# Post-Quantum Cryptography Readiness: eSolia Internal Action Plan

Prepared for

**Rick Cogley**

April 12, 2026

eSolia

## Contents

## Purpose

Define what eSolia must do — to its own infrastructure, accounts, and engineering practices — to be defensibly post-quantum ready ahead of the 2029–2030 industry inflection point. This document is about us, not our clients. A separate client-facing guide will follow.

## TL;DR

Most of the Q-Day-relevant cryptography in our stack is already being upgraded for us, automatically, by Cloudflare, Microsoft, and GitHub. Our job is **not** to roll our own crypto. Our job is to:

1. **Stop blocking** the upgrades that vendors are already deploying (mainly: don't turn on FIPS mode unless contractually required).
2. **Turn on the opt-in flags** where vendors require explicit consent (WARP MDM, GitHub SSH key types, modern OpenSSH on workstations).
3. **Inventory long-lived secrets** so we know what's actually exposed to harvest-now-decrypt-later (HNDL) risk.
4. **Build crypto-agility into Pulse** so we can swap algorithms without re-architecting.

Think of post-quantum migration like seismic retrofitting in Tokyo: the building codes have changed, the structural engineers (Cloudflare, Microsoft) are doing the heavy lifting on shared infrastructure, but each tenant still has to bolt down their own bookcases. This document is the bookcase list.

## Primer: What This Is All About

If you're picking this document up cold, read this section first. The rest of the plan assumes you understand the concepts here.

### What is cryptography, in 30 seconds

Cryptography is the math we use to keep secrets and verify identities on a network. There are two flavors:

- **Symmetric cryptography** uses one shared key to both encrypt and decrypt. AES-256 is the canonical example. It's fast, it's used for the actual bulk data encryption in almost every protocol, and — importantly — **quantum computers don't really break it.** A quantum attack against AES-256 only weakens it to the equivalent of AES-128, which is still well beyond brute force. Symmetric crypto is fine.
- **Asymmetric cryptography** (also called public-key cryptography) uses a mathematically linked pair of keys: a public key anyone can see and a private key only the owner holds. RSA, ECC (elliptic curve cryptography), and ECDSA are the common examples. This is what we use to establish a shared symmetric key over an untrusted network, and to sign things so others can verify they came from us. **This is what quantum computers break.**

Every TLS connection (HTTPS, SSH, VPNs, you name it) uses asymmetric crypto for the "let's agree on a session key" handshake, and then switches to symmetric crypto for the actual conversation. The handshake is the vulnerable part.

### What is a quantum computer threat, concretely

A quantum computer is a fundamentally different kind of machine that exploits quantum mechanics to do certain calculations exponentially faster than classical computers. Most things it's not faster at. But for two specific math problems — integer factorization and discrete logarithms — it's catastrophically faster, thanks to an algorithm called **Shor's algorithm** (published in 1994). Those two problems happen to be exactly what RSA and ECC rely on for their security. So a sufficiently large quantum computer running Shor's algorithm can take a public RSA key and recover the private key in minutes instead of the billions of years it would take a classical computer.

The day a quantum computer is built that can actually do this to real-world key sizes is called **Q-Day**. The machine itself is called a **CRQC** — a cryptographically relevant quantum computer. No CRQC exists today. The honest expert estimates range from "maybe never" to "by 2030." Recent algorithmic breakthroughs have shifted the consensus toward the earlier end of that range, which is why the industry is moving now.

### Why the urgency, when no CRQC exists yet

The threat that matters today is called **harvest-now, decrypt-later** (HNDL). The idea is simple: an adversary doesn't need a quantum computer today to attack data that needs to stay confidential for ten years. They just record the encrypted traffic now, store it, and decrypt it the day a CRQC becomes available.

This means anything we send across a network today that's still sensitive in 2032 is already at risk. For eSolia, this includes FSA audit evidence, client PII subject to retention requirements, signing keys, M&A documents, and long-lived credentials. Authentication has the opposite property — a forged credential in 2032 is only useful in 2032 — so authentication can be migrated later. Confidentiality has to be migrated first.

