WP-033
June 2026
12 sections

RESEARCH SURVEY & ARCHITECTURE

The Convergence of Post-Quantum Cryptography and Zero-Knowledge Proofs: Closing the Verification Gap

PQC secures the channel. ZK proofs protect the claim. Neither alone closes the verification gap that exposes identity infrastructure to quantum-era adversaries. This paper maps the research landscape and the architecture that bridges them.

AffixIO Research | June 2026 | Download PDF

ABSTRACT

The NIST post-quantum cryptography standards (FIPS 203 ML-KEM, FIPS 204 ML-DSA, FIPS 205 SLH-DSA) have entered enterprise deployment. The zero-knowledge proof research community has produced practical SNARK and STARK systems at production scale. These two bodies of work have largely evolved in parallel, addressing different threat surfaces with different tools. This paper argues that closing the verification gap between them is now both technically feasible and practically urgent. We survey the current state of lattice-based ZK proofs (PLAZA, LaBRADOR, rejection-free MLWE

constructions), document the engineering challenges of embedding NIST PQC operations in arithmetic circuits (the prime mismatch, NTT constraint counts, Keccak cost in R1CS), present the hybrid SNARK+ML-DSA-65 construction with its formal security reduction, introduce the concept of stateless post-quantum verification as an architectural alternative to certificate-database-backed PKI, describe the sublinear Merkle attestation scheme for long-lived compliance records, and enumerate the open research problems that stand between the current state and a fully post-quantum ZK proof system. We situate recent developments, including Google's April 2026 Groth16 proof of quantum cryptanalytic knowledge and its independent verification by Trail of Bits, within the broader convergence narrative. AffixIO's production work in this area is referenced where relevant; specific circuit implementations are omitted from public documentation.

CONTENTS

SECTION 01

Two Deployments, One Gap

Post-quantum cryptography and zero-knowledge proofs entered production at roughly the same moment, solving roughly adjacent problems, and have not yet been systematically combined. Understanding why this matters, and what it costs in practice, requires being precise about what each deployment actually does and does not do.

PQC deployment at the protocol layer is, in structural terms, a substitution. The NIST FIPS 203 (ML-KEM), FIPS 204 (ML-DSA), and FIPS 205 (SLH-DSA) standards define quantum-resistant replacements for ECDH, ECDSA, and RSA respectively. Deploying them in TLS 1.3, as X25519MLKEM768 for key exchange and ML-DSA-65 for certificate signing, means that the transport layer is quantum-resistant: a future adversary with a cryptographically relevant quantum computer cannot break the session key or forge the certificate. The surrounding architecture, however, remains unchanged: PKI hierarchies, OCSP responders, certificate transparency logs, and revocation databases. These are stateful, queryable systems. Their existence creates a surface for a class of attack that PQC does not address.

ZK proof deployment is, in structural terms, an addition. A Groth16 SNARK (~192 bytes), a Plonk proof (~1 KB), or a STARK (~10-200 KB) is inserted into a workflow to allow a verifier to confirm a computational claim without receiving the private inputs that witness the claim. The verifier runs a short deterministic verification algorithm against a fixed verification key and receives a binary result. No state is created at a remote server; no query is sent to a revocation registry; the proof is self-contained. This architecture is stateless.

The gap between these two deployments is the verification gap: the question of who can verify what, about whom, without creating harvestable evidence. Standard PQC deployment answers "the quantum threat to the channel" but does not answer "the quantum threat to the identity layer". ZK proof deployment answers "the disclosure threat to private data" but, until NIST PQC operations are expressed inside ZK circuits, does not provide post-

quantum binding for the proofs it produces. The combination of the two is what provides both properties simultaneously: quantum-resistant binding with no harvestable state.

This paper maps where the research record stands on that combination, what the engineering obstacles are, what has been built, and what remains open. It is a research survey and architecture paper, not a product specification. References to AffixIO's production work appear where relevant; implementation specifics are not disclosed.

SECTION 02

The Harvest Problem at the Identity Layer

The harvest-now-decrypt-later (HNDL) threat is well-documented for transport-layer data. Nation-state adversaries are understood to be capturing encrypted traffic today for retrospective decryption once a cryptographically relevant quantum computer becomes available. Google's March 2026 research updated estimates for the qubit requirements to break 256-bit elliptic curve cryptography, suggesting that the threat timeline may be shorter than previously assumed. The response, accelerating migration to ML-KEM hybrid key exchange in TLS 1.3, is appropriate and underway.

What is less frequently discussed is the harvest threat at the identity layer. Identity infrastructure generates its own class of long-lived sensitive data:

- **OCSP query logs** record which certificates were checked and when. Over time, this creates a record of which parties communicated, even after the session keys are forgotten. If the query logs are captured, the metadata remains useful to a quantum adversary even before they can break session keys.
- **Certificate transparency logs** are public by design, recording every issued certificate with its subject information. Structural analysis of CT logs can reveal organisational relationships, acquisition plans, and operational patterns that the parties did not intend to publish.
- **Credential databases** in enterprise SSO systems, federated identity providers, and authentication services store user attributes, group

memberships, and access patterns. These are signed with ECDSA today; the signatures will be forgeable by a quantum adversary. A captured copy of such a database becomes fully exploitable once a quantum computer is available, enabling retroactive impersonation of any identity.

