# Post-Quantum Risk in Deployed Zero-Knowledge Architectures: A Layered Analysis

Andreas Renz

Encryptorium, Frauenfeld, Switzerland

**Abstract**

Zero-knowledge proof systems are now deployed widely in production cryptographic protocols, yet many rely on assumptions (discrete logarithms, pairings, or structured reference strings) that a fault-tolerant quantum computer would break via Shor's algorithm. This systematization of knowledge (SoK) presents a four-layer decomposition (L1–L4) that separates where quantum risk enters a proof system: arithmetization, polynomial commitment, protocol logic, and non-interactive compilation. Using a two-axis taxonomy that crosses *cryptographic impact* (structural break, modularly replaceable break, or quantitative degradation) with *deployment migration feasibility*, we classify the major proof-system families, derive a modularity test for evaluating upgrade paths, and introduce "collect now, forge later" (CNFL) as the proof-system analogue of harvest-now-decrypt-later. Published resource estimates place the cost of breaking 256-bit elliptic-curve discrete logs at 1,200–1,450 logical qubits, with the pairing-friendly curves underlying KZG (BN254, BLS12-381) of the same order of magnitude but somewhat larger [BZG+26]; under these estimates, such L2 constructions would face structural breaks once fault-tolerant hardware reaches that regime. Hash-based transparent systems, by contrast, degrade quantitatively under Grover-type speedups and QROM reduction losses rather than collapse. Case studies of Zcash, zkSync Era, and StarkNet show that practical post-quantum outcomes depend on deployment governance and upgrade architecture as much as on cryptographic primitives. The scope covers IOP/PCS-based and algebraic proof families; MPC-in-the-Head constructions are excluded.

**Keywords:** Post-Quantum Cryptography, Zero-Knowledge Proofs, Post-Quantum Zero-Knowledge, zk-SNARKs, zk-STARKs, Polynomial Commitments, Quantum Random Oracle Model, Interactive Oracle Proofs, FRI Soundness, Deployment Migration

Email: andreas@encryptorium.com (Andreas Renz)


Date of this document: 2026-06-30.

# Contents

# 1 Introduction

Zero-knowledge (ZK) proof systems let a prover convince a verifier that a statement is true without revealing the underlying witness. What began as a theoretical primitive is now a core building block in production systems, enabling privacy-preserving authentication, regulatory compliance with minimal disclosure, verifiable outsourced computation, and blockchain scalability [LHW+25, SAKK25].

Quantum computing changes the long-term security outlook for several assumptions that underpin widely deployed proof-system architectures. Shor's algorithm shows that integer factorization and discrete logarithms can be solved in polynomial time on sufficiently powerful quantum computers [Sho97]. Because many practical proof systems rely on assumptions in the discrete-logarithm family (including DLP, CDH, DDH, and pairing-based variants), quantum adversaries raise a concrete design question: which proof-system architectures suffer a *structural break*, which ones can be *migrated by swapping components*, and which ones *degrade gracefully* and can be recovered by parameter changes?

"Post-quantum ZK" is not a single property. Modern proof systems are layered constructions. A typical deployed zkSNARK/zkSTARK stack combines: (i) an arithmetization of computation (constraint encoding), (ii) a commitment and consistency layer that lets the verifier reason about large algebraic objects via small queries, (iii) a protocol layer (often IOP/PCS-based) that enforces identities and soundness, and (iv) a non-interactive transformation, commonly Fiat-Shamir [FS87]. The quantum resilience of the overall system therefore depends on *which layer carries security* and *which assumptions are ultimately required*. We make this explicit using the L1–L4 decomposition introduced in §2 (see table 3).

In practice, the boundaries between layers are not always cleanly separable; for instance, FRI serves both as a commitment mechanism (L2) and as a core component of the protocol logic (L3). The layered decomposition is an analytical lens, not a claim that every system factors into four independent modules. Its value lies in identifying where quantum-vulnerable assumptions enter and whether they can be isolated for replacement.

## 1.1 Problem statement

We study the following question:

> How does quantum computation affect the security of modern zero-knowledge proof systems? Which architectural choices determine whether a deployed system experiences a structural break, a modularly replaceable break, or quantitative degradation under quantum adversaries, and which deployment characteristics determine whether migration is feasible in practice?

## 1.2 Contributions

This paper delivers three concrete outputs: an analytical framework, a comparative analysis that applies it to deployed proof-system families, and case studies that connect the analysis to real deployments. The analysis also introduces "collect now, forge later" (CNFL), the proof-system analogue of harvest-now-decrypt-later, as the threat model that makes the deployment-migration axis operationally meaningful (§6.1).

1. An *analytical framework* with two components. First, we decompose proof systems into arithmetization, commitment/consistency, protocol, and non-interactive layers (L1–L4), so that end-to-end soundness can be tracked to the specific assumptions it depends on. Second, we formalize a two-axis classification that separates the cryptographic effect of quantum attacks (structural break, modularly replaceable break, quantitative

degradation) from deployment migration feasibility, and we define a modularity test that distinguishes "replaceable in theory" from "migratable in practice" (§2.4).

2. A *comparative analysis* of representative deployed families (pairing-based SNARKs, Plonky2/3, Halo 2, circle STARKs, Binius, and lattice-based directions). Beyond classification, we derive system-level results not available from the primary sources in isolation: a Grover-adjusted accounting of hash-based FRI soundness (a $k$-bit grinding target contributes only $\approx k/2$ post-quantum bits and must be doubled), a post-quantum margin assessment of the ethSTARK parameterization, and a separation between the proven Fiat-Shamir security of FRI pipelines and the still-conjectured interactive FRI soundness on which deployed bit-security figures rest. Other families receive taxonomic classification with system-specific PQ notes identifying open verification targets (§5).
3. *Case studies and a migration playbook.* We apply the modularity test to three real deployments (Zcash Sapling, zkSync Era, StarkNet) and derive stakeholder-specific guidance for protocol designers, library maintainers, rollup operators, auditors, and governance owners (§6).