## What "post-quantum" means in practice

**Post-quantum cryptography (PQC)** is a new family of asymmetric algorithms whose security is based on math problems that quantum computers don't know how to break. The U.S. National Institute of Standards and Technology (NIST) ran a multi-year competition starting in 2016 to evaluate candidates and standardized the winners in 2024. The two that matter for us:

- **ML-KEM** (Module-Lattice Key Encapsulation Mechanism, formerly known as **Kyber**, standardized as NIST FIPS 203). This replaces RSA and ECDH for key agreement — the "let's agree on a session key" step of TLS, SSH, IPsec, and similar protocols. When you see `X25519MLKEM768` in a TLS handshake log, that's hybrid ML-KEM at work.
- **ML-DSA** (Module-Lattice Digital Signature Algorithm, formerly known as **Dilithium**, standardized as NIST FIPS 204). This replaces RSA and ECDSA for digital signatures — the "this certificate really came from this CA" step.

Both are based on a math problem called Module Learning With Errors, which is believed to be hard for both classical and quantum computers. "Believed to be" is the operative phrase — these algorithms are newer than RSA and have had less real-world stress-testing, which is why everyone is deploying them in **hybrid** mode for now.

## What is hybrid cryptography

A **hybrid** scheme runs both a classical algorithm (like X25519) and a post-quantum algorithm (like ML-KEM-768) in parallel during the handshake, then combines their outputs. The connection is secure as long as **at least one** of the two algorithms holds up. If ML-KEM turns out to have a flaw discovered next year, the classical X25519 still protects you. If a CRQC is built next year, ML-KEM still protects you. It's a belt-and-suspenders approach, and it's the consensus default for 2025–2028. Pure post-quantum will follow once the algorithms have more battle-testing.

## What is "crypto-agility"

**Crypto-agility** is the property of a system that lets you swap out cryptographic algorithms without rewriting the application. Concretely: if your code has `RSA_4096` hard-coded, you don't have crypto-agility. If your code reads an algorithm identifier from a versioned header in the data and dispatches to the right library, you do. This matters because we're going to be doing this migration again in five or ten years when ML-KEM gets replaced by something else, and the systems we build today should make that future migration cheap. The Pulse work item PU-2 is specifically about this.

## The acronyms you'll see in this document

A condensed glossary lives at the end of the document. The most important ones:

- **PQC** — post-quantum cryptography
- **HNDL** — harvest-now, decrypt-later (the active threat)
- **Q-Day** — the day a CRQC capable of breaking real-world crypto exists
- **CRQC** — cryptographically relevant quantum computer
- **KEX** — key exchange (the handshake step that establishes a session key)
- **ML-KEM** — the NIST-standardized PQ key exchange algorithm (was: Kyber)
- **ML-DSA** — the NIST-standardized PQ signature algorithm (was: Dilithium)
- **TLS** — Transport Layer Security, the encryption layer under HTTPS
- **NIST** — U.S. National Institute of Standards and Technology, sets the standards we follow

- **FIPS** — Federal Information Processing Standards; FIPS 140-x is the U.S. government cryptographic module validation program

