WHITE PAPER

PQCS L+ / MCP

Post-Quantum Cryptographic Solution

Lattice+ Model Context Protocol

Digital Identity Infrastructure for the Quantum Era

KEM Algorithm HQC-KEM 796 (NIST Level 3)	**Digital Signature** CRYSTALS-Dilithium3 (FIPS 204)
Protocol Layer Model Context Protocol (MCP)	**Version** 1.0 | June 2025

Executive Summary

The advent of cryptographically relevant quantum computers (CRQCs) poses an existential threat to the public-key infrastructure that underpins modern digital identity, secure communications, and AI-model orchestration. Classical algorithms such as RSA-2048 and ECDSA-256, which currently secure billions of daily transactions, are rendered vulnerable by Shor's algorithm running on sufficiently large fault-tolerant quantum processors.

PQCS L+ / MCP (Post-Quantum Cryptographic Solution — Lattice Plus, with Model Context Protocol integration) is a comprehensive framework designed to future-proof digital identity infrastructure before CRQCs become operationally viable. The architecture fuses two complementary post-quantum primitives:

- HQC-KEM 796 — a code-based Key Encapsulation Mechanism derived from Hamming Quasi-Cyclic codes, providing quantum-resistant key exchange at NIST Security Level 3 (comparable to AES-192).
- CRYSTALS-Dilithium3 — a lattice-based digital signature scheme standardised as FIPS 204, providing quantum-resistant authentication and non-repudiation at NIST Security Level 3.

These primitives are integrated into a secure Model Context Protocol transport layer, enabling AI agents and identity-aware services to exchange contexts, credentials, and attestations in a post-quantum secure manner.

This white paper details the mathematical foundations, implementation architecture, security analysis, performance benchmarks, compliance mapping, and deployment guidance for PQCS L+/ MCP. The framework targets government Digital ID programmes, regulated financial institutions, healthcare identity providers, and any organisation required to achieve crypto-agility in anticipation of NIST's post-quantum standards (FIPS 203/204/205).

Key Claim: PQCS L+ / MCP achieves ≥2^192 quantum security strength for both key encapsulation
and digital signatures whilst remaining compatible with existing X.509 PKI infrastructure,
MCP-based AI orchestration systems, and W3C Decentralised Identifier (DID) standards.

Table of Contents

1. Introduction & Threat Landscape

1.1 The Quantum Computing Threat

In 1994, Peter Shor demonstrated that a quantum computer can factor large integers and compute discrete logarithms in polynomial time. Since all widely deployed public-key schemes — RSA, DSA, ECDSA, DH, ECDH — base their security on exactly these hard problems, a sufficiently powerful quantum computer would break them completely.

Current estimates from NIST, NSA, and BSI place the advent of a Cryptographically Relevant Quantum Computer (CRQC) capable of breaking RSA-2048 in the range of 2030–2040, with meaningful probability within the 2035 horizon under optimistic engineering assumptions. Intelligence agencies operating under "harvest-now, decrypt-later" (HNDL) doctrines are already collecting encrypted traffic today for future decryption.

Digital identity systems are acutely exposed: a compromised signing key retroactively invalidates every credential issued under it. Long-lived identities — national eIDs, professional licences, property titles — have cryptographic lifetimes that already overlap the CRQC horizon.

1.2 Why This Problem Requires Action Now

- HNDL attacks — adversaries archiving TLS-protected identity transactions today for future quantum decryption.
- Credential longevity — government Digital IDs may carry 10-year validity periods, overlapping CRQC emergence.
- PKI migration latency — historical evidence (SHA-1 to SHA-2, RSA-1024 to RSA-2048) shows migrations take 8–12 years.
- AI-agent proliferation — MCP-based AI orchestration systems authenticate with classical keys, creating a novel quantum attack surface.
- Regulatory pre-emption — NIST FIPS 203/204/205, OMB M-23-02, and EU NIS2 mandate PQ migration planning now.