## 1.3 Terminology conventions

We distinguish two levels of post-quantum assurance throughout this paper. *PQ-plausible* means no known polynomial-time quantum attack; security rests on assumptions conjectured to be quantum-hard (e.g., hash preimage and collision resistance, Module-LWE). This is the default term for hash-based IOP systems and lattice-based candidates. *PQ-secure (under assumption $X$)* means a formal reduction to a named assumption $X$ conjectured to be quantum-hard, with an explicit proof in a quantum-aware model, for example "PQ-secure under collision resistance of SHA-3 in the QROM." We use PQ-plausible as the default and reserve PQ-secure for statements backed by explicit reductions.

## 1.4 Related work

This paper is a systematization of knowledge (SoK) for the post-quantum status of the architectures that dominate current succinct proof deployments, especially IOP/PCS-based and algebraic stacks. It should not be read as a full survey of all post-quantum proof-system paradigms. We position this work relative to four strands of existing literature.

**Proof-system surveys.** Recent systematizations bridge research and practice for zkSNARKs, covering assumptions, setup models, and implementation concerns [LHW+25]. Broader surveys compare proof-system families and application domains [ORES24, SAKK25]. These provide architectural context but do not perform per-layer quantum vulnerability analysis.

**PQC surveys and migration guidance.** The PQC literature surveys quantum attacks and candidate assumption families [BBD09, PPG24, CSD17, Fed21] and focuses on primitives (KEMs, signatures, hash functions) rather than on the composed proof-system architectures studied here. The closest neighbor on the resource-estimate axis is the ledger-level quantum-risk estimate of Babbush et al. [BZG+26], which quantifies the qubit cost of the elliptic-curve discrete-log attack against deployed cryptocurrencies; we instead analyze quantum risk per proof-system layer (L1–L4).

**QROM and Fiat-Shamir transfer results.** Several results address the ROM-to-QROM transfer problem for Fiat-Shamir compilation [DFMS19, DFM20, Zha18, Unr17]. We draw on these results at L4 but emphasize that their applicability to specific deployed compilers is often not automatic and requires system-specific verification.

**PQ status of specific proving systems.** Some works touch on the post-quantum status of individual systems or families: the STARK original paper [BBHR18] includes "post-quantum" in its title, and recent work on quantum rewinding for IOP-based arguments [CDD+25] advances the theoretical foundations but does not address deployment-level migration questions. These treatments typically address a single system rather than providing a cross-family comparative analysis with deployment-oriented migration criteria.

Table 1 maps the features of this paper against the closest related works across the four strands above. To our knowledge, no single existing work combines per-layer (L1–L4) quantum vulnerability analysis, a deployment-oriented modularity criterion that separates structural breaks from replaceable ones, first-class treatment of the oracle model (ROM vs. QROM) as an independent analysis dimension, and concrete case studies of deployed systems. This paper addresses that gap for the IOP/PCS-based and algebraic proof-system families that dominate current deployments.

**Table 1:** Feature coverage of related works versus this paper. A ✓ indicates the feature is substantively addressed; (✓) indicates partial coverage; × indicates the feature is absent.

| Work | *Per-layer (L1–L4)* | *Cross-family comparison* | *QROM as first-class* | *Migration criteria* | *Case studies* |
|---|---|---|---|---|---|
| Liang et al. [LHW+25] | × | ✓ | × | × | (✓) |
| Oude Roelink et al. [ORES24] | × | ✓ | × | × | × |
| Sheybani et al. [SAKK25] | × | ✓ | × | × | (✓) |
| Bernstein & Lange [BBD09] | × | × | × | (✓) | × |
| Don et al. [DFMS19] | × | × | ✓ | × | × |
| Ben-Sasson et al. [BBHR18] | × | × | × | × | (✓) |
| Chiesa et al. [CDD+25] | (✓) | × | (✓) | × | × |
| **This paper** | ✓ | ✓ | ✓ | ✓ | ✓ |

## 1.5 Scope and limitations

This work does not propose new cryptographic constructions. It is a structural analysis of existing deployed proof systems under quantum threat models, focusing on the IOP/PCS-based and algebraic proof-system architectures that dominate current succinct proof deployments.

We focus on polynomial-time quantum adversaries in the standard quantum circuit model. We adopt a capability-based framing: systems designed today should remain secure against quantum capabilities that may emerge during their operational lifetime. Implementation-level vulnerabilities, side channels, and engineering failures are out of scope.

Where relevant, we note when an argument relies on oracle-model assumptions and may require QROM-specific care [BDF+10]. We do not treat "quantum ZK" (constructions that use quantum communication or quantum verifiers) and we do not make specific hardware timeline predictions.

We do not cover MPC-in-the-head (MPCitH) or symmetric-key ZK constructions (e.g., KKW [KKW18], Ligero [AHIV22], Limbo [GOT21]) or purely symmetric-key-based ZK approaches. These are arguably the most naturally post-quantum ZK constructions, since they rely entirely on symmetric primitives. Our analytical framework targets the

IOP/PCS layered architecture that characterizes the most widely deployed systems today; a systematic PQ analysis of MPCitH-style constructions would require a different structural decomposition and is left for future work.

### 1.6 Paper organization and reading guide

§2 defines terminology, adversary models, the L1–L4 lens, and the impact taxonomy. §3 introduces classical proof-system families and the layered architecture. §4 summarizes quantum threats and the PQC migration toolbox needed for the impact analysis. §5 applies the layered analysis to major proof-system architectures with concrete technical depth. §6 discusses present-day consequences, case studies, and practical recommendations. §7 identifies research directions and limitations of this work, and §8 summarizes the findings.

Table 2 provides role-specific entry points for readers with different backgrounds.

**Table 2:** Reader's map (where to start depending on role).

| **If you are...** | **Start with...** |
|---|---|
| A protocol designer choosing a proof system | §2 (taxonomy + L1–L4 lens), §5 (impact by architecture), §6 (case studies + migration). |
| A library maintainer (arkworks, halo2, Plonky2/3) | §3 (interfaces and layers), §5 (L2/L4 analysis with concrete bounds), §6 (engineering trade-offs). |
| A rollup / on-chain deployer | §2 (modularity test), §5 (which stacks collapse), §6 (case studies + migration playbook). |
| A security auditor | §2 (adversary model + taxonomy), §5 (assumption chains, attack costs, and break modes). |
| A reader who wants quantum-hardware context | §4 (algorithms and hardware reality check). |

## 2 Threat model and analytical framework

The notation and analytical framework below are used throughout the paper. We assume familiarity with zero-knowledge proofs and modern succinct arguments (argument systems with short proofs and fast verification); only definitions needed for the post-quantum analysis are made explicit. A compact glossary of abbreviations appears in appendix A.