## Background

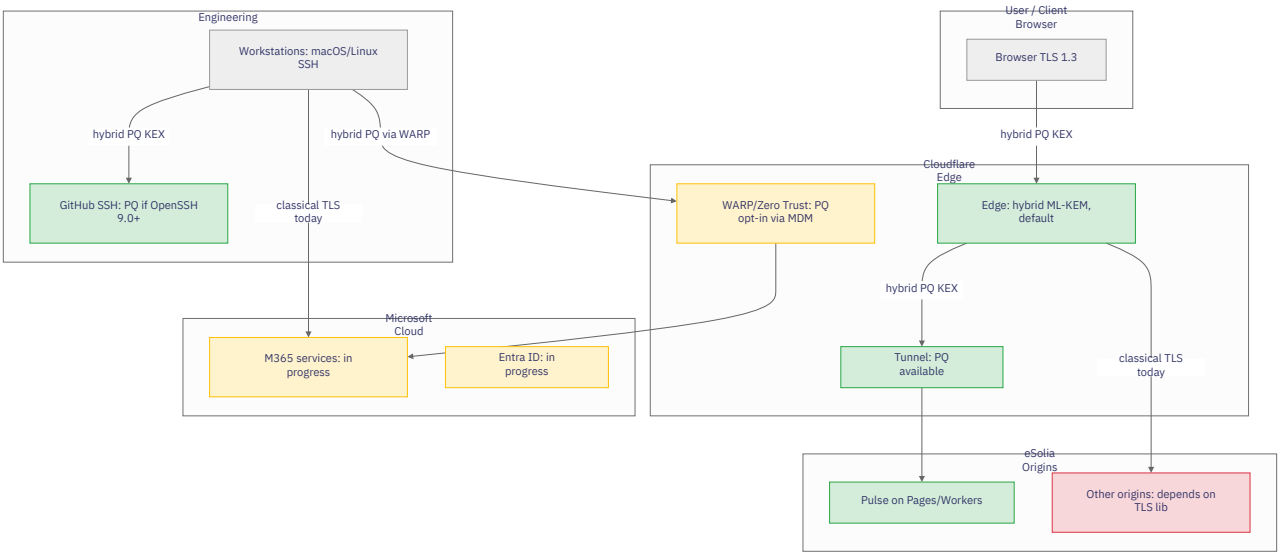
The threat is well-known: a sufficiently capable quantum computer running Shor's algorithm breaks RSA and ECC. The realistic near-term threat is **not** real-time decryption — it's HNDL, where an adversary records encrypted traffic today and decrypts it once a cryptographically relevant quantum computer (CRQC) exists. That makes anything with long-term confidentiality value (FSA audit evidence, M&A documents, signing keys, password hashes, client PII subject to retention) the priority for protection today.

Two industry data points anchor our timeline:

- **Cloudflare** and **Google** have both committed to full post-quantum security — including authentication — by **2029**, with intermediate milestones starting mid-2026.
- **Microsoft's** Quantum Safe Program targets early adoption by **2029** and full transition by **2033**, with NIST ML-KEM and ML-DSA already shipping in Windows 11 24H2/25H2, Windows Server 2025, and .NET 10 as of November 2025.
- **NIST** has set deprecation of RSA and ECC for **2030**, with full disallowance by 2035.

Our internal target: **be in a verifiable, defensible PQC posture by end of 2027**, so we have margin against the 2029–2030 deadlines and can credibly answer FSA-regulated client procurement questionnaires.

## Where PQC Sits in Our Stack




Diagram

**Legend:** Green = PQ-protected today. Yellow = vendor work in progress, partial coverage. Red = no PQ protection yet, requires workaround or acceptance.

## Action Plan by Platform

### 1. Cloudflare (Our Hosting and Zero Trust)

**Current state:** All eSolia zones front-ended by Cloudflare already get hybrid ML-KEM key agreement by default for TLS 1.3 / HTTP/3 connections from modern browsers. This is on for free plans and up. We don’t have to enable it; we just have to not break it.

**Required actions:**

| # | Action | Owner | Target | Notes |
|---|---|---|---|---|
| CF-1 | Audit all zones for FIPS mode. Disable unless a specific client contract mandates FIPS. | Rick | 2026-04-30 | FIPS mode disables PQ key agreement entirely. Trade-off must be documented per zone. |
| CF-2 | Confirm TLS 1.3 is enabled on every zone (SSL/TLS → Edge Certificates). | Rick | 2026-04-30 | TLS 1.2 cannot carry hybrid ML-KEM. |
| CF-3 | Migrate any remaining direct-to-origin pulls to Cloudflare Tunnel where the origin is under our control. | Rick | 2026-06-30 | Tunnel uses PQ-protected transport between edge and origin without origin-side TLS library upgrades. |
| CF-4 | Verify Cloudflare One Appliance (if/when deployed) is on version >= 2026.2.0. | Rick | On deployment | This version added hybrid ML-KEM to Cloudflare IPsec automatically. |
| CF-5 | Enable WARP post-quantum override via MDM for the eSolia fleet. Set `enable_post_quantum=true` | Rick | 2026-05-31 | Without this, WARP defaults to a cautious phased rollout with downgrade allowed. The MDM flag forces PQ-only and prevents downgrade attacks. |
| CF-6 | Document zones where PQ posture is incomplete (origin can't be tunneled, FIPS required, etc.) in a tracking sheet. | Rick | 2026-05-31 | This becomes the FSA audit trail when asked. |

