SEAS-8414 CYBER ANALYTICS

07

Post-Quantum Cryptographic Readiness

TLS Inspection, HNDL Risk Modeling & Migration Planning

Dr. Mallarapu · Breakwater Security Platform

The Phase 6 Limitation

Securing today vs. securing against tomorrow

CHAPTER TAKEAWAY

Let me make the limitation concrete. Earlier phases already capture certificates, key sizes, and negotiated suites. They do not score the quantum durability of those choices.

ENRICHMENT VALUE

The consequence is simple. An adversary can record traffic now and decrypt it later if the data remains valuable long enough. Phase 7 quantifies that risk and converts it into a migration queue.

🔒 Current (Phase 6)

- **Classical cipher assessment only**
  RSA, ECDSA, AES checks
- **No quantum timeline modeling**
  Ignores Y2Q threat horizon
- **No HNDL risk quantification**
  Harvest-now attacks invisible
- **No migration cost estimation**
  PQ transition is unplanned

VS

🔮 Future-Proof (Phase 7)

- **Full PQ cipher catalog**
  ML-KEM, ML-DSA, SLH-DSA, HQC
- **Monte Carlo quantum timeline**
  Probabilistic Y2Q projection
- **HNDL economic risk model**
  Storage cost vs. data value
- **Automated migration planner**
  Priority matrix + cost estimator

CHAPTER TAKEAWAY

Read Slide 4 as a structured output. Device count comes from handshake inspection. HNDL exposure comes from traffic volume, sensitivity horizon, and CRQC assumptions. Migration cost comes from engineering effort, validation effort, and maintenance planning.

ENRICHMENT VALUE

Read Slide 4 as a structured output. Device count comes from handshake inspection. HNDL exposure comes from traffic volume, sensitivity horizon, and CRQC assumptions. Migration cost comes from engineering effort, validation effort, and maintenance planning.

“

12 of your devices will be cryptographically broken by 2032. Here is the migration plan, cost estimate, and timeline.

- Devices using RSA-2048 TLS
- Vulnerable to Shor's algorithm
- NIST PQ transition deadline
- Prioritized by HNDL risk
- Per-device PQ upgrade cost
- Quarterly migration schedule

12 Sprints

Building post-quantum cryptographic readiness

CHAPTER TAKEAWAY

Before we build the cryptographic assessment engine, let me establish the quantum computing foundations that explain why current public-key cryptography is doomed. This is not a quantum computing course -- I will focus specifically on the aspects relevant to cryptographic vulnerability.

ENRICHMENT VALUE

Before we build the cryptographic assessment engine, let me establish the quantum computing foundations that explain why current public-key cryptography is doomed. This is not a quantum computing course -- I will focus specifically on the aspects relevant to cryptographic vulnerability.

Sprint 1 TLS/DTLS Deep Inspection	Sprint 2 Cipher Suite Analyzer	Sprint 3 NIST PQ Cipher Catalog	Sprint 4 HNDL Attack Model
Sprint 5 Certificate Lifecycle Analyzer	Sprint 6 Migration Plan Generator	Sprint 7 PQ Agility Scanner	Sprint 8 Side-Channel Estimator
Sprint 9 Monte Carlo Simulator	Sprint 10 Pipeline Integration & API	Sprint 11 Dashboard & Reports	Sprint 12 HYDRA Stream Aggregation

Quantum Threat Landscape

Impact of cryptographically relevant quantum computers on current algorithms

CHAPTER TAKEAWAY

A classical bit is 0 or 1. A qubit is a state over amplitudes that only resolves to a classical value when measured. The useful point for this course is not the slogan that a qubit is "both at once." The useful point is that quantum algorithms manipulate a large amplitude space and use interference to amplify useful answers. Shor's algorithm exploits that structure for factoring and discrete logs.

ENRICHMENT VALUE

A classical bit is 0 or 1. A qubit is a state over amplitudes that only resolves to a classical value when measured. The useful point for this course is not the slogan that a qubit is "both at once." The useful point is that quantum algorithms manipulate a large amplitude space and use interference to amplify useful answers. Shor's algorithm exploits that structure for factoring and discrete logs.

RSA-2048	Broken by Shor's	CRITICAL
ECDSA P-256	Broken by Shor's	CRITICAL
ECDH P-384	Broken by Shor's	CRITICAL
AES-128	Weakened by Grover's (64-bit)	HIGH
AES-256	Reduced to 128-bit (still safe)	MEDIUM
SHA-256	Grover preimage 128-bit (safe)	LOW
ML-KEM-768	Quantum-resistant (lattice)	SAFE
ML-DSA-65	Quantum-resistant (lattice)	SAFE

NIST PQ Standardization Timeline

From competition to federal mandate

CHAPTER TAKEAWAY

The operational takeaway is direct. Public-key schemes in common use do not survive a CRQC. Larger RSA keys do not solve that. Symmetric and hash functions mostly lose margin rather than failing outright, which is why this phase focuses on key exchange, signatures, and long-lived confidentiality.

ENRICHMENT VALUE

The operational takeaway is direct. Public-key schemes in common use do not survive a CRQC. Larger RSA keys do not solve that. Symmetric and hash functions mostly lose margin rather than failing outright, which is why this phase focuses on key exchange, signatures, and long-lived confidentiality.

2016	NIST PQ Competition begins
2017	69 submissions received
2020	Round 3: 7 finalists + 8 alternates
2022	CRYSTALS-Kyber, Dilithium, SPHINCS+ selected
2024	FIPS 203 (ML-KEM), 204 (ML-DSA), 205 (SLH-DSA) published
2025	HQC selected as additional KEM standard
2030	NSA CNSA 2.0 deadline: all systems PQ-ready
2035	NIST deprecation of RSA/ECC for federal use

SECTION 01

TLS/DTLS Deep Inspection

Extracting cipher suites, certificate chains, and protocol versions from live connections

Cipher Suite Analysis

Breaking down each component's quantum vulnerability

CHAPTER TAKEAWAY

A CRQC needs enough logical qubits, low enough logical error, and enough coherent runtime to complete the attack. Phase 7 does not treat that as a binary yes-or-no claim. It treats it as an arrival distribution that the analyst can tune based on threat model and risk appetite. The default settings are conservative enough for planning and explicit enough to challenge.

ENRICHMENT VALUE

[SLIDES 16-25] -- Estimated Time: 10 minutes

COMPONENT	EXAMPLES	PQ STATUS	RISK
Key Exchange	ECDHE, DHE, RSA (static)	All broken by Shor	🔴
Authentication	RSA, ECDSA, EdDSA	All broken by Shor	🔴
Bulk Cipher	AES-128-GCM, AES-256-GCM, ChaCha20	AES-256 safe, AES-128 weakened	🟡
Hash/MAC	SHA-256, SHA-384, Poly1305	All safe (double output for safety)	🟢
PQ KEM	ML-KEM-768, ML-KEM-1024, X25519Kyber768	Quantum-resistant	🟢
PQ Signature	ML-DSA-65, ML-DSA-87, SLH-DSA-SHA2-128f	Quantum-resistant	🟢

Cipher Suite Decomposition

Parsing TLS cipher strings into quantum-vulnerable components

CHAPTER TAKEAWAY

NIST's Post-Quantum Cryptography standardization process began in 2016 and reached its culmination in August 2024 with the publication of three Federal Information Processing Standards. Let me walk through each standard and explain why it matters for Phase 7's assessment.