We reserve *zero-knowledge* for the property that a proof reveals nothing beyond the validity of the statement (formally, the existence of a simulator). Many systems analyzed here are deployed as succinct *arguments* whose security goal is soundness or knowledge-soundness rather than hiding; validity rollups, for instance, reveal their inputs. We therefore write *proof system* or *argument system* for property-agnostic claims and use *zero-knowledge* only for statements about the hiding property itself, analyzed in §5.4. Established names such as zk-SNARK, zk-STARK, and zkVM are retained as proper nouns.

### 2.1 Notation

We write $\lambda$ for the security parameter. A function $\mu(\lambda)$ is *negligible* if it decreases faster than any inverse polynomial. Finite fields used for arithmetization are denoted $\mathbb{F}$. We write $q_H$ for the adversary's bound on (quantum) oracle queries and $n$ for hash output length in bits. Unless stated otherwise, "efficient" refers to algorithms running in probabilistic polynomial time in $\lambda$ [BS23, PPG24].

## 2.2 Adversary and oracle models

Throughout this paper: A classical adversary is modeled as a probabilistic polynomial-time (PPT) algorithm. A quantum adversary runs in quantum polynomial time and may use algorithms such as Shor's and Grover's [Sho97, Gro96].

Unless stated otherwise, *post-quantum security* means security against a quantum adversary interacting with classical inputs and outputs (classical statements, classical proofs, classical verification), but possessing quantum computational power internally. This is the standard attacker model in PQC migration discussions [Fed21, CSD17].

Most practical systems are rendered non-interactive via the Fiat-Shamir transform [FS87]. Security proofs are typically given in the *Random Oracle Model* (ROM). In the quantum setting, adversaries may query the oracle in superposition; the resulting model is the *Quantum Random Oracle Model* (QROM) [BDF+10, Unr14, Unr17].

For Fiat-Shamir-based constructions, post-quantum security depends both on the underlying hardness assumption and on whether the ROM-based reduction extends to the QROM. This distinction is the primary analysis boundary at Layer L4.

## 2.3 Layered decomposition (L1–L4)

We analyze proof systems through four layers, summarized in table 3. The decomposition is an analytical lens: in practice, layer boundaries are not always cleanly separable (e.g., FRI serves as both a commitment mechanism and a protocol component), but the factoring identifies where quantum-vulnerable assumptions enter and whether they can be isolated for replacement.

**Table 3:** Layered decomposition for post-quantum analysis.

| | |
|---|---|
| **L1** | Arithmetization / encoding: computation → constraints or polynomials over a field. |
| **L2** | Commitment + consistency: binding commitments and consistency mechanisms (PCS, Merkle+FRI, IPA-style, etc.). |
| **L3** | Protocol logic: IOP/argument structure enforcing identities and soundness amplification. |
| **L4** | Non-interactivity + extraction model: Fiat-Shamir, oracle assumptions (ROM/QROM), and extractability. |

Each layer carries a distinct role in the security argument.

L1 encodes computation as algebraic constraints over $\mathbb{F}$, using formats such as R1CS/QAP (circuit-based) or AIR (trace-based). It determines what is being proved but is not typically where post-quantum breaks originate.

L2 is the load-bearing layer for post-quantum analysis. The prover commits to large objects and proves small consistency claims. *Binding* (or an equivalent unforgeability guarantee) is the primary property for soundness; *hiding* is required for zero-knowledge. In modern succinct systems, L2 is typically a polynomial commitment scheme (PCS) or a Merkle+FRI-based commitment interface. The canonical pairing-based PCS is KZG [KZG10], which provides constant-size openings but whose binding is inseparable from pairing-group hardness assumptions.

L3 uses L2 to enforce global correctness through limited checks: polynomial identities in SNARK-style systems, or low-degree/proximity structure in STARK/IOP-style systems. It determines how soundness error composes and what the transcript structure looks like for Fiat-Shamir at L4.

L4 is where most deployed systems apply the Fiat-Shamir transform [FS87]. Classical security arguments work in the ROM; the post-quantum setting requires the QROM [BDF+10, Unr17].

Two security properties are tracked through these layers: *soundness* (no cheating prover can convince the verifier of a false statement) and *knowledge-soundness* (an efficient extractor can recover a valid witness from any prover that produces a convincing proof). The distinction matters because quantum adversaries affect extraction most directly: classical rewinding-based extractors interact poorly with superposition and no-cloning (§5.3).

**Definition 1** (Component)**.** A *component* is the concrete cryptographic primitive or sub-protocol that instantiates a single layer of the decomposition above: an arithmetization at L1, a commitment and consistency mechanism at L2 (e.g., KZG, an IPA-based scheme, or Merkle+FRI), an interactive protocol at L3, or a Fiat-Shamir instantiation at L4. A component is *replaceable* when it exposes a stable interface to its layer, so that substituting another component for the same layer leaves the statement semantics and the remaining layers unchanged. The conditions under which a deployed stack admits such substitution are formalized by the modularity test (definition 5).

## 2.4 Impact taxonomy

Quantum attacks affect proof systems in qualitatively different ways. The common binary distinction ("collapse" vs. "degradation") is too coarse for architectural decisions. We introduce a two-axis classification: *Axis 1* captures the cryptographic effect of a quantum attack on the proof system itself, and *Axis 2* captures the deployment migration feasibility of addressing it. The practical outcome for a deployed system is determined by the combination of both axes.

The classification is an analytical framework for assessment, not a formal theorem taxonomy: the assignments it produces depend on current knowledge of attacks, reductions, and deployment architectures, and individual entries may shift as the literature evolves. We use it to structure the per-family analysis in §5 and the case studies in §6.

Unless stated otherwise, the headline classifications refer primarily to soundness and knowledge-soundness under quantum adversaries. Zero-knowledge, simulation-extractability, and stronger composition properties may follow a different post-quantum status and are treated separately in §5.4 and §7.