- **Audit logs** generated by AI governance, financial compliance, and healthcare systems are retained for 5 to 10 years under regulatory mandate. Signed today with ECDSA or RSA, they are vulnerable to retroactive forgery. An adversary who can forge audit log entries can retrospectively alter the compliance record of any regulated entity.

Zero-knowledge proofs address the harvest threat at the identity layer in a way that PQC alone does not, because they remove the need for the harvestable data to exist in the first place. A ZK proof of eligibility does not require a credential database: the credential holder presents a proof; the verifier checks it; no record of which credential was checked is created at the verifier's infrastructure. If the proof itself carries post-quantum binding (ML-DSA-65 signing of the proof transcript), then even a quantum adversary who obtains the proof cannot retroactively forge it, because they would need to break ML-DSA-65 to produce a fraudulent proof transcript with a valid signature.

SECTION 03

The Research Record: Where PQC and ZK Have Met

The intersection of lattice-based cryptography and zero-knowledge proofs has a substantial theoretical literature, but most of it predates the NIST PQC standardisation process and does not directly address the question of ZK proofs about NIST-standardised operations. The following survey covers the threads that are most directly relevant.

Lattice-Based ZK Proofs: PLAZA and LaBRADOR

The European Research Council's PLAZA project (2021-2026) represents the most sustained funded research programme specifically on practical lattice-based zero-knowledge proofs. PLAZA's output includes improved protocols for proving knowledge of short vectors satisfying linear relations over module lattices, based on the Module-SIS and Module-LWE hardness assumptions. These are the same assumptions underlying ML-KEM and ML-DSA, which creates a structural affinity: a PLAZA-style ZK proof system and an ML-KEM circuit are both grounded in lattice hardness, avoiding the prime mismatch problem that arises when ML-KEM is embedded in a pairing-based SNARK field.

The LaBRADOR construction (Bootle et al., 2023) is the current state of the art for recursive lattice-based SNARKs. LaBRADOR achieves proof sizes in the range of tens of kilobytes for circuits of practical scale, compared with 192 bytes for Groth16. The verification time is higher than Groth16 but practical for non-interactive compliance applications. The key property that LaBRADOR provides, which Groth16 does not, is soundness based on lattice hardness assumptions: a quantum adversary who can break the BN254 discrete logarithm (and thus forge Groth16 proofs) cannot forge LaBRADOR proofs.

Rejection-Free MLWE Zero-Knowledge Proofs

The 2025 IACR ePrint paper "Rejection-Free Framework of Zero-Knowledge Proof Based on Hint-MLWE" (2025/2239) addresses a specific efficiency bottleneck in lattice-based ZK proofs: the rejection sampling step. Classical lattice ZK proofs based on Fiat-Shamir require the prover to restart a non-trivial fraction of proof attempts because the generated response vector does not have the correct distribution. Rejection-free constructions eliminate this overhead, improving practical proving times significantly. The Hint-MLWE assumption used in the construction is a structured variant of MLWE that permits rejection-free proof generation while maintaining post-quantum security under standard lattice assumptions.

Post-Quantum Audit Evidence: Formal Security Models

Kao's December 2025 preprint "Post-Quantum-Resilient Audit Evidence for Long-Lived Regulated Systems" (arXiv 2512.00110) introduces three formal security notions specifically for audit evidence structures: Q-Audit Integrity (a

quantum adversary cannot produce a fraudulent audit entry that passes verification), Q-Non-Equivocation (a signing party cannot produce two different audit records for the same event that both pass verification), and Q-Binding (the audit root commits unambiguously to its set of records). The paper analyses hash-and-sign instantiations in the quantum random-oracle model and evaluates Merkle-root anchoring as one of three migration strategies. This formalises the security argument for the Merkle-anchored ML-DSA-65 construction described in WP-032 from AffixIO's whitepaper series.

Zero-Knowledge Software Auditing for AI

A parallel thread in the 2025-2026 literature applies ZK proofs to AI system compliance without focusing specifically on PQC. The paper "Show Me You Comply... Without Showing Me Anything" (arXiv 2510.26576) proposes a ZKMLOps framework for generating cryptographic evidence of AI system compliance without exposing model weights or training data. The paper demonstrates stable orchestration overhead across ZK backends and establishes the "audit-on-demand" scenario as the primary use case, noting that full ZK auditing provides confidentiality and integrity guarantees that lightweight alternatives cannot match. The gap in this work is that it uses classical ZK proofs without PQC binding; the signed audit records are quantum-vulnerable.