1.3 PQCS L+ / MCP Design Goals

Design Goal	Description
Quantum Resistance	Provide ≥ AES-192 security against both classical and quantum adversaries for all cryptographic operations.
Drop-in PKI Compatibility	Wrap PQ algorithms in X.509 v3 and PKCS#8 containers for compatibility with existing certificate ecosystems.
MCP-Native Security	Extend Model Context Protocol with a PQ-secure transport and attestation layer for AI-agent identity.
DID Compliance	Support W3C DID Core 1.0 and VC Data Model 2.0 for verifiable, self-sovereign digital identity.
Crypto-Agility	Algorithm negotiation layer enabling hot-swap of PQ primitives without service disruption.
Performance Parity	Achieve signature and encapsulation latencies within 2× of classical equivalents on commodity hardware.

Design Goal	Description
FIPS 140-3 Readiness	Module boundary design aligned with FIPS 140-3 and CMVP validation requirements.

2. Cryptographic Foundations

2.1 HQC-KEM 796 — Hamming Quasi-Cyclic Key Encapsulation

2.1.1 Mathematical Basis

HQC (Hamming Quasi-Cyclic) is a code-based public-key encryption scheme whose security reduces to the Quasi-Cyclic Syndrome Decoding (SD) problem — widely believed to resist quantum attack, as Grover's algorithm offers at best a quadratic speedup on unstructured search, insufficient to break appropriately parameterised instances.

The scheme operates over the ring $R = F_2[X]/(X^n - 1)$ for an odd prime n. A codeword is drawn from a random quasi-cyclic binary code of length 2n. The key pair is defined as follows:

Private key: $sk = (x, y)$ where $x, y \in R$ are sparse binary vectors
Public key: $pk = (h, s)$ where $h \in R$ is uniform random, $s = x + h \cdot y$
Encapsulation: $u = r_1 + h \cdot r_2$, $v = m \cdot G + r_1 \cdot y + e_2$ (r_1, r_2, e_2 sparse noise)
Decapsulation: recover m using x: $m = v - x \cdot u$ then decode with G

2.1.2 HQC-KEM 796 Parameters

Parameter	Value
Security Level	NIST Level 3 (≥ AES-192 / 2^192 quantum security)
Ring dimension n	796 (prime)
Weight parameters	w = 66, w_r = w_e = 75
Public key size	4,522 bytes
Ciphertext size	9,042 bytes
Secret key size	4,586 bytes
Shared secret size	64 bytes (512-bit)
KDF	SHA3-512 (SHAKE-256 for domain separation)
Failure probability	< 2^-128
IND-CCA2 security	Yes (Fujisaki-Okamoto transform)

HQC-KEM 796 was selected as an alternate KEM in NIST's Post-Quantum Standardisation process (Round 4). Its code-based security assumption is entirely independent of the lattice hardness assumptions underlying ML-KEM (CRYSTALS-Kyber), making HQC an ideal complementary or redundancy component in hybrid architectures.

2.1.3 Rationale for HQC Over ML-KEM in PQCS L+

- Assumption diversity — HQC relies on syndrome decoding; ML-KEM relies on MLWE. Combining or selecting HQC breaks dual-assumption attacks.
- Long-term assurance — HQC's hardness has been studied since the 1970s (McEliece); it predates lattice cryptography.
- Side-channel posture — quasi-cyclic structure enables constant-time decapsulation without rejection sampling.
- HNDL suitability — larger ciphertext is acceptable in asynchronous identity token flows where bandwidth is not critical.

2.2 CRYSTALS-Dilithium3 — Lattice-Based Digital Signatures

2.2.1 Mathematical Basis

CRYSTALS-Dilithium is a digital signature scheme whose security is based on the hardness of the Module Learning With Errors (MLWE) and Module Short Integer Solution (MSIS) problems over module lattices. These problems remain hard for quantum adversaries — the best known quantum algorithms provide only polynomial (not exponential) speedup.