### 2.4.1 Axis 1: Cryptographic effect

**Definition 2** (Structural break)**.** A proof system suffers a *structural break* under a quantum adversary if the soundness or knowledge-soundness of its verification semantics relies on an assumption broken in quantum polynomial time, and that assumption is not cleanly replaceable without changing the system's verification interface or statement semantics.

*Clarification.* We distinguish *statement semantics* (what is being proved: the constraint system, public inputs, and the logical claim) from *verification mechanics* (how the proof is checked: the algorithms, group operations, and commitment-opening procedures). A break is structural when the quantum-vulnerable assumption is fused into the statement or verification equation itself, so that the arithmetization (L1) and the statement being proved cannot remain unchanged while L2 is swapped. If L2 can be replaced while preserving the same arithmetization and statement (changing only verification mechanics), the break is modularly replaceable (definition 3).

**Definition 3** (Modularly replaceable break)**.** A proof system suffers a *modularly replaceable break* if a quantum-broken component is critical for soundness, but the overall stack admits principled migration by replacing that component while preserving most of the surrounding architecture. Survival requires swapping a module, not merely increasing parameters.

**Definition 4** (Quantitative degradation)**.** A proof system exhibits *quantitative degradation* if quantum attacks reduce concrete security (e.g., via Grover-type speedups), and security can be restored through parameter increases without replacing the core commitment or verification interface.

*Adversary and oracle model.* Throughout, "quantum adversary" refers to the quantum polynomial-time model of the threat model above; the applicable idealized model depends on the layer under analysis: the standard model for L2 algebraic assumptions (e.g., Shor on discrete log) and the ROM or QROM for L4 Fiat-Shamir-based claims.
*Meaning of "security" in this taxonomy.* When a component change *preserves* or *re-establishes* security, or a parameter increase *restores* it, we distinguish the two evidence levels defined in appendix A: *PQ-secure* (a formal reduction to a conjectured quantum-hard assumption exists in the relevant standard/ROM/QROM model) and the weaker *PQ-plausible* (no known polynomial-time quantum attack, but no such reduction). A migration target is PQ-secure where a reduction is available and PQ-plausible otherwise; the per-family analysis in §5 states which level applies in each case. Re-establishing security thus means restoring a system to the strongest of these two levels that current results support, not merely the absence of a specific attack.

### 2.4.2 Axis 2: Deployment migration feasibility

The cryptographic classification alone does not determine the practical outcome. A system with a modularly replaceable break may still fail to migrate if deployed verifiers cannot be upgraded. We assess deployment feasibility along three levels:

*Upgrade-ready* deployments have explicit mechanisms for verifier replacement, state continuity across versions, and defined transition protocols. *Upgrade-constrained* deployments have some upgrade path but face governance, coordination, or engineering obstacles (e.g., requiring a hard fork, multi-party coordination, or re-engineering). *Upgrade-blocked* deployments have no practical upgrade path without replacing the entire system (e.g., an immutable on-chain verifier with no proxy or governance mechanism).

### 2.4.3 Combined outcome

The practical outcome for a deployed system is the combination of both axes. Table 4 maps the nine possible combinations to practical outcomes.

**Table 4:** Two-axis outcome map: cryptographic effect × deployment feasibility.

| | **Upgrade-ready** | **Upgrade-constrained** | **Upgrade-blocked** |
|---|---|---|---|
| **Structural break** | System replacement (feasible but heavy) | **Total collapse** | **Total collapse** |
| **Modularly replaceable break** | Managed migration (PCS swap + verifier upgrade) | Difficult migration (governance bottleneck) | Practical failure (theory allows, deployment blocks) |
| **Quantitative degradation** | Parameter tightening | Parameter tightening (with coordination cost) | Degraded but operationally stuck |

For conciseness, we use the shorthand labels in table 5 in tables and running text throughout the paper.

**Table 5:** Shorthand labels used in tables and running text.

| Shorthand | Formal term | Meaning |
|---|---|---|
| Total collapse | Structural break + upgrade-constrained/blocked | System failure; no viable in-place migration. |
| Modular collapse | Modularly replaceable break | Broken component can be swapped; migration required. |
| Graceful degradation | Quantitative degradation | Security reduced but recoverable via parameter increases. |
| CNFL | "Collect now, forge later" | Proof-system analogue of HNDL; future proof forgery against legacy verifiers (§6). |

### 2.5 Post-quantum modularity test

The impact taxonomy classifies what happens; the modularity test determines whether migration is feasible for a specific deployment. We formalize a practical criterion that captures the deployment axis.

**Definition 5** (Post-quantum modularity test)**.** A deployed proving stack is *PQ-modular* if:

1. Replaceable commitment interface: the commitment and consistency mechanism (polynomial commitment scheme, Merkle+FRI, vector commitment, or analogous primitive) is modular and exposes a stable verification interface.
2. Replaceable Fiat-Shamir layer: transcript hashing and challenge derivation can be re-instantiated without altering the statement being proved.
3. Upgradable verification logic: deployed verifiers (e.g., smart contracts, embedded clients, audited libraries) can be versioned or replaced without invalidating long-lived commitments.

Failure of any condition constrains migration feasibility, potentially making a cryptographically replaceable break operationally irrecoverable.

Condition (3) involves several operational requirements in practice:

(3a) the verification logic resides behind an upgrade mechanism such as a proxy contract, a hard-fork path, or a governance-controlled deployment;

(3b) there is a defined transition protocol for moving between verifier versions without state loss;

(3c) historical commitments (e.g., state roots, attestation records) remain *parseable* by new verification logic; and

(3d) historical proofs remain *verifiable* under the new verifier, or a defined re-proving protocol exists that allows holders of valid witnesses to produce new proofs accepted by the successor verifier.

We illustrate this modularity test through three deployment case studies in §6.3.

## 3 Proof-system families

The proof-system families below are organized through the L1–L4 decomposition from §2.3. For each family, we identify the concrete mechanisms at each layer so that the quantum impact analysis in §5 has clear targets. Readers already familiar with the systems may skip to table 6 for the summary.