Google's April 2026 Groth16 Proof

In April 2026, Google Quantum AI published a Groth16 SNARK proof demonstrating knowledge of a quantum cryptanalytic attack on elliptic curve encryption, built using SP1 zkVM and Groth16, without revealing the attack details. This represented the first publicly documented use of a production SNARK for responsible disclosure of a quantum threat. Trail of Bits independently verified the proof and published their analysis the same month, demonstrating that the ZK verification framework was correct and that the approach generalises. The episode is notable for this survey because it validates the practical use of Groth16 SNARKs in a quantum-relevant context, though notably the proof itself does not carry ML-DSA-65 post-quantum binding: the proof is about a quantum attack but is not itself quantum-resistant.

Zero-Knowledge Federated Learning with Lattice-Based Hybrid Encryption

A March 2026 arXiv paper (2603.03398) combines lattice-based encryption with ZK proofs for federated learning in the healthcare context. The construction uses ML-KEM for encryption and ZK proofs for gradient verification, operating in adjacent spaces to the circuit-level combination this paper addresses. It demonstrates demand for PQC+ZK combinations in regulated-data AI contexts, and illustrates that the research community is approaching the combination from multiple directions simultaneously.

SECTION 04

The Prime Mismatch: ML-KEM in R1CS

The central engineering obstacle to expressing ML-KEM operations in Groth16 ZK circuits is the prime mismatch between ML-KEM's working modulus and the SNARK field characteristic. This section restates the problem precisely, because it is the root cause of the large constraint counts that make ML-KEM circuits more expensive than classical alternatives.

ML-KEM-768 (FIPS 203) operates over the polynomial ring $Z_q[x]/(x^{256} + 1)$ where $q = 3,329$. This prime was chosen because it admits an efficient Number Theoretic Transform: $3,329 = 13 \times 256 + 1$, so a primitive 512th root of unity exists in Z_q, enabling the NTT-based fast polynomial multiplication at the heart of ML-KEM's efficiency.

Groth16 over BN254 performs arithmetic over a field F_p where:

```
p = 21888242871839275222246405745257275088548364400416034343698204186575808495617
```

This is a 254-bit prime. Addition and multiplication in F_p automatically reduce modulo p. They do not automatically reduce modulo 3,329. A circuit that simply adds two ML-KEM coefficients $a, b \in Z_q$ obtains the correct sum $a + b$ in F_p (since $a + b < 2q < p$), but without further constraints, the circuit does not enforce that the result lies in $[0, q-1]$. The circuit must explicitly compute:

- The quotient d = floor((a + b) / q), which is either 0 or 1 for a single addition.
- The remainder r = (a + b) - d × q.
- Range constraints asserting r ∈ [0, q-1] (requiring approximately 12 constraints for binary decomposition of a 12-bit range).
- A multiplication constraint for d × q.

Each modular reduction therefore costs approximately 15 to 25 R1CS constraints. Applied to the 7 layers of 128 butterfly operations in an ML-KEM NTT transform, each butterfly requiring one modular multiplication and two modular additions/subtractions, the total constraint count for one NTT forward transform reaches approximately 22,400 constraints. The full encapsulation operation, including matrix-vector multiplication and SHAKE-128 pseudorandom generation (which requires Keccak-f[1600] permutation expansion, costing approximately 150,000 to 250,000 constraints per invocation), produces circuits of 1.5 to 3 million constraints.

This is not an implementation failure; it is a structural consequence of the mismatch between q = 3,329 and the BN254 field prime. Three strategies exist to mitigate it:

- **Lazy reduction**: defer modular reductions until the accumulated value risks field overflow. Since p / q ≈ 2^242, approximately 2^242 additions can be accumulated before overflow. This eliminates per-addition range checks but not per-multiplication checks.
- **Plonkish lookup tables**: Plonk-based constraint systems with lookup arguments can precompute a table of (a, b, a + b mod q) for all a, b ∈ Z_q (approximately 11 million entries), reducing modular arithmetic to a table lookup. This improves constraint efficiency by 3-5x but requires a larger trusted setup and is less portable.
- **Lattice-native proof systems**: LaBRADOR and PLAZA-style lattice-based proof systems operate over the same ring structures as ML-KEM, avoiding the prime mismatch entirely at the cost of larger proof sizes.

The choice between these strategies depends on the application's constraints: proof portability (Groth16 is universally verifiable), proof size (Groth16's 192 bytes is unmatched), proving time, and the requirement for fully

post-quantum soundness. For compliance applications with moderate throughput requirements, Groth16 with lazy reduction is the current best practice; for high-throughput or fully post-quantum soundness requirements, lattice-native proof systems are the research frontier.

SECTION 05

Circuit Architecture for NIST PQC Operations

With the prime mismatch understood, the circuit architecture for each NIST PQC operation follows from systematic constraint analysis. The following summarises the key circuits and their properties.

ML-KEM-768 Key Validity Circuit

The key validity circuit proves knowledge of ML-KEM private key (s, e) satisfying the public key relation t = A·s + e in R_q, without revealing (s, e). This requires one matrix-vector NTT multiplication (approximately 500,000 constraints) and coefficient range checks (approximately 6,000 constraints for the small-norm assertion on s and e). Total: approximately 600,000 to 800,000 constraints. Proving time at 2025-generation hardware: approximately 0.3 to 0.5 seconds. Proof size: 192 bytes (Groth16). This is the post-quantum analogue of a Schnorr proof of discrete logarithm knowledge and is the foundational circuit for post-quantum anonymous credential systems.