**Verification command** (for any zone):

```
# From a modern Chrome/Edge/Firefox, visit:

# https://pq.cloudflareresearch.com

# Should report: X25519MLKEM768
```

## 2. Microsoft 365, Entra, and Windows

**Current state:** Microsoft has shipped ML-KEM and ML-DSA in the Windows CNG API and certificate API as of the November 2025 Windows update for Windows 11 24H2/25H2 and Windows Server 2025. .NET 10 exposes the same algorithms to developers. Active Directory Certificate Services PQC support is GA early 2026. The actual M365 services (Exchange, SharePoint, Teams) are still in Microsoft's "Phase 3" rollout, with foundational signing and identity services prioritized through 2026–2028 and full coverage by 2033.

**What this means for us:** We are downstream of Microsoft on the service side. Our job is workstation and account hygiene, not service-side configuration (we have no levers there).

**Required actions:**

| # | Action | Owner | Target | Notes |
|---|---|---|---|---|
| MS-1 | Confirm all eSolia-managed workstations are on Windows 11 24H2 or later (or current macOS / current Linux distro). | Rick | 2026-05-31 | Older Windows builds lack the CNG ML-KEM/ML-DSA APIs entirely. Track via Intune compliance. |
| MS-2 | Force-rotate any service principal credentials, app registration secrets, or Entra-issued tokens with effective lifetimes longer than 24 months. | Rick | 2026-06-30 | These are HNDL targets if currently in transit logs. Combine with the existing E5-tier credential audit. |
| MS-3 | Inventory Conditional Access policies that depend on certificate-based authentication. Note which CAs will need PQ certificates by 2028. | Rick | 2026-07-31 | Pure documentation step; no remediation possible until ADCS PQ cert issuance is GA in our environment. |
| MS-4 | For the eSolia tenant: enable strong authentication methods (FIDO2, Windows Hello for Business) where possible. PQ-resistant authenticators reduce reliance on RSA-backed credential flows. | Rick | 2026-09-30 | FIDO2 keys themselves aren't PQ yet, but the architecture is more crypto-agile than legacy MFA. |
| MS-5 | Review M365 DLP and sensitivity label encryption settings. Document which use Microsoft-managed keys (covered by Microsoft's roadmap) vs. customer-held | Rick | 2026-08-31 | Most eSolia content uses Microsoft-managed keys. Confirm for any Purview-encrypted client deliverables. |
| MS-6 | Watch for Microsoft Entra PQ authentication GA announcements; | Rick | Immediate | Standing item. |

**Verification:**

```
# On a Windows 11 workstation, check CNG providers expose ML-KEM:
certutil -csplist | Select-String -Pattern "ML-KEM|ML-DSA"
```

### 3. GitHub

**Current state:** GitHub enabled `sntrup761x25519-sha512` (a hybrid post-quantum SSH key exchange) for SSH access to GitHub.com and non-US Enterprise Cloud regions on **2025-09-17**. HTTPS access is unaffected (already covered by transport TLS). The change is automatic if your SSH client supports it — meaning **OpenSSH 9.0 or newer**. Older clients fall back to classical key exchange silently.

**Required actions:**

| # | Action | Owner | Target | Notes |
|---|---|---|---|---|
| GH-1 | Audit all engineering workstations for OpenSSH version. Require >= 9.0. | Rick | 2026-05-15 | macOS Sonoma+ ships 9.x. Most current Linux distros do too. Windows native OpenSSH may need explicit update. |
| GH-2 | For each workstation, verify the active KEX algorithm against github.com is post-quantum. | Rick | 2026-05-15 | See verification command below. |
| GH-3 | Standardize on HTTPS + GitHub CLI (`gh auth login`) as the default Git remote protocol for new repos. | Rick | 2026-06-30 | HTTPS goes through Cloudflare-fronted endpoints and inherits our edge PQ posture. SSH remains supported. |
| GH-4 | Document the policy: “no production deploy keys with > 1 year validity.” Rotate any existing long-lived deploy keys. | Rick | 2026-07-31 | Reduces HNDL exposure window for the SSH transport layer. |
| GH-5 | For Pulse and any client-deliverable repos, enable Sigstore / cosign-style signed commits where workflow allows. | Rick | 2026-09-30 | Crypto-agility win; Sigstore is migrating toward PQ signature support. |

