6, no. 14 (2020).

Lambert, N. et al. “Quantum biology.” Nature Physics 9 (2013): 10–18.

Quantum Algorithms and Complexity

Shor, P.W. “Algorithms for Quantum Computation: Discrete Logarithms and Factoring.” Proceedings 35th Annual Symposium on FOCS (1994): 124–134.

Grover, L.K. “A fast quantum mechanical algorithm for database search.” Proceedings 28th Annual ACM STOC (1996): 212–219.

Gidney, C. and Ekerå, M. “How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits.” Quantum 5, 433 (2021).

Gidney, C. “Factoring integers with sublinear resources on a superconducting quantum processor.” arXiv:2505.15917 (May 2025).

Mosca, M. and Piani, M. Quantum Threat Timeline Report. Global Risk Institute, 2022.

Classical Cryptography Foundations

Diffie, W. and Hellman, M. “New Directions in Cryptography.” IEEE Transactions on Information Theory 22, no. 6 (1976): 644–654.

Rivest, R.L., Shamir, A., and Adleman, L. “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems.” Communications of the ACM 21, no. 2 (1978): 120–126.

Lenstra, A.K. “Key Lengths.” The Handbook of Information Security (2004). GNFS factoring complexity.

Lyubashevsky, V. “Fiat-Shamir With Aborts: Applications to Lattice and Factoring-Based Signatures.” ASIACRYPT 2009.

NIST Standards and Publications

NIST Special Publication 800-37 Revision 2. Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy. December 2018. https://csrc.nist.gov/pubs/sp/800/37/r2/final

NIST Special Publication 800-207. Zero Trust Architecture. Scott Rose, Oliver Borchert, Stu Mitchell, Sean Connelly. August 2020. https://csrc.nist.gov/pubs/sp/800/207/final

NIST Special Publication 800-207A. A Zero Trust Architecture Model for Access Control in Cloud-Native Applications in Multi-Cloud Environments. September 2023. https://csrc.nist.gov/pubs/sp/800/207/a/final

NIST FIPS 203. Module-Lattice-Based Key-Encapsulation Mechanism Standard (ML-KEM). August 2024.

NIST FIPS 204. Module-Lattice-Based Digital Signature Standard (ML-DSA). August 2024.

NIST FIPS 205. Stateless Hash-Based Digital Signature Standard (SLH-DSA). August 2024.

NIST FIPS 206 (Draft). FFT-over-NTRU-Lattice Digital Signature Standard (FN-DSA). Expected 2025–2026.

NIST. "NIST Selects HQC as Fifth Algorithm for Post-Quantum Encryption." News release, March 11, 2025. HQC specification: https://pqc-hqc.org

NIST IR 8547 (Initial Public Draft). Transition to Post-Quantum Cryptography Standards. November 2024.

NIST SP 800-57 Part 1 Rev. 5. Recommendation for Key Management: Part 1. May 2020.

NIST SP 800-131A Rev. 3. Transitioning the Use of Cryptographic Algorithms and Key Lengths. November 2024.

NIST SP 800-208. Recommendation for Stateful Hash-Based Signature Schemes (LMS, XMSS). December 2019.

NIST SP 800-227 (Draft). Recommendations for Key-Encapsulation Mechanisms. 2024.

NIST SP 1800-38A/B/C (Preliminary Drafts). Migration to Post-Quantum Cryptography, Volumes A–C. NCCoE.

NIST CSWP 39 (Final). Considerations for Achieving Crypto Agility. December 2025.

NIST PQC Standardization Project. Initiated 2016, 82 submissions from 25 countries. https://csrc.nist.gov/projects/post-quantum-cryptography

NIST PQC Round 3 Report. Status Report on the Third Round of the NIST PQC Standardization Process. 2022.

NIST PQC Conference Presentations (2024–2026). Fifth and Sixth PQC Standardization Conferences. Regenscheid, A. & Newhouse, B. "NIST PQC Update." December 2024.

Moody, D. (NIST). HQC selection announcement and commentary on algorithmic diversity. March 2025.

NSA and U.S. Government

Department of Defense Instruction 8510.01. Risk Management Framework for DoD Systems. July 19, 2022. https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/851001p.pdf

Federal Risk and Authorization Management Program (FedRAMP). RFC-0004 Boundary Policy (Draft). January 16, 2025. Defines authorization boundary scope for cloud service offerings.

https://www.fedramp.gov/rfcs/0004/

Federal Risk and Authorization Management Program (FedRAMP). Continuous Monitoring Playbook Version 1.0. November 17, 2025. https://www.fedramp.gov/resources/documents/Continuous_Monitoring_Playbook.pdf

Federal Aviation Administration. Acquisition Management System (AMS) Policy. FAA Acquisition System Toolset (FAST). https://fast.faa.gov/AMS_Policy.cfm

CISA. Zero Trust Maturity Model Version 2.0. April 2023. https://www.cisa.gov/sites/default/files/2023-04/CISA_Zero_Trust_Maturity_Model_Version_2_508c.pdf

Office of Management and Budget (OMB) Memorandum M-22-09. Moving the U.S. Government Toward Zero Trust Cybersecurity Principles. January 26, 2022.