ML-KEM-768 Encapsulation Proof

The encapsulation proof demonstrates that ciphertext ct was correctly produced from public key pk using randomness r, yielding shared key commitment K. The circuit includes the full ML-KEM encapsulation algorithm: matrix sampling from SHAKE-128 (approximately 500,000 Keccak constraints), matrix-vector multiplication (500,000 constraints), compression (approximately 100,000 constraints), and key derivation hashing (approximately 200,000 constraints). Total: 1.5 to 3 million constraints. Proving time: 0.7 to 2 seconds. This circuit enables post-quantum key escrow proofs and verifiable key establishment records.

ML-DSA-65 Verification Circuit

The ML-DSA verification circuit takes as public inputs the verification key (matrix seed and commitment vector t) and as private witness the message M and signature sig. The circuit verifies all ML-DSA-65 verification checks: matrix-vector multiplication (500,000 constraints), norm check on z (approximately 30,000 constraints), challenge hash reconstruction (approximately 400,000 Keccak constraints), and hint processing (approximately 50,000 constraints). Total: 1 to 2 million constraints. Proving time: 0.5 to 1.5 seconds. This circuit enables proofs that a valid ML-DSA-65 signature exists over a message satisfying some predicate, without revealing the message.

CIRCUIT	APPROX. R1CS CONSTRAINTS	PROVING TIME	PROOF SIZE (GROTH16)	KEY CAPABILITY
ML-KEM-768 key validity	600K–800K	0.3–0.5s	192 bytes	Post-quantum Schnorr analogue
ML-KEM-768 encapsulation proof	1.5M–3M	0.7–2s	192 bytes	Verifiable key establishment
ML-DSA-65 verification circuit	1M–2M	0.5–1.5s	192 bytes	ZK signed-content predicates
Merkle inclusion (SHA-256, depth 20)	~520K	0.2–0.4s	192 bytes	Succinct batch inclusion proof
Ed25519 verify (reference)	3K–5K	<0.01s	192 bytes	Classical analogue

The constraint overhead relative to classical alternatives (100x to 600x) is a direct consequence of the prime mismatch and the cost of Keccak in field-arithmetic constraint systems. It represents an inherent cost of combining NIST PQC standards with pairing-based ZK proofs, not an implementation artefact. Lattice-native proof systems would reduce or eliminate this overhead at the cost of larger proof sizes.

SECTION 06

The Hybrid SNARK+PQC Construction

The hybrid SNARK+PQC construction combines a classical ZK proof system (Groth16) with a post-quantum signature (ML-DSA-65) to produce a proof transcript with two distinct security properties that neither component provides alone.

> **HYBRID SNARK+PQC TRANSCRIPT**
>
> Let C be a ZK circuit with statement x and witness w. Let vk be the Groth16 verification key and sk_PQ be an ML-DSA-65 signing key. The hybrid transcript T(x, w) is defined as: (1) $\pi \leftarrow$ Groth16.Prove(pk, x, w); (2) h $\leftarrow$ SHA-256(π || x); (3) $\sigma \leftarrow$ ML-DSA.Sign(sk_PQ, h); (4) T = (π, x, σ, pk_PQ). Verification requires both Groth16.Verify(vk, x, π) = 1 and ML-DSA.Verify(pk_PQ, h, σ) = 1.

The security properties of this construction are as follows.

Zero-knowledge (from Groth16): The transcript reveals nothing about the witness w beyond what the statement x and the truth of the circuit imply. This follows from the zero-knowledge property of Groth16 under the BN254 group assumptions.

Classical soundness (from Groth16): A computationally bounded classical adversary without the Groth16 proving key cannot produce an accepted proof for a false statement x, under the knowledge-of-exponent assumption in BN254.

Post-quantum binding (from ML-DSA-65): A quantum adversary who can forge Groth16 proofs by breaking BN254 discrete logarithm still cannot produce a hybrid transcript that passes verification without either: (a) obtaining the ML-DSA-65 signing key, or (b) forging an ML-DSA-65 signature on the hash of the forged proof. Under the MLWE assumption, (b) is computationally infeasible for both classical and quantum adversaries.

The independence of the two security assumptions is the key property. Breaking ML-DSA-65 (by solving MLWE) does not help forge Groth16 proofs, and breaking Groth16 (by solving BN254 discrete logarithm with Shor's algorithm) does not help forge ML-DSA-65 signatures. The construction therefore has no single point of cryptographic failure against a quantum adversary: simultaneous breaks of both assumptions would be required, which is not achievable with any known technique.