## 3.1 Proof-system families considered

The families below were selected to cover the commitment mechanisms that drive post-quantum outlook in current practice. We distinguish *deployed families* (pairing-based, PLONK-style with KZG, IPA-based discrete-log systems, and hash-based STARK/IOP systems) from *research-direction families* (binary-tower constructions and lattice-based candidates) that are actively developed but not yet in widespread production use. The split matters for the analysis in §5: deployed-family classifications bear on migration decisions now, while research-family classifications indicate which directions may be viable PQ successors. We use the following high-level labels throughout. *Pairing-based SNARKs* are succinct arguments built from elliptic-curve pairings [Gro16]. *PLONK-style systems* are universal-setup arguments with PLONKish arithmetization [GWC19]; they are commonly instantiated with KZG at L2, but the PCS is architecturally separable. *Transparent IOP/STARK-style systems* are transparent constructions based on interactive oracle proofs and hash-based commitments, including classic STARK pipelines [BBHR18, BGKS19], Plonky2/3 [Pol22] (FRI over small fields with recursion), and circle STARKs [HLP24]. *Discrete-log transparent systems* are IPA-based systems like Bulletproofs [BBB$^+$17] and Halo 2 (building on Halo [BGH19]), which avoid trusted setup but rely on discrete-log hardness. *Binary-tower systems* are represented by Binius [DP23], which operates over binary extension fields with distinct algebraic properties. *Lattice-based proof systems* are candidates whose security reduces to lattice problems [Lyu11, ISW21]; these remain research-stage, with larger commitment sizes than pairing-based schemes.

Concrete instantiations and parameter choices are introduced when they become relevant to the quantum analysis.

## 3.2 Architectural comparison by layer

Table 6 maps each system family to its concrete mechanisms at each layer. The L2 column is the primary determinant of post-quantum outlook: families whose L2 binding relies on discrete-log or pairing assumptions face structural or replaceable breaks under Shor, while hash-based L2 mechanisms face only quantitative degradation. At L4, all systems except Groth16 use Fiat-Shamir compilation, so the QROM obligation is near-universal.

**Table 6:** Proof-system families mapped to L1–L3 mechanisms. Bold entries mark the layer carrying the dominant PQ-relevant assumption. All systems except Groth16 use Fiat-Shamir compilation at L4 (ROM); Groth16 achieves non-interactivity through its algebraic NIZK structure (CRS, no FS).

| System | L1 | L2 | L3 |
|---|---|---|---|
| Groth16 | R1CS/QAP | **Pairing verification** | Algebraic NIZK |
| PLONK+KZG | PLONKish | **KZG PCS** | Polynomial identities |
| STARK-style | AIR | **Merkle + FRI** | IOP (proximity + consistency) |
| Plonky2/3 | PLONKish / AIR | **FRI (Goldilocks)** | IOP + recursion |
| Circle STARKs | AIR | **Circle-FRI + Merkle** | IOP (circle-domain) |
| Halo 2 | PLONKish | **IPA (DL-based)** | Poly. identities + accumulation |
| Bulletproofs | R1CS | **IPA / DL commits** | Inner-product argument |
| Binius | Multilinear | **Hash-based / Merkle** | Multilinear poly IOP + proximity |
| Nova [KST21] | R1CS | **Pedersen vector commits (DLP)** | Folding |
| Lattice PCS | Varies | **Lattice assumptions** | Varies |

## 3.3 Representative families by L2 choice

Since the L2 mechanism drives the post-quantum outlook, we briefly characterize each family's architectural choices and the assumptions they carry.

**Pairing-based SNARKs.** Groth16 [Gro16] combines R1CS/QAP arithmetization with pairing-based verification, yielding small proofs (≈128–192 bytes compressed, depending on curve; three group elements per proof [Gro16]) but inheriting pairing-group assumptions at L2. Groth16 achieves non-interactivity through its algebraic NIZK structure rather than Fiat-Shamir, so L4 is not a separate fault line: the pairing assumption carries the full security burden.

**PLONK-style with KZG.** PLONK-style systems [GWC19] target universal setups with PLONKish arithmetization. When instantiated with KZG at L2, binding reduces to the $d$-Strong Diffie–Hellman ($d$-SDH) assumption in the SRS group [KZG10]. The architectural distinction from Groth16 is that the PCS interface is separable: L1 and L3 do not intrinsically depend on the choice of L2, making PCS replacement a viable migration path.

**Transparent IOP/STARK-style.** STARK-style systems [BBHR18] use hash-based Merkle commitments and FRI proximity testing [BGKS19] at L2, with no trusted setup and no algebraic hardness assumptions at the commitment layer. We note that FRI straddles L2 and L3 in our decomposition: the proximity test is both a commitment consistency mechanism and a structural component of the IOP, so placing it under L2 is an analytical choice rather than a canonical factoring. Plonky2/3 combines FRI over the Goldilocks field with PLONK-style arithmetization for recursive composition [Pol22]; Plonky3 is its modular successor [Pol26]. Circle STARKs [HLP24] adapt FRI to circle groups on Mersenne prime fields.

**Discrete-log transparent systems.** Halo 2 (building on Halo [BGH19]) and Bulletproofs [BBB+17] use inner-product arguments (IPA) over prime-order elliptic-curve groups. These are transparent (no trusted setup) but their commitment binding relies entirely on discrete-log hardness, making them Shor-vulnerable at L2. Halo 2 adds an accumulation scheme for recursive composition, which further propagates the DLP dependency through the recursion chain.

**Binary-tower systems.** Binius [DP23] operates over binary tower fields ($\mathbb{F}_{2^k}$) with hash-based commitments. Its polynomial IOP structure differs from standard FRI (it uses multilinear extensions with distinct evaluation and proximity arguments), so its soundness guarantees require independent analysis.

**Lattice-based directions.** Lattice-based proof systems [ISW21] offer a plausible foundation for PQ-secure commitments at L2. Earlier lattice polynomial commitments reported megabyte-scale proofs [HSS24], but 2024–2026 constructions have brought proof sizes into the tens of kilobytes at practical degrees (Greyhound composed with LaBRADOR [NS24], and the Hachi multilinear construction [NOZ26]), closing much of the gap to hash-based alternatives while remaining larger than KZG's constant-size openings. The evolving state of the art is discussed in detail in §4.2.

The post-quantum consequences of each layer are analyzed in §5, which includes a consolidated classification table (table 10) mapping each family to its setup model, L2 mechanism, and two-axis PQ impact assessment.