NSA. CNSA Suite 2.0 Algorithm Guidance (PP-22-1338, Ver. 1.0). September 2022. FAQ Ver. 2.1, December 2024.

NSA. Quantum Key Distribution (QKD) and Quantum Cryptography (QC). Cybersecurity Advisory.

NSA. Quantum Computing and Post-Quantum Cryptography FAQ. August 2021.

NSA. draft-guthrie-cnsa2-ipsec-profile: CNSA Suite 2.0 Profile for IPsec. IETF Internet-Draft.

White House. National Security Memorandum 10 (NSM-10). May 4, 2022.

White House. Executive Order 14144: Strengthening Cybersecurity. January 2025.

White House. Executive Order 14306: Amending EO 14028 and EO 14144. June 2025.

Quantum Computing Cybersecurity Preparedness Act. Pub. L. No. 117-349. December 21, 2022.

OMB Memorandum M-23-02: Migrating to Post-Quantum Cryptography. November 2022.

CISA. Post-Quantum Cryptography Initiative. https://www.cisa.gov/quantum

CISA. Strategy for Migrating to Automated Post-Quantum Cryptography Discovery and Inventory Tools. September 2024.

USDA Acquisition Regulation (AGAR). Revised September 2025. Explicit PQC procurement language.

U.S. Federal Government PQC Migration Cost Estimate: $7.1 billion (2025–2035). Referenced in OMB budget documents.

IETF Standards and Drafts

RFC 6928. Increasing TCP's Initial Window. April 2013.

RFC 8446. The Transport Layer Security (TLS) Protocol Version 1.3. August 2018.

RFC 8555. Automatic Certificate Management Environment (ACME). Barnes, R., Hoffman-Andrews, J., McCarney, D., Kasten, J. March 2019.

RFC 8784. Mixing Preshared Keys in IKEv2 for Post-Quantum Security. June 2020.

RFC 8879. TLS Certificate Compression. December 2020.

RFC 9580. OpenPGP (Updated Specification). July 2024.

IETF draft-ietf-tls-ecdhe-mlkem (draft-04). Hybrid ECDHE-MLKEM Key Agreement for TLS 1.3. February 2026. Kwiatkowski (PQShield), Kampanakis (AWS), Westerbaan (Cloudflare), Stebila (Waterloo).

IETF draft-ietf-tls-hybrid-design-16. Hybrid Key Exchange in TLS 1.3. September 2025.

IETF draft-ietf-sshm-mlkem-hybrid-kex. ML-KEM Hybrid Key Exchange for SSH.

IETF draft-fregly-research-agenda-for-pqc-dnssec-02. Research Agenda for a Post-Quantum DNSSEC. 2024–2025.

IETF draft-sheth-pqc-dnssec-strategy-00. Post-Quantum Cryptography Strategy for DNSSEC. October 2025.

DNS Flag Day 2020. EDNS buffer size recommendation of 1,232 bytes. https://dnsflagday.net/2020/

IETF LAMPS Working Group. Composite certificate formats for CMS (S/MIME PQC integration). Drafts in progress, 2025–2026.

IETF PLANTS Working Group. Post-quantum Lightweight Authentication for Network TLS Security. Formed 2026 to standardize Merkle Tree Certificates.

Industry, Vendor, and Cloud Provider Sources

Cloudflare Blog. “Automatically Secure: How We Upgraded 6,000,000 Domains.” September 2025.

Cloudflare Blog. “State of the Post-Quantum Internet in 2025.” October 2025.

Cloudflare Blog. “Keeping the Internet Fast and Secure: Introducing Merkle Tree Certificates.” October 2025.

Cloudflare Blog. “Post-Quantum Zero Trust.” March 2025.

Google Security Blog. “Cultivating a Robust and Efficient Quantum-Safe HTTPS.” February 2026.

Chromium Blog. “Advancing Our Amazing Bet on Asymmetric Cryptography.” May 2024. PQC key exchange vs. authentication priority.

OpenSSH 9.0 Release Notes. April 2022. sntrup761x25519-sha512 default key exchange.

OpenSSH 10.0 Release Notes. April 2025. mlkem768x25519-sha256 default key exchange.

Microsoft. “Post-Quantum Cryptography APIs Now Generally Available on Microsoft Platforms.” November 2025. Windows Server 2025, Windows 11, .NET 10. ADCS PQC support targeted early 2026.

AWS. “AWS KMS Adds Support for Post-Quantum ML-DSA Digital Signatures.” June 2025.