Comparison with Google's April 2026 ZK Proof

Google's April 2026 Groth16 proof of quantum cryptanalytic knowledge (built using SP1 zkVM) demonstrated that classical ZK proofs can prove facts about quantum algorithms without revealing the algorithm. It is notable that Google's proof did not use the hybrid SNARK+PQC construction: the Groth16 proof was published without ML-DSA-65 post-quantum binding. A quantum adversary who can break BN254 discrete logarithm could, in principle, produce a fraudulent version of Google's proof claiming to prove knowledge of an attack that does not exist. The hybrid construction would close this gap: an ML-DSA-65 signature on the proof hash would ensure that even a quantum adversary who breaks the Groth16 component cannot forge a signed proof transcript. Trail of Bits' independent verification of Google's proof (April 2026) validated the proof's correctness under the classical Groth16 soundness assumption; it does not validate quantum resistance of the proof transcript itself.

SECTION 07

Stateless Post-Quantum Verification

The architectural implication of combining ZK proofs with NIST PQC operations is a qualitatively different verification system. We formalise the distinction that motivates it.

> STATEFUL VERIFICATION SYSTEM
>
> A verification system V is stateful if there exists a verification event e such that V's acceptance decision for e depends on data beyond the proof

> object and the public inputs: specifically, on a remote query result, a database lookup, or a cached state that changes over time.

> **STATELESS VERIFICATION SYSTEM**
>
> A verification system V is stateless if, for all verification events e, V's acceptance decision depends only on: (1) the proof object π; (2) the public inputs x; and (3) a fixed verification key vk. No external query is performed; no mutable state is consulted.

Standard PKI verification is stateful by this definition: the acceptance decision for a certificate depends on an OCSP response (a remote query result that changes as certificates are revoked) or a CRL lookup (a cached state that changes over time). Even hybrid PQC certificate verification, using ML-DSA-65-signed certificates, remains stateful: the quantum-resistant signature on the certificate does not remove the need to check the revocation database.

Groth16 verification is stateless by this definition: the verification algorithm is a fixed deterministic function of (π, x, vk). Adding ML-DSA-65 signing of the proof transcript preserves statelessness: the verifier checks the Groth16 proof and the ML-DSA-65 signature, both against fixed keys, without any remote query.

The operational consequence of statelessness is the removal of the query metadata surface. There is no OCSP log entry recording that entity A checked entity B's certificate at time T. There is no CRL server that records which certificates were queried. The verifier learns only the binary verification result. Under the harvest-now-decrypt-later threat model, this means there is no identity metadata to harvest: the verification event leaves no trace beyond what the proof itself reveals (which, by the zero-knowledge property, is only the truth of the statement).

Progressive Verification Without Cross-Query Leakage

A specific privacy vulnerability of stateful verification is cross-query leakage: the fact that entity A queried the revocation status of entity B at time T, and then queried entity C at time T+5 minutes, creates a graph of interactions

that may be sensitive even if each individual certificate is valid. Stateless verification removes this linkage: each proof verification is independent, and the verifier accumulates no state from which to infer interaction patterns.

This property is particularly important for healthcare and legal contexts, where the parties' consultation of each other's credentials may itself be sensitive. A hospital's verification of a specialist's credentials at a particular time implies that the specialist was consulted at that time; under a stateless ZK credential system, the verification leaves no such trace.

SECTION 08

Sublinear Post-Quantum Attestation

The harvest threat applies not only to identity credentials but to the audit logs that regulated organisations must retain for 5 to 10 years. A naive approach to post-quantum audit attestation (sign each record with ML-DSA-65) creates a storage and verification problem whose severity is proportional to the retention period and record volume.

The sublinear solution is a Merkle hash tree whose root is signed once with ML-DSA-65. The mathematical result is immediate from the structure of Merkle trees and does not require novel cryptography, but it has not been widely applied to the NIST PQC attestation context. The scaling numbers are significant:

SCALE	PER-RECORD ML-DSA-65 SIGNATURES	MERKLE-ML-DSA-65 (SUBLINEAR)	STORAGE REDUCTION
10K records/day	31.4 MB/day	3,293 bytes/batch	>99.9%
1M records/day	3.14 GB/day	3,293 bytes/batch	>99.9%
1M records, 10 years	11.4 TB	~12 MB (roots) + records	>99.9%
Verify one record (1M batch)	1 ML-DSA verify	20 SHA-256 + 1 ML-DSA verify	Equivalent latency
Verify all records (1M batch)	~16,667 seconds	~5 seconds	>99.9% time

The formal security of this construction reduces to two independent post-quantum assumptions: SHA-256 collision resistance (for which Grover-based quantum attacks achieve at most quadratic speedup, leaving approximately 85 bits of post-quantum collision resistance) and ML-DSA-65 unforgeability under MLWE. Kao's 2512.00110 formalises this using the Q-Audit Integrity, Q-Non-Equivocation, and Q-Binding notions in the quantum random-oracle model, providing the formal foundation that regulatory frameworks will require for compliance evidence structures.