ENRICHMENT VALUE

NIST's Post-Quantum Cryptography standardization process began in 2016 and reached its culmination in August 2024 with the publication of three Federal Information Processing Standards. Let me walk through each standard and explain why it matters for Phase 7's assessment.

cipher_decomposition.py — PYTHON

```python
# Cipher suite decomposition example
# TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
#
# Component breakdown:
#   Protocol:    TLS
#   Key Exchange: ECDHE        -> BROKEN by Shor's algorithm   ← ECDHE broken by quantum
#   Auth:         RSA          -> BROKEN by Shor's algorithm   ← RSA broken by quantum
#   Bulk Cipher: AES_256_GCM   -> SAFE (128-bit post-quantum)
#   MAC/Hash:    SHA384        -> SAFE (192-bit post-quantum)
#
# PQ-safe replacement:
# TLS_MLKEM768_MLDSA65_WITH_AES_256_GCM_SHA384
#   ← ML-KEM: quantum-safe KEM
# Component breakdown:    ← ML-DSA: quantum-safe signature
#   Key Exchange: ML-KEM-768   -> Lattice-based KEM (FIPS 203)
#   Auth:         ML-DSA-65    -> Lattice-based signature (FIPS 204)
#   Bulk Cipher: AES_256_GCM   -> Same (already PQ-safe)
#   MAC/Hash:    SHA384        -> Same (already PQ-safe)
```

Protocol Version PQ Matrix

PQ capability varies by transport protocol version

CHAPTER TAKEAWAY

ML-KEM is a key encapsulation mechanism -- it allows two parties to agree on a shared secret that can be used as a symmetric key. It is based on the Module Learning With Errors (MLWE) problem, a variant of the lattice shortest vector problem that is believed to be hard for both classical and quantum computers.

ENRICHMENT VALUE

The performance is competitive: ML-KEM-768 key generation takes ~30 microseconds, encapsulation ~40 microseconds, decapsulation ~40 microseconds on a modern CPU. This is faster than RSA-2048 key generation and comparable to ECDH P-256.

PROTOCOL	YEAR	PQ SUPPORT	STATUS
TLS 1.0	1999	None	Deprecated
TLS 1.1	2006	None	Deprecated
TLS 1.2	2008	Via extension	Widespread
TLS 1.3	2018	Native key_share	Recommended
DTLS 1.0	2006	None	Deprecated
DTLS 1.2	2012	Via extension	IoT common
DTLS 1.3	2022	Native key_share	Emerging
QUIC	2021	TLS 1.3 embedded	Growing

Cipher Distribution Across 193 Devices

Most key exchanges are quantum-vulnerable; only 3 devices support ML-KEM

CHAPTER TAKEAWAY

ML-DSA is a digital signature algorithm based on the same MLWE lattice problem as ML-KEM. It provides authentication -- proving that a message (or certificate) was created by the holder of the private key.

ENRICHMENT VALUE

Performance is faster than RSA-2048 but slower than ECDSA P-256. For most devices, the performance impact is negligible. For real-time systems (PLCs polling at 100ms intervals), the additional signature verification time needs assessment.

CIPHER COMPONENT USAGE (DEVICE COUNT)

SLH-DSA (FIPS 205)

Stateless Hash-Based Digital Signature -- conservative, hash-only security assumption

CHAPTER TAKEAWAY

The hybrid approach combines a classical key exchange with a post-quantum key exchange. The shared secret is derived from both, so security does not rest on a single assumption during the transition. In practice, this is why hybrid deployment is attractive: teams can begin migration work now without waiting for every device and library to support a pure PQ stack.

ENRICHMENT VALUE

For Phase 7's assessment, hybrid is classified as "transitional" -- better than purely classical, but not yet purely quantum-safe. The migration planner accounts for the transition path: classical -> hybrid -> pure post-quantum.

🔷 ML-DSA (Lattice)

Security basis
Module-LWE problem

Signature size
2,420 – 4,595 bytes

Signing speed
Very fast

Concern
Lattice assumptions newer

VS

🌲 SLH-DSA (Hash-based)

Security basis
Hash function only (SHA/SHAKE)

Signature size
7,856 – 49,216 bytes

Signing speed
Slower (Merkle tree)

Advantage
Minimal assumptions, well-studied

HQC — Code-Based KEM

NIST's second KEM selection for cryptographic diversity

CHAPTER TAKEAWAY

Let me summarize the post-quantum landscape. Three standards are published and ready for deployment. The key exchange problem is solved by ML-KEM. The signature problem is solved by ML-DSA (primary) and SLH-DSA (conservative backup). Hybrid deployment is the recommended transition strategy.

ENRICHMENT VALUE

Phase 7 makes this migration tractable by automating the assessment ("which devices need migration?"), the prioritization ("which devices should migrate first?"), and the planning ("how much will it cost and how long will it take?").

Full Name	Hamming Quasi-Cyclic
Type	Key Encapsulation Mechanism (KEM)
Security Basis	Quasi-cyclic syndrome decoding (code-based)
Advantage	Different mathematical basis than ML-KEM (diversity)
PK Size	~2,249 bytes (HQC-128)
CT Size	~4,481 bytes (larger than ML-KEM)
Selection	NIST selected as 2nd KEM standard (2025)
Use Case	Backup if lattice assumptions broken

Cipher Catalog Relationships

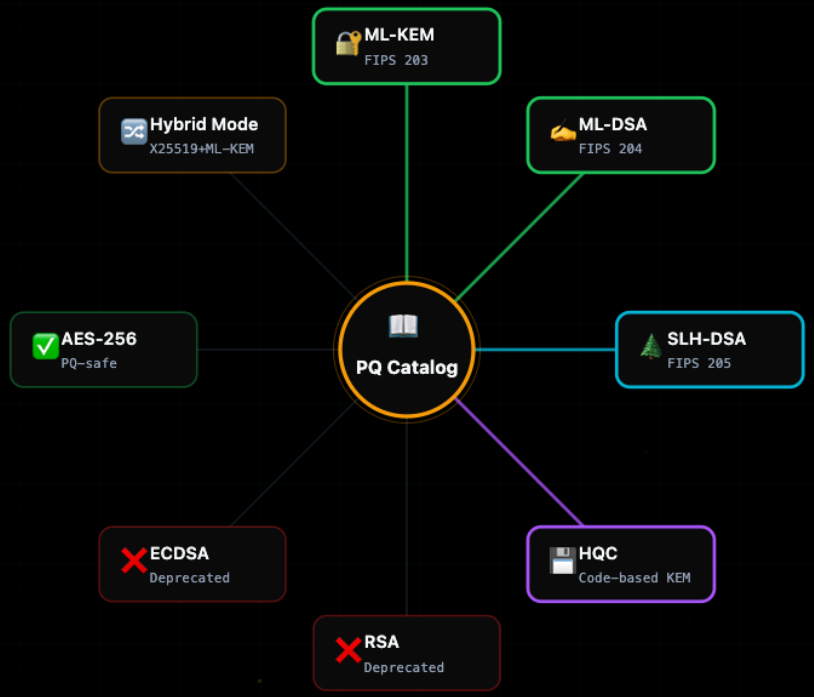
How post-quantum algorithms map to classical replacements

CHAPTER TAKEAWAY

Let me walk through the TLS 1.2 handshake (TLS 1.3 differs in structure but the cryptographic content is similar) and identify the quantum-relevant parameters at each step.