# 4 Quantum threats and PQC toolbox

The three quantum phenomena below drive all layer classifications in §5; the PQC toolbox that follows frames the available defenses as targeted module replacements.

## 4.1 Quantum threats

**Shor's algorithm.** Shor's algorithm solves integer factorization and discrete logarithms in polynomial time [Sho97]. In Shor's original analysis, the algorithm runs in $O(n^2 \log n \log\log n)$ gate operations using $O(n)$ logical qubits for an $n$-bit input; concrete resource estimates for specific key sizes vary by more than an order of magnitude across hardware assumptions (see the hardware discussion below). Any construction whose security depends on factoring or discrete log suffers a structural break under a sufficiently capable fault-tolerant quantum computer. For proof systems, this matters because widely deployed stacks embed discrete-log-type assumptions at L2. When those assumptions are the source of binding or evaluation soundness, security is not recovered by parameter tuning.

For pairing-based proof systems, the vulnerability is direct: Shor solves the discrete-log problem in the source groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of the pairing. Once discrete log is broken in the base group, all $q$-type assumptions that derive their hardness from DLP collapse as well. In particular, the $d$-Strong Diffie–Hellman ($d$-SDH) assumption underlying KZG binding [KZG10] and the knowledge-of-exponent-type assumptions used in Groth16 [Gro16] are all Shor-vulnerable, because an adversary who can compute discrete logs can solve $d$-SDH directly. This is the mechanism behind the structural-break classification in §5.1: the break is not specific to one assumption but affects the entire family of DLP-derived hardness problems.

**Grover and collision-related speedups.** Grover's algorithm provides a quadratic speedup for unstructured search [Gro96]: $k$-bit preimage security is reduced to at most $k/2$ bits against Grover-type adversaries. Quantum collision-finding algorithms reduce collision-resistance bounds to at most $\approx 2^{n/3}$ queries for $n$-bit outputs [BHT98, Zha15]. For multi-target preimage search with $M$ known targets, the query complexity improves to $O(\sqrt{N/M})$ [BBHT98], but this does not change the asymptotic impact on hash-based proof-system components: the speedup remains quadratic, not exponential.

Grover's speedup applies to any brute-force search; hash inversion is one instance. In the proof-system context, this extends to grinding (proof-of-work) steps used by some STARK implementations to amplify soundness: a $k$-bit grinding target provides only $\approx k/2$ bits of PQ security (see §5.5 for the detailed analysis).

The BHT $2^{n/3}$ collision bound assumes access to large quantum random-access memory (QRACM). Whether QRACM is practically realizable at cryptographic scales remains debated; if it is not, the effective collision-finding complexity may be closer to $2^{2n/5}$ or higher [CNS17], making the $2^{n/3}$ bound a conservative (worst-case) estimate for deployment planning.

For hash-dominated proof-system stacks, the main post-quantum effect is quantitative: security can be restored by increasing parameters at a performance cost.

**Oracle access and the QROM.** Classical Fiat-Shamir security arguments work in the ROM. In the quantum setting, adversaries may query the oracle in superposition, yielding the QROM [BDF+10]. Concretely, a quantum adversary submits a quantum state $\sum_x \alpha_x |x\rangle$ to the oracle and receives $\sum_x \alpha_x |x, H(x)\rangle$; the consequence is that the adversary's queries are not individually observable by the extractor or simulator. Attempting to measure a superposition query to learn which inputs were queried collapses the adversary's state, potentially invalidating any subsequent rewinding argument.

This breaks the standard ROM proof strategy for knowledge extraction. In a typical ROM extractor, the proof is rewound to a point where the adversary re-queries the oracle, and the extractor reprograms the oracle's response to extract a witness from the diverging transcripts. In the QROM, two obstacles arise: (i) the quantum no-cloning theorem prevents the extractor from copying the adversary's state before rewinding, so each rewind attempt irreversibly alters the state; and (ii) measuring which query the adversary made (to decide how to reprogram) disturbs the superposition.

These obstacles produce a characteristic security loss: QROM reductions for Fiat-Shamir on $\Sigma$-protocols lose a factor of $\approx q_H^2$ (where $q_H$ is the adversary's query bound) compared to the $q_H$ loss in the ROM. Intuitively, the extractor must guess which of the adversary's $q_H$ queries to "measure," succeeding with probability $\approx 1/q_H$ per attempt, which compounds into a quadratic overhead. The concrete implications for parameter selection are analyzed in §5.2.3.

The modern QROM proof toolkit comprises three main threads: (i) DFMS-style results covering Fiat-Shamir on $\Sigma$-protocols [DFMS19]; (ii) measure-and-reprogram techniques for multi-round protocols [DFM20]; and (iii) Zhandry's compressed-oracle framework for recording quantum oracle queries [Zha18]. These tools are deployed at L4 in our analysis.

**Hardware reality and planning context.** Implementing Shor at cryptographically relevant sizes requires fault-tolerant quantum computation, and the resource estimates depend strongly on the target problem. Gidney and Ekerå estimate roughly $2 \times 10^7$ physical qubits to factor RSA-2048 under superconducting surface-code assumptions [GE21]; a 2025 revision lowers this to fewer than $10^6$ physical qubits, trading the reduction for a multi-day rather than multi-hour runtime [Gid25] (a preprint pending peer review). For the 256-bit elliptic-curve discrete-log problem underlying secp256k1 and pairing-friendly curves of similar size, Babbush et al. report fewer than $5 \times 10^5$ physical qubits at $10^{-3}$ error rate,

with 1,200–1,450 logical qubits [BZG+26] (an industry whitepaper, not externally peer reviewed). Current devices are transitioning from the NISQ regime [Pre18] toward early fault tolerance: the first below-threshold surface-code error correction on a superconducting processor was demonstrated in 2024 [AAA+25], though logical-qubit counts remain far below those required for cryptographically relevant attacks.

*Evidence status: established (attack facts), heuristic (planning horizon).* The distinction we draw is between *formal attack facts* (Shor and Grover solve their respective problems; the speedups are polynomial-time and quadratic respectively) and *engineering planning implications* (when cryptographically relevant hardware arrives, and how deployments should prepare). The attack facts are model-independent and settled; the planning implications depend on hardware forecasts and policy timelines that are separately sourced and carry correspondingly weaker evidence. For long-lived systems, the existence of polynomial-time quantum attacks and multi-year migration timelines is sufficient to justify proactive architectural design today, even without committing to a specific hardware forecast.