The open extension, noted in WP-031 from AffixIO's whitepaper series and relevant here as a research problem, is the succinct variant: a Groth16 or STARK proof of Merkle inclusion compressed to constant size, with ML-DSA-65 post-quantum binding over the proof transcript. This would reduce inclusion proof size from O(log n) hash values (~640 bytes at n = 1 million) to 192 bytes (Groth16) plus 3,293 bytes (ML-DSA-65 signature), regardless of tree depth. The circuit for Merkle inclusion at depth 20 is approximately 520,000 constraints, well within current proving capacity.

SECTION 09

Comparison: Standard PQC vs ZK+PQC Architecture

The following table contrasts the architectural properties of standard PQC deployment (hybrid TLS, ML-DSA certificates) with the ZK+PQC architecture described in this paper.

PROPERTY	STANDARD PQC DEPLOYMENT (EVERYONE'S APPROACH)	ZK+PQC STATELESS ARCHITECTURE
Transport-layer quantum resistance	Yes (ML-KEM hybrid key exchange)	Yes (same, where applicable)
Certificate quantum resistance	Yes (ML-DSA-65 signed certificates)	Replaced by ZK proofs; no certificates required
Revocation infrastructure	Required (OCSP, CRL); stateful	Not required; nullifier-based stateless check
Identity metadata harvestability	High (OCSP logs, CT logs, credential databases)	None (no remote query; no state created)
Audit record quantum resistance	Linear storage (3,293 bytes × n records)	Sublinear (1 root + O(log n) proofs)
Proof of specific operation (e.g., ML-KEM encapsulation)	Not possible; operation is opaque to verifier	Yes (ML-KEM encapsulation ZK circuit)
Privacy of verified claim	None; verifier learns the credential content	Full ZK; verifier learns only binary result
Infrastructure migration required	Yes; 5-10 year PKI migration	No; proof-based, overlay deployment
ZK proof quantum resistance	N/A	Hybrid: ZK privacy + ML-DSA-65 PQ binding
Fully post-quantum ZK soundness	N/A	Open problem (LaBRADOR, PLAZA direction)

The standard PQC deployment solves the transport-layer HNDL problem. The ZK+PQC architecture solves the identity-layer and audit-layer HNDL problems while additionally providing zero-knowledge privacy for the verified claims. The two architectures are complementary rather than competing: organisations deploying ML-KEM hybrid TLS for transport security can simultaneously deploy ZK+PQC credential systems for identity, without any conflict between the two.

SECTION 10

Regulatory Landscape and Compliance Drivers

The convergence of PQC and ZK proofs is driven partly by theoretical interest and partly by a regulatory environment that has become specific about quantum-resistant audit evidence requirements in 2025 and 2026.

NIST SP 800-208 and the 2035 Deprecation Deadline

NIST has established 2035 as the endpoint for deprecating quantum-vulnerable algorithms (RSA, ECDSA, ECDH) in US government and regulated systems. The 2026 compliance clock has begun: US federal agencies are required to complete cryptographic inventories and begin migration planning. The enterprise implications are broad, with lattice-based cryptography (the foundation of ML-KEM and ML-DSA) commanding approximately 50% of the post-quantum security market by 2026. This creates immediate demand for quantum-resistant audit evidence structures, not merely quantum-resistant transport.

EU AI Act Article 12: Quantum-Resistant Audit Logs

The EU AI Act's Article 12 requires providers of high-risk AI systems to maintain logs for 10 years. These logs must be sufficient to reconstruct the AI system's reasoning. A log signed today with ECDSA will be forgeable by a quantum adversary within the 10-year retention window if Google's 2026 qubit estimates prove accurate. The intersection of Article 12's retention requirements with the quantum timeline makes post-quantum audit evidence a regulatory compliance question rather than merely a security best practice.

The ZK+PQC architecture provides Article 12 compliance with an additional privacy property: the audit log can be verified without disclosing the underlying AI inputs, satisfying both the transparency requirements (a verifier can confirm the log is authentic) and data minimisation requirements (the verifier need not access the personal data that informed the AI decision).

DORA, HIPAA, and Financial Services

DORA (EU 2022/2554) requires ICT incident records for 5 years. HIPAA requires security documentation for 6 years. MiFID II requires transaction records for 5 to 7 years. All of these retention periods overlap with the plausible arrival of cryptographically relevant quantum computers under current hardware roadmaps. The 2026 CIAM (Customer Identity and Access Management) evaluation standards have begun explicitly incorporating quantum-resistant encryption requirements, reflecting the recognition that identity infrastructure is as vulnerable as transport infrastructure. Sublinear Merkle-anchored ML-DSA-65 attestation provides the scalable solution for long-lived compliance records under these frameworks.

CNSA 2.0 and the US Defence Context

The US National Security Agency's CNSA 2.0 (Commercial National Security Algorithm Suite 2.0) specifies ML-KEM and ML-DSA as mandatory algorithms for all classified and sensitive systems, with transition deadlines beginning in 2025. Defence contractors and critical infrastructure operators subject to CNSA 2.0 need not only to deploy ML-KEM and ML-DSA for transport and signing, but to prove compliance with the deployment requirements. ZK proofs of ML-KEM key validity and ML-DSA signing operations provide exactly this: a verifiable record of PQC compliance without disclosing the key material or signed content.