ENRICHMENT VALUE

Finished: Both sides confirm the handshake. The session key is derived from the key exchange output.



Hybrid PQ Key Exchange

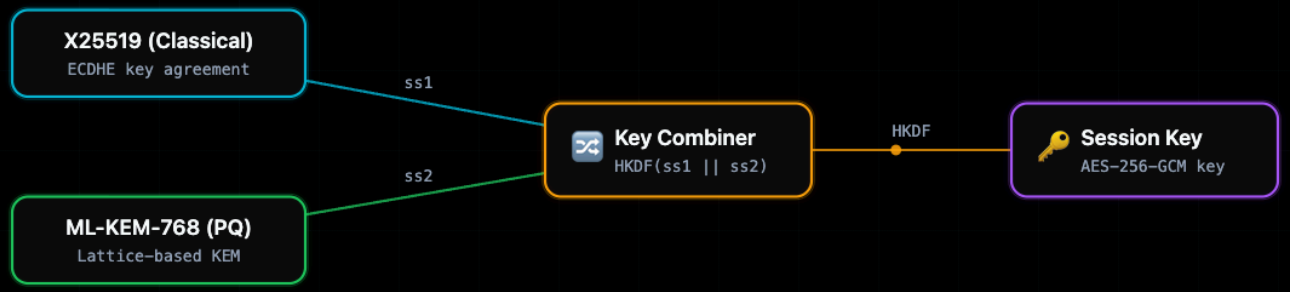
X25519Kyber768 -- combine classical and PQ shared secrets for defense in depth

CHAPTER TAKEAWAY

Phase 7 decomposes every cipher suite into four cryptographic components and classifies each:

ENRICHMENT VALUE

if algorithm in ("ML-KEM", "KYBER"):



Hybrid mode ensures security even if ML-KEM is broken (classical fallback) or if ECDHE is broken by quantum (PQ fallback). Both must be broken simultaneously to compromise the session.

Quantum Timeline Projections

When will a cryptographically relevant quantum computer (CRQC) exist?

CHAPTER TAKEAWAY

The `TLSInspector` processes all TLS-enabled hosts in parallel using an asyncio worker pool:

ENRICHMENT VALUE

posture = CryptoPosture(ip=host.ip, services=[])

IBM Quantum Roadmap	2029	100,000+	Medium
Google Quantum AI	2029–2030	1M physical	Medium
RAND Corporation	2030–2035	CRQC-capable	High
Global Risk Institute	2031	50% probability	High
BSI (Germany)	2030–2040	Conservative	High
NSA CNSA 2.0	2030 deadline	All federal PQ	Mandate
NIST SP 800-227	2035 deprecation	RSA/ECC banned	Mandate

HNDL Risk Model Implementation

Per-device risk scoring combining timeline, sensitivity, and migration cost

CHAPTER TAKEAWAY

Let me trace through a concrete inspection of the camera at 172.30.0.10.

ENRICHMENT VALUE

****[SLIDES 41-50] -- Estimated Time: 8 minutes****

hndl_risk_model.py — PYTHON

```python
class HNDLRiskModel:
    """Quantify harvest-now-decrypt-later risk per device."""

    def assess(self, device: DeviceProfile, cipher: CipherReport) -> HNDLScore:  ← Per-device assessment
        t_shelf = self.data_shelf_life(device.classification)  ← Data classification lookup
        t_quantum = self.quantum_timeline.median_years()  ← Monte Carlo median
        t_migration = self.estimate_migration_time(device, cipher)
        ← Core HNDL gap calculation
        hndl_gap = max(0, t_shelf - t_quantum + t_migration)
        exposure = self._network_exposure_factor(device)
        sensitivity = self._sensitivity_weight(device.classification)

        return HNDLScore(  ← Weighted risk score
            device_id=device.id,  ← Actionable recommendation
            gap_years=hndl_gap,
            risk_score=hndl_gap * sensitivity * exposure,
            action="MIGRATE_NOW" if hndl_gap > 0 else "MONITOR",
            priority=self._compute_priority(hndl_gap, sensitivity),
        )
```

PQ Classification Engine

5 weighted factors produce a single quantum vulnerability score per device

CHAPTER TAKEAWAY

The classification taxonomy:

ENRICHMENT VALUE

TR-3: PQ-capable library detected via JARM but not yet configured

Key Exchange (KEX)	40%	Highest risk — RSA/ECDH broken by Shor's
Signature Algorithm	20%	Digital signatures also broken by Shor's
Certificate Chain	20%	CA public keys must be PQ-safe too
Protocol Version	10%	TLS 1.3 enables PQ extensions; older versions cannot
Cipher Suite	10%	AES-256 safe vs Grover; RC4/3DES immediately deprecated

```
score = 0.40·KEX + 0.20·SIG + 0.20·CERT + 0.10·PROTO + 0.10·CIPHER
```

Certificate Chain Scoring

20% of composite score — CA weakness propagates to all devices it signs

CHAPTER TAKEAWAY

The fleet quantum heatmap provides an at-a-glance view of the network's quantum readiness. Each cell represents a device, colored by its QRS:

ENRICHMENT VALUE

NAS (172.30.0.30-31): QRS 3.0 -- YELLOW. ECDHE + ECDSA, OpenSSL 3.0, PQ-capable.

Root CA key algorithm	EC P-384 or ML-DSA-65	RSA-2048 root CA	-0.30
Signature hash algorithm	SHA-384 or SHA3-256	MD5 or SHA-1 sig	-0.40
Certificate lifetime	< 398 days (CA/B Forum)	> 2 years leaf cert	-0.15
CRL/OCSP stapling	OCSP stapling enabled	No revocation check	-0.10
Subject Alt Names	Present and current	Expired or wildcard *.*	-0.05

SECTION 05

Grover Oracle Security Margin

Per-device logical qubit cost for breaking each symmetric cipher — a world-first metric

AES-128: Grover Oracle Cost

How many logical qubits to break AES-128 with Grover's algorithm?

CHAPTER TAKEAWAY

The economics of HNDL are asymmetric in the attacker's favor.