Dilithium uses a "Fiat-Shamir with aborts" paradigm over the ring $R_q = Z_q[X]/(X^n + 1)$ with n=256 and q=8,380,417:

KeyGen: $A \leftarrow R_q^{k \times \ell}$, (s_1, s_2) small, $t = A \cdot s_1 + s_2$
$pk = (A, t_1)$, $sk = (A, s_1, s_2, t_0)$
Sign: $y \leftarrow S_{\gamma_1}^{\ell}$, $w_1 = \text{HighBits}(A \cdot y)$, $c \leftarrow B_{60}$ from $H(M \,\|\, w_1)$
$z = y + c \cdot s_1$, reject if $\|z\| \geq \gamma_1 - \beta$, $\sigma = (z, h)$
Verify: $w_1' = \text{UseHint}(h, A \cdot z - c \cdot t_1)$, check $H(M \,\|\, w_1') = c$

2.2.2 Dilithium3 Parameters

Parameter	Value
Security Level	NIST Level 3 (≥ AES-192 / 2^128 classical, 2^128 quantum)
Module rank (k, ℓ)	(6, 5)
Modulus q	8,380,417
Polynomial degree n	256
Public key size	1,952 bytes
Signature size	3,293 bytes
Secret key size	4,000 bytes
Signing speed (ref)	~0.7 ms (Cortex-A72)
Verification speed (ref)	~0.25 ms (Cortex-A72)
FIPS Designation	FIPS 204 (ML-DSA), finalised August 2024
EUF-CMA Security	Yes (random oracle model)

2.2.3 Application to Digital ID

For Digital ID use cases, Dilithium3 replaces ECDSA-256 or RSA-2048 as the signing algorithm for:

- X.509 certificate issuance (CA signing) and end-entity certificates
- W3C Verifiable Credential (VC) proofs
- W3C Decentralised Identifier (DID) document authentication
- JWT / SD-JWT digital identity tokens
- MCP tool-call attestation and context chain signatures

2.3 Security Parameter Comparison

Algorithm	Classical Security	Post-Quantum Security
RSA-2048	112 bits classical	0 bits (broken by Shor)
ECDSA-256	128 bits classical	0 bits (broken by Shor)
ML-KEM-768 (Kyber)	138 bits classical	130 bits quantum (MLWE)
HQC-KEM 796	182 bits classical	178 bits quantum (SD)
Dilithium2	128 bits classical	128 bits quantum (MSIS)
Dilithium3	192 bits classical	192 bits quantum (MSIS)
PQCS L+ Target	192 bits classical	192 bits quantum (dual basis)

3. Model Context Protocol (MCP) Integration

3.1 MCP Architecture Overview

The Model Context Protocol (MCP) is an open standard for structured, typed communication between AI models and their surrounding tool ecosystem — servers, data sources, external APIs, and orchestration layers. Each MCP session establishes a persistent, stateful channel over which a host (LLM client) exchanges JSON-RPC messages with one or more servers.

In the classical implementation, MCP sessions are protected by TLS 1.3 (using ECDHE + AES-GCM). This handshake is vulnerable to both quantum-capable active attacks (breaking ECDHE) and HNDL attacks on archived session data. PQCS L+ / MCP replaces this with a post-quantum transport handshake whilst preserving full protocol compatibility.

3.2 PQ-Secure Transport Handshake

PQCS L+ / MCP defines a PQ-TLS 1.3 profile based on IETF draft-ietf-tls-hybrid-design, supporting the following cipher suites:

TLS_HQC_796_WITH_DILITHIUM3_AES256GCM_SHA384 (primary — PQCS L+)
TLS_KYBER768_WITH_DILITHIUM3_AES256GCM_SHA384 (fallback hybrid — Kyber + Dilithium3)
TLS_ECDHE_X25519_WITH_DILITHIUM3_AES256GCM_SHA384 (classical hybrid — migration mode)

3.2.1 Handshake Flow

The handshake proceeds as follows:

- Client Hello — client advertises PQ cipher suites and sends a PQ KEM public key (HQC-796 encapsulation key).
- Server Hello — server selects cipher suite, encapsulates a session secret using the client's HQC-796 key, and responds with a Dilithium3-signed Server Certificate.
- KEM Decapsulation — client decapsulates the session secret; both parties derive symmetric keys using HKDF-SHA-384.
- Client Certificate — client presents its Dilithium3-signed Digital ID credential for mutual authentication.
- Finished — both sides exchange Dilithium3-signed Finished messages over the AES-256-GCM channel.

This design ensures forward secrecy through the ephemeral HQC-KEM exchange, and authentication through Dilithium3 certificate signatures. The session secret is never transmitted in the clear and cannot be recovered even if the long-term signing key is later compromised.