SECTION 11

Open Problems and Research Agenda

The following open problems are ranked roughly by their impact on the practical deployment of ZK+PQC systems. Each represents a tractable research direction with a clear problem formulation and significant practical consequence if solved.

1. Fully Post-Quantum ZK Proof Systems for ML-KEM Circuits

Groth16 soundness relies on the knowledge-of-exponent assumption in the BN254 pairing group, which is broken by Shor's algorithm. The ML-DSA-65 signing of proof transcripts provides a practical mitigation (post-quantum binding without post-quantum soundness), but the ideal construction would have soundness grounded in quantum-hard assumptions throughout. LaBRADOR provides this in principle, but current proof sizes (tens of kilobytes) are substantially larger than Groth16's 192 bytes. The research question is: can lattice-based proof systems achieve proof sizes competitive with Groth16 for ML-KEM-scale circuits (1.5 to 3 million constraints), and what are the precise constraint-size trade-offs?

2. Rejection-Free ML-KEM Decapsulation Circuits

ML-KEM decapsulation includes an implicit rejection mechanism (re-encapsulation and comparison) that introduces branch logic in the circuit. Branch logic whose outcome depends on a secret value creates a constraint formulation challenge: the circuit cannot simply branch on the secret without leaking information about which branch was taken. Non-deterministic hint techniques, in which the prover supplies auxiliary inputs that guide the circuit through the correct branch, have been proposed for similar constructions but have not been applied specifically to the ML-KEM implicit rejection mechanism. This is a concrete circuit design problem with a clear solution space.

3. Native Field Analogue of ML-KEM

The prime mismatch could be eliminated by designing a KEM that operates natively over the BN254 field prime p. Such a construction would have the same algorithmic structure as ML-KEM (polynomial ring, NTT-based multiplication, hash-based key derivation) but use p as the working modulus rather than q = 3,329. The open question is whether such a construction can achieve NIST-comparable security levels and whether its hardness can be grounded in standard lattice assumptions. A field-native ML-KEM analogue would reduce ML-KEM circuit constraint counts by two to three orders of magnitude, making sub-millisecond proving times achievable. This is a theoretical cryptography question as much as an engineering one.

4. Regulatory Acceptance Frameworks for ZK Compliance Evidence

The mathematical properties of hybrid SNARK+ML-DSA-65 transcripts are well-defined and the security reductions are clean. The regulatory properties are not. No EU AI Act implementing regulation, DORA technical standard, or HIPAA guidance addresses whether a Groth16 SNARK over an ML-DSA-65-signed Merkle root constitutes admissible audit evidence. The European Cybersecurity Agency (ENISA) has begun work on quantum-resistant cryptographic standards for regulated industries, and NIST's AI RMF 1.0 provides a framework for AI governance evidence but does not address cryptographic proof systems specifically. This is a policy and standardisation problem as much as a technical one, and it requires joint engagement between the cryptographic research community and the regulatory bodies.

5. Formal Privacy Analysis of Cross-Proof Leakage

The stateless verification property eliminates query-level metadata leakage. It does not eliminate leakage through correlations across proof transcripts. If a prover generates multiple ZK proofs using the same ML-KEM public key as a public input, a passive observer who sees the public inputs to multiple verifications can determine that they all involve the same party (because the public key is the same). Nullifier constructions can provide selective unlinkability within a context, but a formal model of cross-proof privacy for

NIST PQC ZK systems has not been published. The existing literature on anonymous credentials (Camenisch-Lysyanskaya, BBS+) provides relevant tools, but they were not designed with NIST PQC operations as circuit inputs.

6. Trusted Setup Ceremony Design for ML-KEM Circuits

Groth16 requires a circuit-specific trusted setup (Structured Reference String ceremony). An ML-KEM-768 encapsulation circuit with 2 to 3 million constraints requires an SRS of approximately 60 to 90 MB. The Powers of Tau ceremony format (used by Zcash, Hermez, and others) supports circuits of this scale, but each distinct ML-KEM circuit variant requires its own phase-2 ceremony. The practical question for organisations deploying multiple ML-KEM circuit variants (key validity, encapsulation, decapsulation, combined) is how to organise and run multiple ceremonies efficiently and how to structure the circuit library so that ceremonies can be reused across circuit variants sharing common subcircuits.

SECTION 12

Conclusion

The convergence of post-quantum cryptography and zero-knowledge proofs is not a theoretical curiosity. It is the response to a specific and documented gap in the current deployment model: PQC secures the channel but does not address the identity-layer and audit-layer harvest threat, while ZK proofs provide stateless privacy-preserving verification but have not, until recently, incorporated NIST-standardised post-quantum operations inside their circuits.