**Verification command:**

```
# On macOS / Linux / Git Bash:
ssh -Q kex | grep -E 'sntrup|mlkem'

# Should return at least: sntrup761x25519-sha512 or mlkem768x25519-sha256
```

```
# Then verify the actual negotiated algorithm:
ssh -v git@github.com 2>&1 | grep 'kex: algorithm:'

# Should show: kex: algorithm: sntrup761x25519-sha512@openssh.com (or mlkem768x25519-
sha256)
```

If the output shows `curve25519-sha256` instead, the client is too old or the SSH config is forcing a legacy KEX.

### 4. Developer Workstations

**Current state:** OpenSSH 9.0+ (released April 2022) introduced `sntrup761x25519-sha512` as a default KEX, and OpenSSH 9.9+ added `mlkem768x25519-sha256` (the NIST-standardized variant). Modern macOS and modern Linux distros are fine. Windows is the laggard.

**Required actions:**

| # | Action | Owner | Target | Notes |
|---|---|---|---|---|
| WS-1 | Establish baseline: every eSolia engineering workstation running macOS >= Sonoma, Windows 11 24H2+, or a current LTS Linux. | Rick | 2026-05-31 | Track in a workstation inventory spreadsheet. |
| WS-2 | On each workstation, ensure the default `~/.ssh/config` does **not** pin a `KexAlgorithms` line that excludes PQ algorithms. | Rick | 2026-05-31 | A common gotcha — old hardening guides locked KEX to `curve25519-sha256` only. |
| WS-3 | Update Git for Windows installations to a build that bundles OpenSSH 9.5+ (Git for Windows 2.45+). | Rick | 2026-05-31 | Older Git for Windows ships its own legacy OpenSSH. |
| WS-4 | For any GPG signing keys in use (commits, releases), document the algorithm and key length. Plan migration path to a PQ-capable signing tool when one is GA. | Rick | 2026-09-30 | GPG itself has no PQ support yet. Sigstore is the most likely successor for our use case. |
| WS-5 | Disk encryption: confirm FileVault 2 (macOS) and BitLocker (Windows) are on. Both use AES-256, which is symmetric and already considered PQ-safe (Grover's gives only quadratic speedup, easily compensated by key length). | Rick | Immediate | Sanity-check item; mostly already in compliance. |
| WS-6 | Password manager review: 1Password, Bitwarden vault | Rick | 2026-06-30 | Document for the audit trail. |

## 5. Pulse and Internal Development Work

**Current state:** Pulse is built on SvelteKit + Cloudflare Pages + D1 + R2. The runtime cryptography is therefore inherited from Cloudflare's stack and benefits automatically from edge-side PQ KEX. The application-layer cryptography we control is the encrypted client data export, which currently uses RSA-4096 to wrap an AES-256-GCM session key.