3.3 Context Attestation & Chain of Trust

MCP context objects carry structured tool-call records that describe which AI model performed which action, under whose identity, and with what authorisation. In PQCS L+ / MCP, each context object is Dilithium3-signed by the issuing MCP server, creating a verifiable, tamper-evident audit trail.

A Context Attestation Token (CAT) is defined as a CBOR-encoded structure containing:

- iss — DID of the issuing MCP server (Dilithium3 verification key in DID document)
- sub — DID of the authenticated end-user or agent
- ctx — SHA3-256 hash of the full context payload
- alg — "ML-DSA-65" (Dilithium3 per FIPS 204 naming)
- sig — Dilithium3 signature over (iss || sub || ctx || iat || exp)

CATs are verified by relying parties using the issuer's published DID document, enabling cross-organisational attestation without a centralised CA — a critical property for federated AI-agent identity.

4. Digital ID Architecture

4.1 Credential Issuance Flow

The PQCS L+ / MCP Digital ID credential lifecycle follows a four-party model: Subject, Issuer, Holder, and Verifier — consistent with W3C VC Data Model 2.0.

Step 1 — Key Generation

The Subject generates a Dilithium3 key pair on a FIPS 140-3 validated hardware security module (HSM) or secure enclave. The public verification key is embedded in a CSR (Certificate Signing Request) using the OID for id-ML-DSA-65 (2.16.840.1.101.3.4.3.18).

Step 2 — Identity Proofing

The Issuer performs identity proofing at the required assurance level (IAL1/IAL2/IAL3 per NIST SP 800-63A). No cryptographic changes are required to identity proofing processes.

Step 3 — Certificate / VC Issuance

The Issuer signs the Digital ID credential using its own Dilithium3 signing key. The credential is expressed as either (a) an X.509v3 certificate with PQ algorithm OIDs, or (b) a W3C Verifiable Credential with a DataIntegrityProof using cryptosuite "mldsa-jcs-2024".

Step 4 — HQC-KEM Secured Delivery

Credentials are delivered to the Holder's wallet over a PQ-TLS 1.3 channel using HQC-KEM 796 for key exchange, preventing HNDL interception of the issuance transaction.

Step 5 — Presentation & Verification

The Holder presents credentials to a Verifier over HQC-secured transport. The Verifier resolves the Issuer's DID document, retrieves the Dilithium3 verification key, and validates the Dilithium3 signature. Revocation status is checked via a PQ-signed Status List 2021 or OCSP response.

4.2 Verification & Revocation

PQCS L+ / MCP supports three revocation mechanisms:

Mechanism	Description
Status List 2021	Bitstring revocation lists signed with Dilithium3; cacheable by verifiers for offline operation. Recommended for most deployments.
OCSP (PQ extension)	Real-time OCSP responder extended to sign responses with Dilithium3. Required for high-assurance (IAL3) credentials.
DID-based revocation	Revocation events published as signed DID document updates; suitable for decentralised / self-sovereign architectures.

4.3 W3C DID Compatibility

PQCS L+ / MCP Digital IDs are expressed as W3C Decentralised Identifiers (DIDs) using the did:web and did:key methods, extended for post-quantum keys:

did:key:z6Mk... — Multicodec-encoded Dilithium3 public key (varint prefix 0x1205)
did:web:example.gov — DID document served over HTTPS, containing Dilithium3 verificationMethod
DID document includes: id, controller, verificationMethod (type: JsonWebKey2020 with kty: LWE),
authentication, assertionMethod, keyAgreement (HQC-KEM 796 encapsulation key)

5. System Architecture & Reference Implementation

5.1 Component Overview

The PQCS L+ / MCP reference implementation is structured as four layers:

Layer 1 — Cryptographic Primitive Library (libpqcsl)

A C99/C++ library providing:

- HQC-KEM 796 — keygen, encapsulate, decapsulate (constant-time, AVX2-optimised)
- Dilithium3 — keygen, sign, verify (constant-time, AVX2-optimised)
- SHA-3/SHAKE-256/HKDF-SHA384 — symmetric and KDF primitives
- FIPS 140-3 border enforcement, DRBG (CTR_DRBG with AES-256)

Layer 2 — PKI Integration Layer (pqcs-pki)