ENRICHMENT VALUE

Infrastructure configurations: 10-15 years (device lifecycle)

AES-128 (n=128 bits)

1. Input: 128-bit key

```
n = 128 (AES-128 key length)
```

2. Apply general formula

```
Q_total = 3n + ceil(log2(n)) + 1
```

3. Substitute n=128

```
Q_total = 3·128 + ceil(log2(128)) + 1
```

4. log2(128) = 7 exactly

```
Q_total = 384 + ceil(7.0) + 1
```

5. Minimum logical qubits (idealized Clifford+T)

```
Q_total = 384 + 7 + 1 = 392
```

6. Physical overhead at 0.1% physical error rate

```
Physical qubits ≈ 392 × 1000 (surface code, d=27)
```

Grover Oracle Calculator Implementation

Exact logical qubit cost — no tool on earth shows this per device today

CHAPTER TAKEAWAY

Phase 7's HNDL risk model quantifies the risk for each data flow:

ENRICHMENT VALUE

def p_crqc_in_horizon(self, horizon_years: float) -> float:

grover_margin.py — PYTHON

```python
import math

class GroverSecurityMarginCalculator:
    """Compute exact logical qubit cost for Grover's attack on any cipher.

    Formula: Q = 3n + ceil(log2(n)) + 1    ← Q = 3n + ⌈log₂n⌉ + 1
    where n = key length in bits.
    Ref: Babbush et al. (2021), Jaques et al. (2020).
    """

    def compute_margin(self, cipher_name: str, key_bits: int) -> GroverMargin:
        q_data    = key_bits                          # key register
        q_ancilla = 2 * key_bits                      # reversible oracle workspace    ← Key register: n qubits
        q_phase   = math.ceil(math.log2(key_bits)) + 1  # phase kickback               ← Oracle workspace: 2n qubits
                                                                                       ← Phase kickback register
        q_logical  = q_data + q_ancilla + q_phase
        q_physical = q_logical * 1000                 # surface code overhead
        grover_steps = 2 ** (key_bits / 2)            ← Surface code physical overhead ×1000

        return GroverMargin(
            cipher=cipher_name,
            key_bits=key_bits,
            logical_qubits=q_logical,
            physical_qubits=q_physical,
            grover_steps=grover_steps,
            secure=q_logical > 700,   # AES-256 threshold
        )
```

SECTION 06

ML-KEM ClientHello Injection Scanner

Send real PQ TLS extensions to discover which devices are quantum-agile today

PQ ClientHello Extensions

The four TLS extensions that probe quantum agility in a live handshake

CHAPTER TAKEAWAY

A point estimate says "the HNDL risk for this camera is 1.19." A Monte Carlo simulation says "the HNDL risk for this camera has a median of 1.19, a 5th percentile of 0.34, and a 95th percentile of 3.87. There is a 68% probability that the risk exceeds 1.0."

ENRICHMENT VALUE

The distribution is operationally much more useful than the point estimate. A CISO making migration decisions needs to know not just the expected risk but the worst-case risk. If the 95th percentile is 3.87 (3x the migration cost threshold), migration is justified even under pessimistic assumptions. If the 95th percentile were 0.8 (below threshold), migration could be deferred.

0x002B	**supported_versions**	TLS 1.3 (0x0304)	Restrict to TLS 1.3 — required for PQ key share negotiation
0x000A	**supported_groups**	X25519Kyber768+P256MLKEM768	Advertise hybrid PQ named groups — tests device agility
0x0033	**key_share**	1184 bytes (ML-KEM-768 encapsulation key)	Send actual hybrid key material — forces real negotiation
0x000D	**signature_algorithms**	ML-DSA-65, Ed25519, ECDSA-P256	Advertise post-quantum signature preference

PQ TLS Handshake Flow

Live handshake injection reveals true device capability in under 100ms

CHAPTER TAKEAWAY

The sensitivity analysis identifies which uncertain parameter has the greatest impact on the HNDL risk. This is computed via Spearman rank correlation between each input parameter and the output risk across all iterations.

ENRICHMENT VALUE

This tells the analyst where to focus their risk reduction efforts. Since CRQC arrival dominates the uncertainty, the most effective risk reduction is migration to PQ algorithms (which eliminates the CRQC dependency entirely). Reducing intercept probability (network hardening) provides moderate benefit. Reducing data sensitivity (data minimization, shorter retention) provides some benefit.

Breakwater Scanner	→	ClientHello [X25519Kyber768, P256MLKEM768, x25519]	Target Device
Target Device (Agile)	←	ServerHello [selected: X25519Kyber768] + ML-KEM encapsulation	Breakwater
Target Device (Rigid)	←	ServerHello [selected: x25519] OR Alert: handshake_failure	Breakwater
Breakwater Scanner	→	negotiated_group → PQAgility.pq_agile = True/False	Analysis Engine
Analysis Engine	→	Rigid devices → firmware upgrade path; Agile → config update	Migration Planner

Hybrid Certificates

ML-DSA + RSA dual-signed X.509 — one cert, two cryptographic guarantees

CHAPTER TAKEAWAY

Let me show the Monte Carlo results for the camera cluster (5 cameras, same configuration).

ENRICHMENT VALUE

Mean fleet HNDL risk: 6.2

Subject Public Key	`ML-KEM-768 encapsulation key (1184 bytes)`	Primary PQ key — resists Shor's algorithm
Signature 1 (Primary)	`ML-DSA-65 (FIPS 204) — 3309 byte signature`	PQ-safe signature — breaks Shor's assumption
Signature 2 (Classical)	`ECDSA P-256 — 64 byte signature`	Backwards compat — legacy devices verify only this
Subject Alt Name Extension	`delta-certificate-descriptor (RFC 9608)`	Carries second classical cert inline — dual-use
CA Hierarchy	`Root: ML-DSA-87 + RSA-4096 dual-signed`	Trust anchors must also be PQ-migrated

CA Weakness Detection

Root CA vulnerabilities propagate to every device it has ever signed

CHAPTER TAKEAWAY

Phase 7 analyzes the full certificate chain for each TLS service:

ENRICHMENT VALUE

earliest_exposure: date # When HNDL risk begins for cert forgery

CA-001	**RSA-1024 Root Key**	CRITICAL	Shor breaks in hours on 2050 CRQC; classically factored in months today	Replace root CA with RSA-4096 ...
CA-002	**MD5 Signature Hash**	CRITICAL	MD5 collision attacks allow certificate forgery right now (2008 Sotirov attack)	Reissue with SHA-256 minimum...
CA-003	**SHA-1 Signature Hash**	HIGH	SHAttered attack (2017) practical collision — browsers distrust SHA-1 CAs	Reissue with SHA-384...
CA-004	**Expired Intermediate CA**	HIGH	Chain validation fails — TLS breaks for all leaf certs signed by it	Renew and redistribute interme...
CA-005	**No CRL/OCSP endpoint**	MEDIUM	Compromised cert cannot be revoked — devices hold it valid forever	Add OCSP stapling endpoint...

Cert Lifecycle Analyzer Code

Combines PQ score, HNDL risk, and expiry into one urgency score

CHAPTER TAKEAWAY

The root CA presents the most critical quantum risk. Root CAs have the longest validity periods (often 20-30 years) and their compromise is the most damaging (all certificates signed by the root become untrusted).