**This is the highest-impact eSolia-controlled gap.** RSA-4096 is the part that breaks on Q-Day. AES-256-GCM is fine.

**Required actions:**

| # | Action | Owner | Target | Notes |
|---|---|---|---|---|
| PU-1 | Add ML-KEM-1024 as a hybrid wrapping option alongside RSA-4096 for the client export feature. The exported file should carry both wraps so older clients can still decrypt and forward-secure clients use ML-KEM. | Rick | 2026-09-30 | Implementation: use a JS/WASM ML-KEM library (e.g., `@noble/post-quantum`). Schema-version the export envelope. |
| PU-2 | Build crypto-agility into the export envelope format from day one: a versioned JSON header listing the algorithms used, so future migrations don't require breaking changes. | Rick | Same as PU-1 | Diataxis: add to the Pulse architecture reference doc. |
| PU-3 | Document the RSA-4096 → hybrid migration path in the Pulse SECURITY.md. Include rotation procedure for encryption keypairs. | Rick | 2026-09-30 | This is what FSA audit will ask for. |
| PU-4 | Audit Pulse dependencies for any cryptographic libraries with hard-coded RSA/ECDSA assumptions. Replace or wrap. | Rick | 2026-10-31 | npm audit + manual review. SvelteKit itself has no crypto opinions. |
| PU-5 | For Pulse evidence storage (R2): confirm encryption-at-rest uses AES-256 (Cloudflare default — yes, it does). No action needed beyond enrollment work (already pending). | Rick | 2026-05-31 | One-line audit note. |
| PU-6 | Multi-tenant CPV: when designing | Rick | Tied to CPV timeline | Adds a single line to the existing CPV implementation plan. |

## Cross-Cutting: Long-Lived Secret Inventory

The single most important non-technical task. Build a list of every long-lived secret eSolia holds, classify by exposure window, and prioritize rotation.

| Secret class | Examples | HNDL exposure | Action |
|---|---|---|---|
| Code signing keys | GPG, Sigstore (when adopted) | High — signature longevity matters | Plan rotation, target shorter validity |
| Cloudflare API tokens | Account-level admin tokens | Medium | Already short-lived with API token model; verify no legacy global API keys remain |
| Microsoft Partner Center / GDAP credentials | CPV enrollment, delegated admin | High | Rotate annually; never use long-lived secrets |
| Client CSP customer secrets | Per-tenant app registration secrets | High | Use certificates not secrets where possible; rotate annually |
| GitHub PATs / fine-grained tokens | CI, automation | Medium | 90-day max validity policy |
| Pulse RSA-4096 export keypair | Per-deployment | High | See PU-1 |
| Backup encryption keys | Time Machine, etc. | Medium | Symmetric AES, lower priority but document |
| Wi-Fi WPA2/WPA3 keys | Office network | Low | Symmetric, low value to harvest |

## What We Are Explicitly NOT Solving

To set expectations and avoid scope creep:

- **We are not solving Q-Day for Microsoft 365 services.** We have no control over how fast Microsoft retrofits Exchange Online, SharePoint, or Teams. We track and document.
- **We are not solving Q-Day for client environments.** That's a separate engagement and a separate document.
- **We are not building our own PQ algorithms or libraries.** We use NIST-standardized algorithms via vendor-supplied or well-maintained open-source implementations only.
- **We are not turning on FIPS mode** unless a specific client contract requires it. FIPS and PQ are currently in tension; FIPS-validated PQ implementations are still maturing.
- **We are not migrating away from RSA and ECC immediately.** Hybrid (classical + PQ) is the right posture for 2026–2028; pure PQ is premature for most layers.

## Verification Checklist (Quarterly)

Run this checklist at the end of each quarter and file the result in the eSolia Standards repo as evidence.

- □ Visit `https://pq.cloudflareresearch.com` from a managed workstation. Confirm hybrid ML-KEM negotiated.
- □ Run `ssh -v git@github.com` and confirm `sntrup761x25519-sha512` or `mlkem768x25519-sha256` is the active KEX.
- □ Run `ssh -Q kex` on each engineering workstation and confirm at least one PQ algorithm is listed.
- □ Confirm all Cloudflare zones still have TLS 1.3 enabled and FIPS mode disabled (or documented exceptions).
- □ Confirm WARP MDM `enable_post_quantum=true` is still set on all enrolled devices.
- □ Review the long-lived secret inventory; rotate anything past its rotation window.
- □ Check Microsoft 365 message center for new PQC announcements; update this document if the timeline shifts.
- □ Check Cloudflare blog and `developers.cloudflare.com/ssl/post-quantum-cryptography/` for new features.