The research record in 2025 and 2026 shows multiple communities approaching the combination from different directions. The PLAZA project and LaBRADOR work on lattice-native ZK proof systems that share hardness assumptions with ML-KEM and ML-DSA. The rejection-free MLWE ZK framework improves the efficiency of lattice-based proofs. Kao's formal security models provide the audit evidence framework that regulatory bodies will require. Google's Groth16 proof of quantum cryptanalytic knowledge, and Trail of Bits' independent verification, validate the use of production SNARKs

in quantum-relevant contexts. The ZKMLOps work demonstrates enterprise demand for ZK-based AI compliance evidence. Each of these threads contributes to the overall convergence; none of them individually closes the gap.

The specific contributions that close the gap are: ML-KEM and ML-DSA operations expressed as R1CS constraints (addressing the prime mismatch with the strategies described in Section 4); a hybrid SNARK+ML-DSA-65 construction providing both ZK privacy and post-quantum binding (Section 6); stateless post-quantum verification as an architectural alternative to certificate-database-backed PKI (Section 7); and sublinear Merkle-anchored ML-DSA-65 attestation for long-lived compliance records (Section 8). AffixIO has built production infrastructure in this space; the architecture is documented across this whitepaper series without disclosing circuit implementation specifics.

The open problems of Section 11 are tractable. Fully post-quantum ZK soundness for ML-KEM circuits requires progress in lattice-based proof system efficiency. Regulatory acceptance frameworks require joint engagement between researchers and policymakers. Formal privacy models for cross-proof leakage require new definitions and security reductions. None of these is a fundamental impossibility; each is a concrete research programme with clear milestones. The practical deployment context, and the regulatory pressure that drives it, give these problems urgency that should attract sustained research attention over the next two to three years.

The harvest-now-decrypt-later threat to identity infrastructure is not a future problem. Data signed with ECDSA in 2026 may be retrospectively forgeable within the retention window of the regulatory obligations under which it is created. The combination of post-quantum cryptography with zero-knowledge proofs, applied to identity credentials and compliance audit records, is the technically correct response. The research community has the tools; the engineering community is building the infrastructure; the regulatory community is beginning to specify the requirements. What remains is the systematic work of connecting these three, which is precisely what the convergence of PQC and ZK proofs makes possible.

> This paper is part of AffixIO's whitepaper series on post-quantum ZK architecture. Related papers: WP-030: Stateless Post-Quantum Verification (formal model); WP-031: ML-KEM-768 in ZK Circuits (circuit architecture); WP-032: Sublinear Post-Quantum Attestation (Merkle-anchored ML-DSA-65).

FREQUENTLY ASKED

Common Questions

What is the verification gap between post-quantum cryptography and zero-knowledge proofs?

PQC secures the transport channel and authenticates documents against quantum attack. ZK proofs allow claims about private data to be verified without disclosure. The gap is that standard PQC deployment is stateful and infrastructure-bound, relying on certificate databases, OCSP responders, and revocation registries that create harvestable metadata. Closing the gap means expressing ML-KEM and ML-DSA operations as arithmetic circuit constraints, so that a prover can demonstrate correct execution of a post-quantum operation without revealing private key material, and the resulting proof carries post-quantum binding via ML-DSA-65 signing.

Why is it technically difficult to put ML-KEM inside a zero-knowledge circuit?

ML-KEM-768 operates modulo q = 3,329. Groth16 SNARKs operate over a 254-bit prime field. Every modular reduction must be enforced explicitly with auxiliary variables and range proofs, adding approximately 25 constraints per modular operation. The NTT at the heart of ML-KEM requires approximately 22,400 constraints per transform; a full encapsulation proof requires 1.5 to 3 million constraints versus roughly 3,000 for a classical Ed25519 signature in the same proof system. The circuits are feasible but significantly larger than classical analogues.

What does the 2026 research record say about combining PQC and ZK proofs?

Several threads are converging: the PLAZA ERC project on lattice-based ZK proofs; the rejection-free MLWE ZK proof framework (ePrint 2025/2239); Kao's formal security models for post-quantum audit evidence (arXiv 2512.00110); Google's April 2026 Groth16 proof of quantum cryptanalytic knowledge; and ZKMLOps work on AI compliance auditing. No prior publication combines production-grade ZK circuits with NIST-standardised ML-KEM and ML-DSA operations with fully post-quantum soundness. That specific gap remains the open research problem this series addresses.

Can the ZK+PQC architecture be deployed without replacing existing PKI?

Yes. The ZK+PQC architecture is an overlay deployment: ZK credential proofs can be issued for parties who hold existing PKI certificates, proving properties about those certificates inside a ZK circuit without changing the underlying certificate infrastructure. The stateless verification layer operates independently of and alongside the existing PKI. Migration of the PKI to ML-DSA-65 certificates is a separate and longer-term process; the ZK overlay provides post-quantum privacy and stateless verification benefits immediately, without waiting for the full PKI migration to complete.

WP-030: Stateless PQ Verification | WP-031: ZK PQC Circuits | WP-032: Sublinear Attestation

▶ About
▶ Solutions
▶ Legal
▶ Trust & Security

Contact

truth layer | yes | no | proof