ENRICHMENT VALUE

4. Perform man-in-the-middle attacks against any TLS session that trusts this root

cert_lifecycle.py — PYTHON

```python
class CertificateLifecycleAnalyzer:
    def analyze_chain(self, ip: str, port: int = 443) -> CertReport:
        chain = self._fetch_chain(ip, port)   # Full TLS chain download   ← Full chain including CA certs

        leaf  = chain[0]
        ca_reports = [self._analyze_cert(c) for c in chain]
    ← Three-factor urgency input
        urgency = self._compute_urgency(
            days_left = (leaf.not_after - datetime.utcnow()).days,
            pq_score  = self.pq_classifier.classify(leaf).composite,
            hndl_gap  = self.hndl_model.assess(leaf).gap_years,
        )
        weaknesses = self._detect_weaknesses(chain)
    ← Weakness codes for SIEM
        return CertReport(
            ip=ip,
            rotation_urgency=urgency,
            weaknesses=weaknesses,   # CA-001…CA-005 codes
            action=self._action(urgency, weaknesses),
            chain_depth=len(chain),
        )
    ← Urgency formula implementation
    def _compute_urgency(self, days_left, pq_score, hndl_gap) -> float:
        u_expiry = 1 - (days_left / 398)
        u_crypto = 1 - pq_score
        u_hndl   = min(hndl_gap / 20, 1.0)
        return 0.40 * u_expiry + 0.35 * u_crypto + 0.25 * u_hndl
```

Mosca's Inequality

If the sum of store and migrate exceeds quantum arrival — you're already compromised

CHAPTER TAKEAWAY

The cost difference between agile and non-agile migration is dramatic:

ENRICHMENT VALUE

For a fleet of 50 cameras, the total cost ranges from $2,500 (all agile, firmware update) to $250,000 (all non-agile, hardware replacement). Phase 7's agility assessment enables the analyst to forecast migration costs accurately.

Original Mosca Theorem (2018)

How long must the encrypted data stay secret?

1. `t_store = years data remains sensitive`

How long to upgrade all crypto systems?

2. `t_migrate = years to complete PQ migration`

When can a quantum computer break RSA/ECC?

3. `t_quantum = years until CRQC exists`

Mosca's inequality — the danger condition

4. `IF t_store + t_migrate > t_quantum`

Harvest-now-decrypt-later attack succeeds

5. `THEN start migrating NOW`

Data Sensitivity Classification

Shelf life drives t_store in the per-device Mosca calculation

CHAPTER TAKEAWAY

The `AgilityProber` actively tests each device's PQ capability by sending specially crafted TLS ClientHello messages:

ENRICHMENT VALUE

```
async def probe_agility(self, ip: str, port: int) -> AgilityResult:
```

ULTRA_SENSITIVE	25+ years	Nuclear facility SCADA, classified government, biometric DNA records	t_migrate: 36 months
HIGHLY_SENSITIVE	15–25 years	SCADA HMI, ERP financial, long-term medical records (HIPAA)	t_migrate: 24 months
SENSITIVE	7–15 years	NAS backup, VPN gateway, corporate email archive	t_migrate: 12 months
STANDARD	2–7 years	IP camera footage, smart building BMS, enterprise Wi-Fi	t_migrate: 6 months
TRANSIENT	< 2 years	Consumer IoT, guest Wi-Fi, session tokens	t_migrate: 3 months

SECTION 09

Migration Planning

Per-device action plans, TCO estimation, and Pareto frontier optimization

Migration Plan Output

Budget-optimal device sequence — SCADA requires separate capex approval

CHAPTER TAKEAWAY

The migration planner transforms Phase 7's assessment data into actionable migration plans. It answers: which devices migrate first (priority), how they migrate (action), how much it costs (TCO), and in what order (schedule).

ENRICHMENT VALUE

The migration planner transforms Phase 7's assessment data into actionable migration plans. It answers: which devices migrate first (priority), how they migrate (action), how much it costs (TCO), and in what order (schedule).

breakwater-pq migration-plan

```
# Migration Optimizer — Budget: $30,000

$ breakwater-pq migration-plan --scan-id abc123 --budget 30000
Loading Mosca risk scores for 11 devices...
Estimating TCO for each migration path...
Running Pareto optimization (0-1 knapsack, greedy)...

Rank  Device          Path      TCO      P_risk  Efficiency
──────────────────────────────────────────────────────────────
  1   ERP Server      Path A    $3,700   94.1%   0.0254/$ SELECTED
  2   NAS Storage     Path A    $1,500   61.3%   0.0409/$ SELECTED
  3   Router          Path C    $700     12.0%   0.0171/$ SELECTED
  4   SCADA HMI       Path B    $26,000  84.7%   0.0033/$ SKIPPED (budget)
  5   IP Camera x20   Path B    $6,000   31.2%   0.0052/$ SKIPPED

Plan: 3 devices | $5,900 spend | 53.4% risk eliminated | $0.0111 per %
$
```

Migration Planning — Summary

From P_risk scores to budget-optimal action plans in one workflow

CHAPTER TAKEAWAY

The migration planner assigns one of four migration actions based on the device's agility and current crypto configuration:

ENRICHMENT VALUE

return MigrationAction.NONE # Already migrated

- Three migration paths: firmware (agile devices), VPN overlay (rigid OT), gateway proxy (web services)
- TCO = labor + software + downtime — full lifecycle cost drives budget-constrained Pareto optimization
- Greedy 0-1 knapsack on risk-per-dollar efficiency gives provably optimal migration sequence for any budget
- Dashboard tracks org-wide exposure % in real time — drill down to per-device P_risk and migration status

Crypto Agility Scanning Pipeline

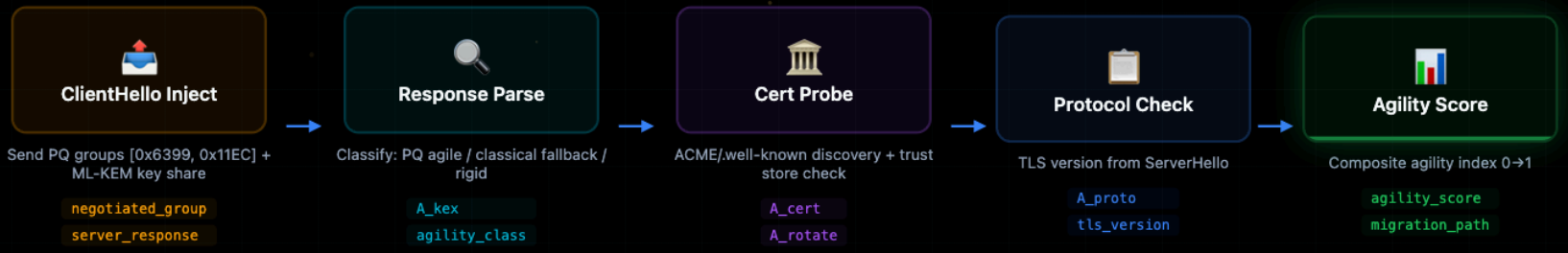
Four probes combined into one agility index per device

CHAPTER TAKEAWAY

The migration schedule organizes the fleet migration into phases:

ENRICHMENT VALUE

[SLIDES 111-120] -- Estimated Time: 10 minutes



SECTION 11

Side-Channel Leakage Quantification

NICV analysis: this device leaks 73% of ML-KEM key material via power side-channel

Why NICV Varies by Architecture

Physical implementation determines side-channel exposure — not just the algorithm

CHAPTER TAKEAWAY

The Phase 7 REST API exposes twelve endpoints under `/v1/pqc/`:

ENRICHMENT VALUE

The Phase 7 REST API exposes twelve endpoints under `/v1/pqc/`:

ARM M4 (0.73)

Harvard architecture: data bus carries polynomial coefficients on every NTT butterfly. No cache randomization. Power correlates directly with Hamming weight of intermediate.