## Glossary

For quick lookup. Terms in **bold** are introduced in the Primer section.

| Term | Meaning |
|---|---|
| **ADCS** | Active Directory Certificate Services. Microsoft's enterprise PKI server role. PQ certificate issuance is targeted for early 2026. |
| **AES-256** | Advanced Encryption Standard with 256-bit keys. Symmetric algorithm; already considered post-quantum safe. |
| **CNG** | Cryptography API: Next Generation. The Windows API layer where ML-KEM and ML-DSA were added in November 2025. |
| **CPV** | Control Panel Vendor. A Microsoft Partner Center enrollment type required for multi-tenant app registrations in CSP scenarios. |
| **CRQC** | Cryptographically relevant quantum computer. A quantum computer powerful enough to break real-world RSA/ECC. None exists yet. |
| **CSP** | Cloud Solution Provider. Microsoft's reseller program; eSolia is a CSP. |
| **ECC / ECDSA / ECDH** | Elliptic Curve Cryptography, ECC-based Digital Signature Algorithm, ECC-based Diffie-Hellman key exchange. The current asymmetric crypto family that quantum computers will break. |
| **Entra ID** | Microsoft's cloud identity service, formerly Azure Active Directory. |
| **FIPS mode** | A configuration that restricts a system to FIPS-validated cryptographic algorithms only. Currently in tension with PQ algorithms because PQ FIPS validation is still rolling out. |
| **GDAP** | Granular Delegated Admin Privileges. Microsoft's modern delegated admin model for CSPs. |
| **HNDL** | Harvest-now, decrypt-later. The current active threat: adversaries record encrypted traffic today to decrypt after Q-Day. |
| **Hybrid (cryptography)** | Running a classical algorithm and a PQ algorithm in parallel and combining their outputs. Secure if either one holds up. |
| **IPsec** | Internet Protocol Security. The Layer 3 VPN protocol used by Cloudflare One Appliance and many corporate VPNs. |
| **KEM** | Key Encapsulation Mechanism. A modern term for "the algorithm that establishes a shared secret over an untrusted channel." ML-KEM is one. |
| **KEX** | Key exchange. The handshake step in TLS, SSH, etc. where both sides agree on a session key. |
| **MDM (Cloudflare)** | Mobile Device Management. A platform for centrally configuring devices. We use it to provision the WARP client and its certificates. |

## References

**Primary sources:**

- Cloudflare PQC roadmap (2026 update): https://blog.cloudflare.com/post-quantum-roadmap/
- Cloudflare PQC SSL/TLS docs: https://developers.cloudflare.com/ssl/post-quantum-cryptography/
- Cloudflare One PQC SASE: https://blog.cloudflare.com/post-quantum-sase/
- WARP PQC support: https://blog.cloudflare.com/post-quantum-warp/
- Microsoft Quantum Safe Program: https://www.microsoft.com/en-us/security/blog/2025/08/20/quantum-safe-security-progress-towards-next-generation-cryptography/
- Microsoft PQC APIs GA (Windows / .NET 10): https://techcommunity.microsoft.com/blog/microsoft-security-blog/post-quantum-cryptography-apis-now-generally-available-on-microsoft-platforms/4469093
- GitHub SSH PQC announcement: https://github.blog/engineering/platform-security/post-quantum-security-for-ssh-access-on-github/

**Standards:**

- NIST FIPS 203 (ML-KEM): module-lattice key encapsulation
- NIST FIPS 204 (ML-DSA): module-lattice digital signatures
- NIST deprecation timeline: deprecate RSA/ECC by 2030, disallow by 2035

**Tooling:**

- `pqcscan` (Anvil Security): https://github.com/anvilsecure/pqcscan — useful for scanning client environments
- Open Quantum Safe project: https://openquantumsafe.org

## Revision History

| Version | Date | Author | Notes |
|---|---|---|---|
| 1.0 | 2026-04-11 | Rick Cogley | Initial draft |
| 1.1 | 2026-04-11 | Rick Cogley | Added Primer section and Glossary; no longer assumes prior PQC knowledge. |

## Contact

**eSolia Inc.**
Shiodome City Center 5F (Work Styling)
1-5-2 Higashi-Shimbashi, Minato-ku, Tokyo, Japan 105-7105
**Tel (Main):** +813-4577-3380
**Web:** [https://esolia.co.jp/en](https://esolia.co.jp/en) **Preparer:** rick.cogley@esolia.co.jp