## 4.2 Post-quantum cryptography toolbox

The remainder of this section frames PQC as a targeted module-replacement toolbox for proof-system stacks.

**Scope of standardized PQC.** NIST has finalized PQ standards for key establishment (FIPS 203) and signatures (FIPS 204, FIPS 205) [Nat24b, Nat24a, Nat24c]. These cover secure communication and authentication but do not make a proof system post-quantum secure. A rollup may authenticate upgrades with PQ signatures while still relying on Shor-vulnerable assumptions inside its proof system. PQ security requires proof-system-specific replacements at L2 and QROM-aware arguments at L4.

*Evidence status: heuristic (policy and planning horizon).* The 2030 and 2035 dates that appear throughout post-quantum migration guidance should be read as policy and planning anchors rather than settled cryptographic facts. There is no scientific consensus on when a cryptographically relevant quantum computer (CRQC) will exist, but policy bodies across jurisdictions converge on a common migration window. The UK National Cyber Security Centre's March 2025 migration roadmap sets explicit milestones of 2028 (discovery and planning), 2031 (high-priority migrations), and 2035 (full migration across systems, services, and products) [Nat25]. On the US side, NIST IR 8547 (initial public draft, November 2024; public comment period closed January 2025) proposes deprecating $\leq 112$-bit asymmetric algorithms after 2030 and disallowing quantum-vulnerable algorithms in NIST standards after 2035 [MPR+24]; the document remains in draft as of this writing, with a final version expected. On the EU side, the NIS Cooperation Group PQC work stream published Part 1 of the "Coordinated Implementation Roadmap for the Transition to Post-Quantum Cryptography" in June 2025 (implementing Commission Recommendation (EU) 2024/1101), which sets milestones of end-2026 (national transition roadmaps established, high- and medium-risk pilots initiated), end-2030 (PQC transition for high-risk use cases completed, quantum-safe software and firmware upgrades enabled by default), and end-2035 (PQC transition for medium-risk use cases completed) [NIS25]. The EU document explicitly targets alignment with the NCSC and NIST IR 8547 2035 endpoints. These documents are not hardware forecasts; they are engineering planning horizons chosen because the transition from standardization to fully deployed migration has historically taken on the order of a decade or more, and harvested ciphertext creates present-day exposure regardless of when fault-tolerant hardware arrives. The most aggressive recent resource estimate for the 256-bit ECDLP attack places the sufficient cost at 1,200–1,450 logical qubits and under $5 \times 10^5$ physical qubits on superconducting architectures [BZG+26]. We observe that hardware at this scale, if realized, would arrive within the design lifetime of systems

being deployed today. For systems whose migration difficulty is high (§6.4), this convergent window reinforces the case for beginning architectural preparation now, independently of the precise date.

**Isogeny-based candidates and the SIDH lesson.** The SIDH/SIKE isogeny-based key-exchange scheme, once a leading PQC candidate in the NIST standardization process, was broken in 2022 by a classical (non-quantum) attack exploiting the auxiliary torsion-point information published as part of the public key [CD23]. PQC candidates can fail unexpectedly and for reasons unrelated to quantum computing. For proof-system architectures, the lesson is the same: modular component replacement (definition 5) rather than permanent commitment to any single cryptographic assumption.

**Hash-based security.** Hash-based constructions degrade rather than collapse under quantum attacks. Collision-critical components (e.g., Merkle commitments at L2) may need larger outputs than preimage-critical components (e.g., Fiat-Shamir challenges at L4) to achieve the same PQ security level; the sizing implications are detailed in §5.5.

**Lattice-based security and the PCS gap.** Lattice-based constructions are among the most mature PQ candidates, and for proof systems they offer a plausible foundation for replacing Shor-vulnerable algebraic groups at L2. The practical gap to hash-based and pairing-based alternatives has narrowed substantially in the past two years. Earlier lattice-based polynomial commitments reported proof sizes in the megabyte range (e.g., 8.93 MB at degree $2^{20}$ [HSS24]), but newer constructions bring proofs into the tens of kilobytes: Greyhound, composed with LaBRADOR, reports evaluation proofs on the order of $10^4$–$10^5$ bytes at degree $2^{30}$ [NS24], Hachi reports comparable sizes for lattice-based multilinear commitments over extension fields [NOZ26], and Jindo [HLSS26] achieves a further 4.8× proof-size reduction over prior lattice PCS at degree $2^{19}$ by integrating coefficient-encoding and evaluation-protocol improvements. At moderate degrees, the transparent lattice construction of Cini et al. achieves evaluation proofs competitive with hash-based FRI on proof size [CMNW24]. The absolute gap to KZG's constant-size openings (tens of bytes) remains several orders of magnitude, and known lattice-based evaluation proofs still have polynomial rather than constant size in the degree parameter, so whether that gap closes further is an open question. The earlier "MB versus KB" framing, however, no longer reflects the state of the art; lattice-based PCS is in the same performance regime as hash-based alternatives for many practical degrees, and the engineering case for lattice L2 in proof systems has strengthened accordingly.

**Code-based and symmetric primitives.** Code-based cryptography provides PQ key encapsulation: in March 2025 NIST selected HQC, a code-based KEM, as a backup to the lattice-based ML-KEM [ABC+25], while Classic McEliece advanced through the NIST process but was not selected [AAC+22]. No natural polynomial commitment or evaluation-proof construction has emerged from code-based assumptions. This family is therefore not currently a candidate for L2 replacement in proof systems. Symmetric-key primitives (block ciphers, MACs) face only Grover-type degradation [Gro96]: doubling key lengths restores pre-quantum security margins. Since proof systems use symmetric primitives primarily for hashing (Merkle trees, Fiat-Shamir), the symmetric PQ story reduces to the hash-based analysis above.

# 5 Quantum impact on proof systems

The L1–L4 decomposition and impact taxonomy from §2.4 classify how quantum adversaries affect deployed proof-system architectures. For each major system family, we provide

attack costs, reduction losses, and parameter implications.