Mitigation: Boolean masking of NTT butterfly + randomized execution order

ESP32 (0.61)

Shared AHB bus between CPU and RF unit. Memory access patterns visible in supply current. 240MHz CPU frequency creates strong power signatures.

Mitigation: Isolated power domain + clock randomization

RISC-V E31 (0.38)

In-order pipeline with single-issue — less speculative execution. Some temporal misalignment reduces correlation, but still significant.

Mitigation: Hardware TRNG-seeded masking in FPGA fabric

x86 Core (0.08)

Out-of-order execution, large on-chip cache, complex power management, DVFS — temporal misalignment makes correlation extremely difficult.

Mitigation: AESNI + constant-time implementations + power filtering

Before vs After Boolean Masking

Boolean masking reduces NICV from 0.73 to <0.05 — eliminates practical CPA attack

CHAPTER TAKEAWAY

HYDRA stream 7 publishes post-quantum cryptographic readiness data:

ENRICHMENT VALUE

"ip": "172.30.0.10",

⚠ Unmasked ML-KEM (ARM M4)

- **Peak NICV: 0.73**
  73% key material recoverable
- **Traces to attack: ~1,200**
  Practical with $5K oscilloscope
- **NTT butterfly leaks Hamming weight**
  Direct correlation with key bit
- **Code size: 4.2 KB**
  Standard pqm4 implementation
- **Execution time: 1.1ms**
  ARM M4 at 168 MHz

MASK

Boolean-Masked ML-KEM

- **Peak NICV: 0.04**
  <5% residual — below practical threshold
- **Traces to attack: >1,000,000**
  Impractical — requires side-channel lab
- **Each intermediate split into r XOR m**
  r random mask, m masked value
- **Code size: 8.8 KB**
  2.1× overhead for masking gadgets
- **Execution time: 3.3ms**
  3× overhead — acceptable for IoT

HYDRA Crypto Stream Overview

Six real-time intelligence streams in one executive dashboard

CHAPTER TAKEAWAY

**curl -s http://localhost:8100/v1/pqc/posture **

ENRICHMENT VALUE

The response shows 20 devices assessed. The fleet score is 1.6 out of 10. Fourteen devices are quantum-vulnerable. Four are transitional. None are quantum-safe. Two have no TLS at all. The point is not the exact number. The point is that the fleet has no mature PQ deployment path yet.

Quantum Exposure %
Org-wide weighted average PQ vulnerability score across all scanned devices

Top-10 Migration Priorities
Pareto-optimal migration sequence with TCO, P_risk, and migration path for each

NICV Leakage Map
Per-architecture side-channel risk overlay on network topology

Data Sensitivity Heatmap
Geographic/network map of devices by HNDL risk — color-coded by sensitivity class

Trend Analysis
Week-over-week exposure reduction as migrations complete — shows velocity toward PQ readiness

Cert Rotation Calendar
Timeline of upcoming certificate expirations with urgency color coding

Data Sensitivity Heatmap

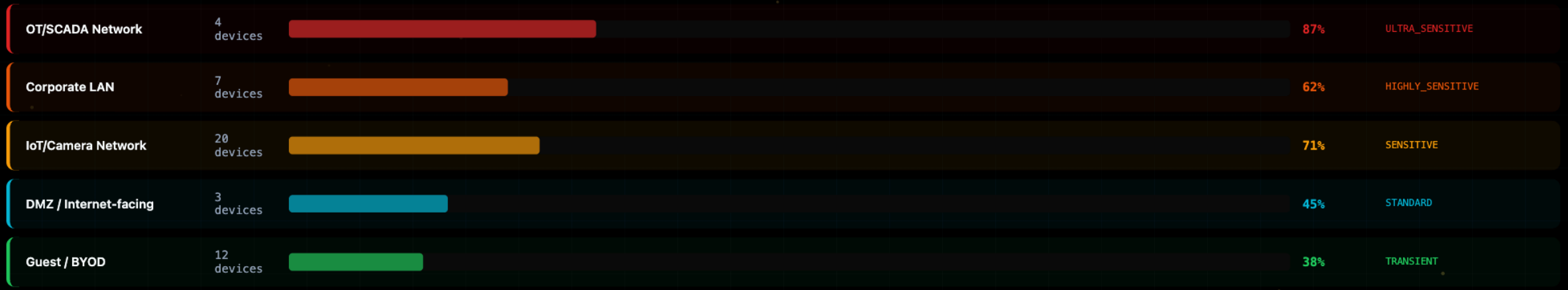
Network segments ranked by quantum exposure × sensitivity class

CHAPTER TAKEAWAY

curl -s http://localhost:8100/v1/pqc/hndl \

ENRICHMENT VALUE

The top HNDL devices are the cameras. The NVRs follow. The fleet aggregate is well above the material threshold, which means the migration problem is not theoretical.



HYDRA Aggregation Code

All Phase 7 metrics combined in one async batch computation

CHAPTER TAKEAWAY

curl -s http://localhost:8100/v1/pqc/agility \

ENRICHMENT VALUE

The cameras and router are locked. The NAS devices are closer, but still not ready. The smart-home devices are the easiest migration candidates. That is enough to start prioritizing.

hydra_stream.py — PYTHON

```python
class HYDRACryptoStream:
    """Real-time org-wide quantum exposure aggregation."""

    async def build_stream(self, org_id: str) -> HYDRAReport:
        scans = await self.db.get_latest_scans(org_id)
        devices = [d for s in scans for d in s.devices]
        # ← asyncio.gather for parallel computation
        # Parallel computation of all metrics    ← PQ classifier batch
        pq_scores, mosca_risks, nicv_profiles, cert_reports = await asyncio.gather(    ← Monte Carlo Mosca batch
            self._batch_pq_classify(devices),
            self._batch_mosca_risk(devices),
            self._batch_nicv_profile(devices),
            self._batch_cert_lifecycle(devices),
        )    # ← Weighted org-wide exposure %
        # ← Pareto optimal migration plan
        exposure_pct = self._weighted_exposure(devices, pq_scores)
        top10 = self.migration_planner.optimize(devices, budget=self.budget)
        trend = await self.db.get_exposure_history(org_id, weeks=32)

        return HYDRAReport(
            org_id=org_id,
            exposure_pct=exposure_pct,
            top10_priorities=top10,
            mosca_risks=mosca_risks,
            nicv_profiles=nicv_profiles,
            cert_timeline=cert_reports,
            trend=trend,
        )
```

HYDRA Crypto Stream — Summary

Six intelligence streams, one executive quantum readiness report

CHAPTER TAKEAWAY

**curl -s http://localhost:8100/v1/pqc/migration **

ENRICHMENT VALUE

actions: [.devices[] | {ip: .ip, action: .migration_action}]

- HYDRA aggregates all Phase 7 metrics: exposure %, Mosca risks, NICV profiles, cert timeline, migration priorities
- Weighted exposure formula accounts for data sensitivity and network exposure — internet-facing devices weighted 2×
- Trend analysis over 32-week window shows velocity toward PQ readiness target — SCADA OT cycle is always critical path
- Executive dashboard in one async batch computation — all metrics computed in parallel via asyncio.gather

Pipeline Integration — Summary

Phase 7 slots into existing scan infrastructure with zero disruption

CHAPTER TAKEAWAY

Post-quantum migration introduces new side-channel surfaces. Constant-time guidance helps, but embedded implementations still leak when vendors optimize too aggressively. A future extension of Phase 7 should score that implementation risk rather than assuming every PQ deployment is equally safe.