Google Cloud Blog. “Announcing Quantum-Safe Digital Signatures in Cloud KMS.” February 2025.

CA/Browser Forum. Ballot SC-081v3: Reducing Certificate Validity Periods. 200 days (March 2026), 100 days (March 2027), 47 days (March 2029).

DigiCert Blog. “Google Merkle Tree Certificates.” March 2026.

Open Quantum Safe (OQS) Project. PQC algorithm benchmarks and liboqs library. https://openquantum-safe.org

PKI Consortium PQC Working Group. Hybrid and composite certificate format development. https://pkic.org

Gartner Research. Almond, Sarah, and Mark Horvath. “A Journey Guide to Postquantum Readiness.” Research note G00843746, 13 March 2026. Provides the CCOE model, five-layer crypto-agility framework, “inventory at source” concept, discovery pilot recommendations, internal cryptographic policy requirements, and CFO-partnership financial planning guidance referenced throughout Chapters 5 and 6.

SafeLogic. “NIST Publishes Next Volume of PQC Migration Guidance.” 2025. Discovery tool findings.

Thales. Luna HSM Firmware 7.9.0+. ML-DSA support with operational caveats for stateful hash-based signatures.

Entrust. nShield 5 firmware. NIST CAVP-validated ML-DSA, ML-KEM, SLH-DSA. FIPS 140-3 certification in progress (2025).

NGINX. “NGINX Introduces Native Support for ACME Protocol.” NGINX Community Blog. August 12, 2025. https://blog.nginx.org/blog/native-support-for-acme-protocol

F5, Inc. Kojot ACME: An ACMEv2 client utility function for integration and advanced features on the F5 BIG-IP. GitHub: f5devcentral/kojot-acme (MIT License). https://github.com/f5devcentral/kojot-acme

International Guidance

AIVD, CWI, TNO. The PQC Migration Handbook: Guidelines for Migrating to Post-Quantum Cryptography. 2nd Edition, December 2024.

UK NCSC. “Quantum Networking Technologies.” Updated white paper, August 2025.

UK NCSC. “Timelines for Migration to Post-Quantum Cryptography.” Three-phase roadmap, March 2025.

ENISA. Post-Quantum Cryptography Integration Study. 2024.

NIS Cooperation Group. “Coordinated Implementation Roadmap for the Transition to Post-Quantum Cryptography.” Early 2025.

European Commission. Proposed directive amending NIS2 with PQC requirements. January 2026.

G7 Cyber Expert Group. Financial Sector PQC Roadmap. January 13, 2026. Co-chaired by US Treasury and Bank of England.

RAND Corporation. “U.S.-Allied Militaries Must Prepare for the Quantum Threat to Cryptography.” June 2025.

Academic Papers and Conference Presentations

Kampanakis, P. et al. “The Impact of Data-Heavy Post-Quantum TLS 1.3.” NIST 5th PQC Standardization Conference. 2024.

Anastasova, M. et al. “The Impact of ML-KEM and ML-DSA on mTLS Connection Time-To-Last-Byte.” PKI Consortium PQC Conference, Austin. 2025.

Rawat, A. and Jhanwar, M. “Post-Quantum DNSSEC with Faster TCP Fallbacks.” INDOCRYPT 2024, LNCS vol. 15496.

arXiv:2512.00110. “Post-Quantum-Resilient Audit Evidence for Long-Lived Regulated Systems.” February 2026. Security proofs for Q-Audit Integrity, Q-Non-Equivocation, Q-Binding.

VeriSign/NIST. “Post-Quantum Diversity for DNSSEC: Routine Performance, Resilient Fallback.” 6th PQC Standardization Conference. 2025.

postquantum.com. “The Complete US Post-Quantum Cryptography (PQC) Regulatory Framework in 2026.” February 2026.

postquantum.com. “Google’s Merkle Tree (MTC) Gambit to Quantum-Proof HTTPS.” March 2026.

F5, Inc. Resources

F5, Inc. BIG-IP PQC / TLS Offload Overview (internal field guidance). 2025.

F5, Inc. BIG-IP v21.1 Release Notes: PQC cipher support, hybrid TLS cipher groups, quantum-resistant VPN.

F5, Inc. BIG-IP 17.5.1: Initial X25519MLKEM768 hybrid key exchange support.

F5, Inc. Application Study Tool (AST): Open-source BIG-IP telemetry. GitHub: f5devcentral/application-study-tool.

F5, Inc. “F5 Strengthens Its Application Delivery and Security Platform.” Press release, March 2026. F5 Insight announcement.

F5, Inc. BIG-IP SSL Orchestrator (SSLO) product documentation. TLS visibility and crypto discovery.

— End of Appendices —

The Post-Quantum Cryptography Field Guide — A Practitioner's Handbook