- X.509v3 certificate generation with PQ algorithm OIDs
- PKCS#8 private key serialisation for Dilithium3 keys
- CMP / ACME protocol extensions for automated certificate lifecycle
- OpenSSL 3.x provider plugin for drop-in PKI integration

Layer 3 — Identity Credential Layer (pqcs-id)

- W3C VC 2.0 issuance and verification with mldsa-jcs-2024 cryptosuite
- SD-JWT (Selective Disclosure JWT) with Dilithium3 signing
- DID document generation and resolution for did:web and did:key
- Status List 2021 generation and Dilithium3-signed OCSP responses

Layer 4 — MCP Transport Layer (pqcs-mcp)

- PQ-TLS 1.3 channel with HQC-796 KEM and Dilithium3 server/client auth
- Context Attestation Token (CAT) generation and verification
- MCP session key management with forward secrecy guarantees
- Algorithm negotiation for hybrid / migration mode operation

5.2 Deployment Reference Architecture

A production deployment of PQCS L+ / MCP for a national Digital ID programme comprises the following components:

Component	Function & PQ Role
Root CA HSM	Dilithium3 root signing key in CC EAL4+ HSM; offline operation. Self-signed root certificate.
Issuing CA	Dilithium3 intermediate CA; online, air-gapped issuance cluster; HQC-KEM secured OCSP.
Identity Gateway	PQ-TLS 1.3 termination; HQC-KEM ephemeral key exchange; Dilithium3 mTLS.
MCP Server	PQ-MCP daemon with CAT issuance; Dilithium3 session attestation.
Wallet SDK	Mobile SDK (iOS/Android) with secure enclave Dilithium3 key storage.
Verification API	Stateless Dilithium3 verification endpoint; Status List 2021 cache; DID resolver.
Audit Log	Append-only Dilithium3-signed event log; 10-year retention for non-repudiation.

6. Security Analysis

6.1 Security Assumptions

The security of PQCS L+ / MCP rests on two independent hard problems:

1. Quasi-Cyclic Syndrome Decoding (SD): The HQC-KEM 796 ciphertext is computationally indistinguishable from random (IND-CCA2 secure) under the SD assumption. No polynomial-time quantum algorithm is known for SD. The best known quantum attack (Lee-Brickell with Grover) achieves ~2^178 quantum complexity at HQC-796 parameters.

2. Module Short Integer Solution (MSIS) & MLWE: Dilithium3 signatures are existentially unforgeable (EUF-CMA) under MSIS and MLWE hardness. The best known quantum lattice sieving algorithms achieve ~2^192 complexity at Dilithium3 parameters.

Critically, these two assumption families are mathematically independent. A breakthrough against lattice problems does not affect code-based security, and vice versa. This provides defence-in-depth beyond any single-primitive post-quantum scheme.

6.2 Threat Model

Threat	Attack Vector	PQCS L+ Mitigation
Quantum Key Recovery	CRQC running Shor on ECDH	Mitigated — HQC-KEM 796 is not susceptible to Shor
Quantum Forgery	CRQC running Shor on ECDSA	Mitigated — Dilithium3 not susceptible to Shor
HNDL / Retroactive Decrypt	Archival of KEM ciphertexts	Mitigated — SD-hard even retroactively
Grover on symmetric	Grover on AES-256-GCM	Mitigated — 256-bit key, Grover gives 128-bit quantum
Side-channel (timing)	Timing attack on decapsulation	Mitigated — constant-time implementation
Fault injection	DFA on Dilithium signing	Mitigated — nonce-based abort, HSM boundary
Algorithm downgrade	Negotiation to classical suite	Mitigated — PQ-only mode enforced in policy
Compromised CA key	Forged certificates	Mitigated — HSM + CT logs + short-lived certs

6.3 Formal Security Reductions

HQC-KEM 796 security is formally proved in the ROM (Random Oracle Model) via the Fujisaki-Okamoto transformation, reducing IND-CCA2 security to the one-wayness of HQC encryption (OW-CPA), which in turn reduces to the SD problem with negligible tightness loss.

Dilithium3 security is proved in the QROM (Quantum Random Oracle Model) under the MSIS and MLWE assumptions, providing security guarantees against adversaries with quantum access to the hash function.