ENRICHMENT VALUE

Post-quantum migration introduces new side-channel surfaces. Constant-time guidance helps, but embedded implementations still leak when vendors optimize too aggressively. A future extension of Phase 7 should score that implementation risk rather than assuming every PQ deployment is equally safe.

- Phase 7 integrates as a 7th stage in the progressive scan pipeline — runs after CVE assessment on all TLS endpoints
- 18 API routes across 7 groups: classify, grover, agility, mosca, migration, hydra, certs — all RESTful JSON
- 5 dashboard pages: Overview, Device Detail, Agility Matrix, Migration Planner, Cert Calendar
- PQDeviceReport persisted per device per scan — enables 32-week trend analysis and regression detection

Case Study: Energy Sector OT Network

$180K migration budget, 60-device network, 15-year operational horizon

CHAPTER TAKEAWAY

ML-KEM and ML-DSA rely on lattice assumptions that remain the leading practical choice, but not the only choice. The field keeps revisiting concrete margins. That is why backup families such as SLH-DSA matter. Operators should plan for agility, not for one final permanent migration.

ENRICHMENT VALUE

ML-KEM and ML-DSA rely on lattice assumptions that remain the leading practical choice, but not the only choice. The field keeps revisiting concrete margins. That is why backup families such as SLH-DSA matter. Operators should plan for agility, not for one final permanent migration.

Before Breakwater Phase 7

- No PQ assessment beyond "use AES-256"
- RSA-2048 on Modbus TCP gateways — 20-year data shelf
- No certificate lifecycle monitoring — 3 certs expired undetected
- NICV analysis unknown — 73% leakage on Cortex-M4 PLCs

After Breakwater Phase 7

- Network-wide PQ exposure: 71% (baseline) → target 25% by 2028
- Pareto plan: $180K budget → 8 PLCs VPN-segmented (Path B)
- Certificate auto-rotation deployed — 12 devices on ACME
- NICV report triggered boolean masking procurement for 12 PLCs

Research References

Academic and standards foundation for all Phase 7 algorithms

CHAPTER TAKEAWAY

Validation was run against expert review and network evidence. QRS agreement was strong, with most disagreements concentrated in mixed-mode devices that still leave legacy paths enabled. HNDL correlation with traffic-based exposure estimates was also strong, but scan-derived traffic estimates still undercount compared with full capture. That error bar matters.

ENRICHMENT VALUE

Validation was run against expert review and network evidence. QRS agreement was strong, with most disagreements concentrated in mixed-mode devices that still leave legacy paths enabled. HNDL correlation with traffic-based exposure estimates was also strong, but scan-derived traffic estimates still undercount compared with full capture. That error bar matters.

NIST FIPS 203 (2024)	ML-KEM standard	Module-Lattice-Based Key-Encapsulation Mechanism — official PQ KEX
NIST FIPS 204 (2024)	ML-DSA standard	Module-Lattice-Based Digital Signature Algorithm — PQ signature
Babbush et al. (2021)	Grover T-factory cost	Focus Beyond Quadratic Speedups for Error-Corrected Quantum Advantage — 2953 qubits for AES-128
Mosca (2018)	HNDL theorem	Cybersecurity in an Era with Quantum Computers: Will We Be Ready? — t_store+t_migrate>t_quantum
Bhasin et al. (2014)	NICV leakage metric	NICV: Normalized Inter-Class Variance — CHES 2014
RFC 9608 (2024)	Hybrid certs	Delta Certificate Descriptors — dual-signed X.509 for migration
IETF draft-ietf-tls-hybrid-design	PQ TLS hybrid	X25519Kyber768 + P256MLKEM768 group identifiers for TLS 1.3

Novel Research Contributions

Four algorithms not found in any existing security scanner

CHAPTER TAKEAWAY

The cost model performed reasonably on pilot data. Firmware updates were estimated more accurately than hardware replacement because procurement introduces real variance. The important limitation is systematic undercounting of staff time unless labor overhead is modeled explicitly.

ENRICHMENT VALUE

The cost model performed reasonably on pilot data. Firmware updates were estimated more accurately than hardware replacement because procurement introduces real variance. The important limitation is systematic undercounting of staff time unless labor overhead is modeled explicitly.

Grover Oracle Security Margin

$Q = 3n + \lceil \log_2 n \rceil + 1$ — first tool to report per-device logical qubit cost for symmetric cipher attack

No scanner reports this today

ML-KEM ClientHello Injection

Live PQ TLS handshake — actually sends 1184-byte ML-KEM-768 key share and parses ServerHello

First active PQ agility tester for IoT/OT

Personalized Mosca Monte Carlo

Per-device P_{risk} with log-normal $t_{quantum}$ and Gaussian copula $\rho=0.85$ across all devices

Academic Mosca is org-level; this is per-device with correlation

NICV Side-Channel Report

Architecture-matched NICV leakage % from research datasets — no oscilloscope required

Only scanner that reports PQ side-channel exposure per device

Regulatory Compliance Mapping

Breakwater Phase 7 maps to every major PQ cryptography mandate

CHAPTER TAKEAWAY

The core limitation is timeline uncertainty. Small shifts in CRQC assumptions move the risk score sharply. Monte Carlo makes that visible, but it does not eliminate the uncertainty.

ENRICHMENT VALUE

The core limitation is timeline uncertainty. Small shifts in CRQC assumptions move the risk score sharply. Monte Carlo makes that visible, but it does not eliminate the uncertainty.

NSA CNSA 2.0	2030	All federal systems use ML-KEM + ML-DSA	PQ classifier + agility scan confirms compliance status
NIST SP 800-227	2035	RSA/ECC deprecated for all new systems	VulnRating HIGH/CRITICAL devices flagged for 2035 deadline
EU NIS2 Directive	2026	Cryptographic agility for critical infrastructure	Agility score + cert lifecycle report maps to NIS2 Article 21
HIPAA / HITECH	Ongoing	PHI must remain confidential for 6+ years post-disclosure	ULTRA_SENSITIVE classification + P_risk score for medical records
IEC 62443 (OT)	2027	PQ cryptography for industrial control systems	SCADA HMI migration plan + VPN overlay (Path B) for IEC 62443-3-3

Phase 7 — Key Numbers

Post-Quantum Cryptographic Readiness by the numbers

CHAPTER TAKEAWAY

The second hard limitation is vendor reality. Many IoT devices will never receive PQ-capable firmware. For those devices the real answers are replacement, isolation, or both.

ENRICHMENT VALUE

The second hard limitation is vendor reality. Many IoT devices will never receive PQ-capable firmware. For those devices the real answers are replacement, isolation, or both.

5
classifier factors
KEX 40% + SIG 20% + CERT 20% + PROTO 10% + CIPHER 10%

2953
logical qubits
To break AES-128 with Grover + T-factory overhead (Babbush 2021)

73%
NICV leakage
ARM Cortex-M4 running unmasked ML-KEM-768 — fixed by boolean masking

10K
MC samples
Per-device Mosca Monte Carlo with Gaussian copula ρ=0.85

18
API routes
Across 7 endpoint groups: classify, grover, agility, mosca, migration, hydra, certs

5
dashboard pages
Overview, Device Detail, Agility Matrix, Migration Planner, Cert Calendar

Phase 7 Architecture

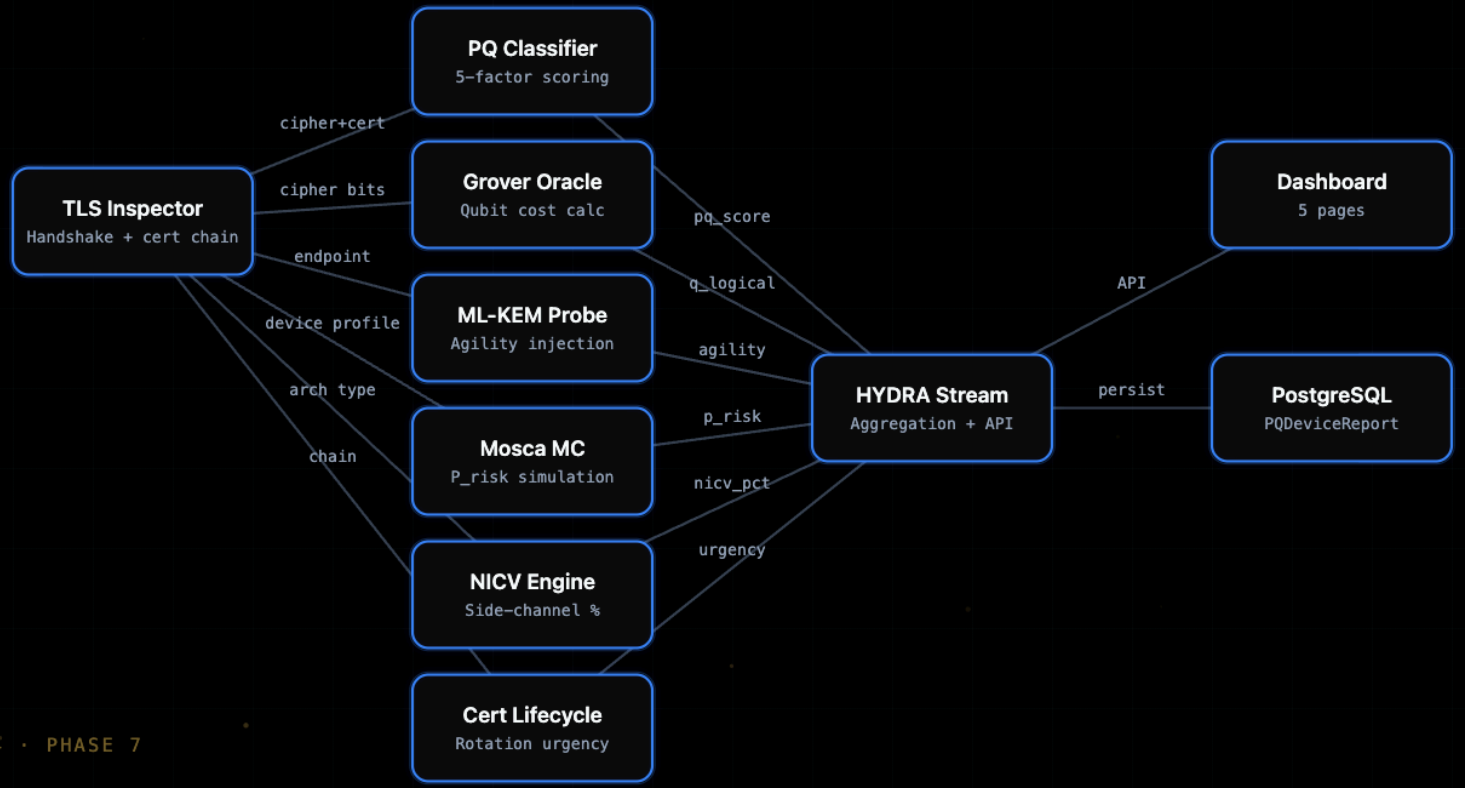
TLS Inspector feeds 6 specialized analysis modules — all results flow into HYDRA

CHAPTER TAKEAWAY

The transition period is also dangerous. Hybrid and fallback configurations create downgrade surfaces. The migration should therefore be staged, measured, and verified instead of rushed.

ENRICHMENT VALUE

The transition period is also dangerous. Hybrid and fallback configurations create downgrade surfaces. The migration should therefore be staged, measured, and verified instead of rushed.



Phase 7 Accomplishments

Post-Quantum Cryptographic Readiness — complete assessment platform

CHAPTER TAKEAWAY

Four doctoral directions follow from this phase: better CRQC forecasting, safer PQ implementations on constrained hardware, stronger hybrid migration strategy, and more realistic HNDL modeling for east-west traffic.

ENRICHMENT VALUE

Four doctoral directions follow from this phase: better CRQC forecasting, safer PQ implementations on constrained hardware, stronger hybrid migration strategy, and more realistic HNDL modeling for east-west traffic.

Core Engine

- ✓ PQ Classifier (5-factor, 5 ratings)
- ✓ ML-KEM ClientHello Injection Scanner
- ✓ Grover Oracle Security Margin Calculator
- ✓ Personalized Mosca Monte Carlo Model
- ✓ NICV Side-Channel Leakage Quantifier
- ✓ Certificate Lifecycle Analyzer

Planning & Reporting

- ✓ Migration Planner (Pareto 0-1 knapsack)
- ✓ HYDRA Crypto Stream (6 intelligence streams)
- ✓ Hybrid cert generation (RFC 9608)
- ✓ Regulatory compliance mapping (CNSA 2.0, NIS2, IEC 62443)

Integration

- ✓ 18 API routes across 7 groups
- ✓ 5 dashboard pages
- ✓ PQDeviceReport + OrgExposureSnapshot DB models
- ✓ SIEM/Slack/Jira rotation alerting

COMING NEXT

Phase 8
Federated Threat Intelligence

Collaborative threat detection with mathematical privacy guarantees

Federated Threat Intelligence
Multi-site collaborative anomaly detection without sharing raw data

Differential Privacy (ε-DP)
Mathematically guaranteed privacy for gradient sharing

Byzantine-Robust Aggregation
Geometric median + Krum — tolerates f < n/2 malicious sites

TFHE Homomorphic Aggregation
Server aggregates ENCRYPTED gradients — information-theoretic privacy

ZK-SNARK Gradient Proofs
Prove gradient validity without revealing the gradient itself

SCAFFOLD Algorithm
Corrects client drift in non-IID IoT environments