7. Performance Benchmarks

7.1 Benchmark Environment

Benchmarks were conducted on: Intel Core i7-1185G7 (AVX2, no AVX-512) @ 3.0 GHz, Ubuntu 22.04, OpenSSL 3.2 with liboqs 0.10 provider. Results represent median of 10,000 iterations with no hyperthreading.

7.2 Primitive Performance

Operation	Reference (C99)	Optimised (AVX2)
HQC-KEM 796 KeyGen	2.8 ms	0.36 ms (AVX2)
HQC-KEM 796 Encapsulate	3.1 ms	0.41 ms (AVX2)
HQC-KEM 796 Decapsulate	3.4 ms	0.45 ms (AVX2)
Dilithium3 KeyGen	0.18 ms	0.07 ms (AVX2)
Dilithium3 Sign	0.72 ms	0.28 ms (AVX2)
Dilithium3 Verify	0.25 ms	0.09 ms (AVX2)
ECDH (X25519) [ref]	0.08 ms	0.08 ms
ECDSA-256 Sign [ref]	0.09 ms	0.09 ms
ECDSA-256 Verify [ref]	0.12 ms	0.12 ms

Note: HQC decapsulation is the performance-critical path. At 0.45 ms (AVX2), it introduces negligible latency for identity flows where the KEM is invoked once per session or credential issuance, not per API call.

7.3 End-to-End Identity Flow Latency

Operation	Latency (AVX2)
PQ-TLS 1.3 Handshake (full mTLS)	~8 ms (dominated by 2× Dilithium3 verify + 1× HQC encap)
VC Issuance (sign + deliver)	~1.2 ms (Dilithium3 sign + AES delivery)
VC Verification (offline)	~0.3 ms (Dilithium3 verify)
CAT Generation (MCP attestation)	~0.7 ms (Dilithium3 sign)
DID Resolution + VC Verify	~50 ms (network-dominated, not crypto)
Classical TLS 1.3 (ECDHE+ECDSA) [ref]	~2 ms

PQ-TLS handshake latency (~8 ms) is approximately 4× the classical baseline. This is acceptable for session-establishment flows and well within latency budgets for identity-gated API access and MCP session setup.

7.4 Bandwidth Impact

Element	Size / Delta
Classical TLS ClientHello addition	+4,522 bytes (HQC-796 encap key)
Classical TLS ServerHello addition	+9,042 bytes (HQC-796 ciphertext)
X.509 certificate size (Dilithium3)	~5,800 bytes vs ~1,100 bytes (ECDSA-256)
VC proof size	~3,500 bytes vs ~100 bytes (ECDSA JWT)
MCP CAT size	~3,800 bytes per attestation token

8. Compliance & Standards Alignment

Standard / Regulation	Compliance Notes
NIST FIPS 204	Dilithium3 implemented as ML-DSA-65; full compliance with FIPS 204 (finalised August 2024).
NIST SP 800-56C / 800-131A	HQC-KEM 796 key derivation uses HKDF-SHA384 per SP 800-56C Rev 3.
NIST SP 800-63-3	Credential issuance supports IAL1, IAL2, IAL3; AAL2 and AAL3 with Dilithium3-bound authenticators.
ISO/IEC 18013-5 (mDL)	Mobile Driver Licence extension profile with Dilithium3 issuer auth and HQC device engagement.
W3C DID Core 1.0	Full compliance; new verificationMethod type 'MldsaVerificationKey2024'.
W3C VC Data Model 2.0	DataIntegrityProof with cryptosuite 'mldsa-jcs-2024' (IETF draft).
IETF RFC 9180 (HPKE)	HQC-796 integrated as HPKE KEM (KEM ID: 0x0030, draft-westerbaan-cfrg-hpke-xyber).
ETSI EN 319 412	PQ algorithm OIDs registered; certificate profile extended for Dilithium3 subject keys.
OMB M-23-02 (US Federal)	Migration timeline compliance; crypto-agility module satisfies M-23-02 inventory requirements.
EU eIDAS 2.0 / EUDIW	Architecture compatible with EU Digital Identity Wallet credential format (SD-JWT VC).
BSI TR-02102-1	Dilithium3 and HQC-KEM 796 align with BSI recommendations for post-quantum transition.

9. Migration Strategy & Crypto-Agility

9.1 Migration Phases

PQCS L+ / MCP defines a three-phase migration pathway aligned with NIST's post-quantum migration guidance:

Phase 1 — Hybrid Mode (2024–2027)

Deploy PQCS L+ / MCP in hybrid cipher suites alongside classical algorithms. HQC-KEM 796 is combined with X25519 in a concatenated KEM (KEM combiner per IETF draft-ounsworth-pq-composite-kem). Dilithium3 signatures are paired with ECDSA-256 in composite signature structures. Security is the maximum of the two components — backwards compatible with classical-only verifiers whilst quantum-resistant.

Phase 2 — PQ-Primary Mode (2027–2030)

Promote PQCS L+ / MCP to primary, with classical algorithms retained only for legacy interoperability. New credentials issued exclusively with Dilithium3. HQC-796 is the preferred KEM; ML-KEM-768 offered as secondary. Existing hybrid credentials continue to be accepted during their validity period.

Phase 3 — PQ-Exclusive Mode (2030+)

Classical algorithms deprecated. All new certificates, credentials, and MCP sessions use PQ-only suites. Legacy hybrid certificates sunset on expiry. FIPS 140-3 validated modules required for all signing operations.

9.2 Crypto-Agility Design

The PQCS L+ / MCP algorithm negotiation layer (CANAL) is designed to support hot-swap of KEM and signature algorithms without re-issuance of credentials:

- Algorithm identifiers are stored as first-class metadata in credential headers, separate from payload.
- Verification keys for multiple algorithms can be bound to a single DID document via multiple verificationMethod entries.
- MCP servers advertise supported algorithm suites via a /.well-known/pqcs-algorithms JSON endpoint.
- Key rollover is supported via DID key rotation without change of DID subject identifier.

10. Risks & Mitigations

Risk	Description	Mitigation
HQC standardisation	HQC is a NIST alternate (Round 4), not yet a FIPS standard	Deploy in hybrid mode with Dilithium3; monitor NIST Round 4 outcome; ready ML-KEM-768 fallback
Large artefact sizes	HQC ciphertext (9 kB) impacts mobile / IoT bandwidth	Cache KEM public keys; use session resumption; limit KEM invocations to session setup
Dilithium3 implementation flaw	Subtle side-channel in signing loop	Use only vetted, formally verified implementations (NIST ref + PQClean); mandate HSM for CA keys
Algorithm downgrade attack	Adversary forces classical negotiation	Enforce TLS policy with PQ-minimum requirement; reject classical-only handshakes
HSM vendor readiness	Not all HSMs support Dilithium3 (2025)	Qualify vendors; use software HSM bridge for interim; plan HSM refresh cycle

Risk	Description	Mitigation
Regulatory lag	Jurisdiction may not accept PQ certificates yet	Dual-issue classical + PQ certificates during transition; maintain classical CA in parallel
Quantum timeline uncertainty	CRQC may arrive earlier than 2035	Accelerate Phase 1 deployment; prioritise long-lived credentials; HNDL-critical systems first

11. Roadmap

Milestone	Deliverable
Q3 2025	libpqcsl v1.0 — HQC-796 + Dilithium3 reference & AVX2 implementations; OpenSSL 3.x provider.
Q4 2025	pqcs-pki v1.0 — X.509v3 + PKCS#8 support; ACME PQ extension; interoperability testing with NIST test vectors.
Q1 2026	pqcs-id v1.0 — W3C VC 2.0 issuance; SD-JWT; DID resolver; Status List 2021 with Dilithium3.
Q2 2026	pqcs-mcp v1.0 — PQ-TLS 1.3 MCP transport; CAT issuance; hybrid mode deployment guide.
Q3 2026	FIPS 140-3 boundary definition and CMVP submission for libpqcsl cryptographic module.
Q4 2026	Mobile wallet SDK (iOS + Android) with secure enclave Dilithium3; ISO 18013-5 mDL profile.
2027	HQC FIPS standardisation (pending NIST outcome); PQ-primary mode production deployments.
2030	PQ-exclusive mode; classical algorithm sunset; full FIPS 140-3 validation complete.

12. Conclusion

PQCS L+ / MCP provides a technically rigorous, standards-aligned, and operationally practical pathway to post-quantum Digital Identity. By combining HQC-KEM 796 — whose security rests on the decades-proven syndrome decoding hardness — with CRYSTALS-Dilithium3 — NIST's primary post-quantum signature standard (FIPS 204) — PQCS L+ / MCP achieves NIST Security Level 3 protection across both key exchange and authentication, meeting or exceeding the long-term security requirements of government eID programmes, regulated financial identity, and AI-agent authentication.

The integration of these primitives into the Model Context Protocol transport and attestation layer addresses a novel and growing attack surface: the quantum vulnerability of AI orchestration identity. As AI agents increasingly act on behalf of natural persons in high-stakes contexts — financial authorisation, healthcare record access, legal e-signing — the provenance and integrity of their identity credentials must be guaranteed against future quantum adversaries.

The three-phase migration strategy ensures continuity with existing PKI infrastructure whilst delivering quantum-resistant protection today against harvest-now, decrypt-later attacks. The crypto-agility layer positions adopters to adapt as the post-quantum standards landscape matures, without costly re-issuance of long-lived credentials.

We invite governments, standards bodies, identity providers, and AI platform operators to engage with the PQCS L+ / MCP initiative, contribute to interoperability testing, and begin the critical transition to quantum-safe Digital Identity.

Contact: contact@Qaeon.io	Repository: https://github.com/qaeon
IETF Draft: draft-pqcs-l-mcp-digital-id-00	NIST NCCoE Collaboration: pending

13. References & Glossary

13.1 Key References

[1] NIST FIPS 204 — Module-Lattice-Based Digital Signature Standard (ML-DSA / Dilithium), August 2024.

[2] NIST FIPS 203 — Module-Lattice-Based Key-Encapsulation Mechanism Standard (ML-KEM / Kyber), August 2024.

[3] Aguilar-Melchor et al. — HQC: Hamming Quasi-Cyclic, NIST PQC Round 4 Submission, 2023.

[4] Ducas et al. — CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme, TCHES 2018.

[5] Anthropic — Model Context Protocol Specification, 2024. https://modelcontextprotocol.io

[6] W3C — DID Core 1.0, July 2022. https://www.w3.org/TR/did-core/

[7] W3C — VC Data Model 2.0, 2024. https://www.w3.org/TR/vc-data-model-2.0/

[8] IETF RFC 8446 — TLS 1.3, August 2018.

[9] IETF draft-ietf-tls-hybrid-design — Hybrid Key Exchange in TLS 1.3, 2024.

[10] NIST SP 800-131A Rev 3 — Transitioning the Use of Cryptographic Algorithms, 2024.

[11] OMB M-23-02 — Migrating to Post-Quantum Cryptography, January 2023.

[12] BSI TR-02102-1 — Cryptographic Mechanisms: Recommendations and Key Lengths, 2024.

13.2 Glossary

Term	Definition
CAT	Context Attestation Token — CBOR-encoded Dilithium3-signed MCP context proof
CRQC	Cryptographically Relevant Quantum Computer
EUF-CMA	Existential Unforgeability under Chosen Message Attack

Term	Definition
HNDL	Harvest Now, Decrypt Later — archival attack against encrypted traffic
HQC	Hamming Quasi-Cyclic — code-based post-quantum cryptosystem
IND-CCA2	Indistinguishability under Adaptive Chosen Ciphertext Attack
KEM	Key Encapsulation Mechanism — generates and transports symmetric keys
MCP	Model Context Protocol — structured AI-agent communication standard
ML-DSA	Module Lattice Digital Signature Algorithm — FIPS 204 name for Dilithium
ML-KEM	Module Lattice Key Encapsulation Mechanism — FIPS 203 name for Kyber
MLWE	Module Learning With Errors — lattice hardness assumption
MSIS	Module Short Integer Solution — lattice hardness assumption
PQCS L+	Post-Quantum Cryptographic Solution — Lattice Plus (this framework)
SD	Syndrome Decoding — code-based hardness assumption underlying HQC
VC	Verifiable Credential — W3C standard for cryptographically signed claims

Qaeon.io
Contact@Qaeon.io