This section mixes formal results from the literature with deployment-oriented inferences. To keep these distinct, we use three labels throughout. *Established* marks claims directly supported by cited formal results in the literature. *Derived* marks engineering consequences that follow from cited formal results under explicitly stated assumptions. *Heuristic* marks deployment-oriented guidance that extrapolates beyond current end-to-end proofs and is compiler- and implementation-dependent. Claims not explicitly labeled should be read as established where they cite formal results and as derived otherwise. Quantitative parameter recommendations are heuristic unless stated otherwise.

## 5.1 Shor-vulnerable commitments and the PCS hinge

*Evidence status: established.* Shor on discrete log and the $d$-SDH reduction to DLP are standard results; the system-level classification is derived from these.

Any system whose binding or evaluation soundness reduces to discrete-log-style hardness has its L2 commitment broken under Shor [Sho97]: security does not return by increasing curve sizes. Whether this manifests as a structural break or a modularly replaceable break depends on whether the commitment interface is separable (definitions 2 and 3).

### 5.1.1 Total collapse when verification semantics are not modular

Groth16 is an archetype: its verification semantics are intrinsically pairing equations [Gro16]. There is no separable "commitment module" with a stable verification interface that can be swapped without changing the verifier itself. In terms of the modularity test (definition 5), deployed Groth16 stacks fail the replaceable-commitment condition, so the cryptographic effect is a structural break (definition 2); for the typical upgrade-constrained on-chain deployment this combines (table 4) to total collapse.

### 5.1.2 KZG: trapdoor recovery and forgery

KZG commitments [KZG10] rely on a structured reference string (SRS) encoding powers of a trapdoor $\tau \in \mathbb{F}_p$. Formally, KZG binding reduces to the $d$-Strong Diffie–Hellman ($d$-SDH) assumption, which Shor breaks by solving discrete log. A quantum adversary recovers $\tau$ from the SRS elements $g$ and $g^\tau$, then fabricates openings consistent with arbitrary evaluation claims. Binding and evaluation soundness collapse entirely.

### 5.1.3 Concrete attack costs for KZG/pairing-based systems

The attack reduces to computing a discrete logarithm in the group used by the SRS. Table 7 summarizes resource estimates from multiple sources.

*Evidence status: derived.* Under surface-code error correction with realistic physical error rates ($\sim 10^{-3}$), the physical qubit overhead is on the order of $10^3$–$10^4\times$ the logical count, placing BN254 at order $10^6$–$10^7$ physical qubits and BLS12-381 somewhat higher [FMMC12, GE21].

The multi-target structure of the SRS (many group elements $g, g^\tau, g^{\tau^2}, \ldots$, all encoding powers of the same $\tau$) does not meaningfully reduce the attack cost: recovering $\tau$ from any single pair $(g, g^\tau)$ suffices to reconstruct all powers.[2] For universal-SRS deployments [GWC19], the consequence is immediate: one recovered trapdoor compromises every circuit instantiated under that SRS, not just a single application-specific verifier.

[2]The pairing-group structure (separate groups $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$) does not change the fundamental resource estimate, since the attack targets the base group where the SRS is defined.

**Table 7:** Quantum resource estimates for cryptographically relevant attacks. Entries are grouped into logical-qubit estimates (top block), physical-qubit or architecture-module estimates (bottom block), and are not directly comparable across blocks.

| Source | Target | Resource estimate | Notes |
|---|---|---|---|
| *Logical-qubit estimates (standard surface-code-style model)* | | | |
| Roetteler et al. [RNSL17] | EC-DLP (256-bit, BN254-class) | ∼2,330 logical qubits; $\sim 1.3 \times 10^{11}$ Toffoli ($\approx 2^{37}$) | $9n+2\lceil \log_2 n \rceil+10$ qubits ($n$-bit). |
| Roetteler et al. [RNSL17] | EC-DLP (BLS12-381) | ∼3,500 logical qubits | From the same Roetteler formula ($n$=381). |
| Babbush et al. [BZG+26] | EC-DLP (secp256k1, 256-bit) | ∼1,200–1,450 logical qubits; $\sim$ 7–9 $\times 10^7$ Toffoli | 2026 refined estimate specifically targeting 256-bit EC curves; other 256-bit curves including BN254 reported as "same order of magnitude," with BLS12-381 requiring a somewhat larger device (cf. the ∼3,500-qubit Roetteler row above). |
| *Physical-qubit / architecture-module estimates (architecture-specific)* | | | |
| Gidney–Ekerå [GE21] | RSA-2048 (anchor) | $\sim 2 \times 10^7$ physical qubits; ∼8 hours | Surface-code physical estimate with error correction. |
| Babbush et al. [BZG+26] | EC-DLP (secp256k1, 256-bit) | $< 5 \times 10^5$ physical qubits | Superconducting surface code at $10^{-3}$ physical error rate; circuit runs in minutes. |
| Litinski [Lit23] | EC private key (256-bit) | ∼6,000 active-volume modules ($\sim 7 \times 10^6$ physical qubits total); ∼50M Toffoli | Active-volume architecture with 1,152 physical qubits per module; organized differently from the standard logical-qubit model and not directly comparable to the rows above. |

Estimates depend on error-correction scheme, gate synthesis, and architectural assumptions. The qualitative conclusion (Shor is polynomial-time; parameter tuning does not help) is robust; specific numbers are indicative. The 2026 Babbush et al. refinement places 256-bit ECDLP attacks at a markedly lower logical-qubit budget than earlier surface-code extrapolations, bringing the physical-qubit requirement below $5 \times 10^5$ on superconducting architectures. Logical-qubit rows and physical-qubit rows are not directly comparable; conversion depends on the error-correction code, target error rate, and scheduling model.

#### 5.1.4 System-level classification

KZG-based stacks collapse unless (i) the commitment interface is replaceable at L2 and (ii) deployed verification logic can be upgraded. If both hold, the classification is modular collapse with an explicit migration step. If either fails, the practical outcome is total collapse.

### 5.2 Fiat-Shamir: classical soundness and the QROM

*Evidence status: established and derived.* DFMS and measure-and-reprogram are established QROM results; parameter implications below are derived or heuristic as flagged.

#### 5.2.1 Two L4 obligations: classical soundness and the QROM lift

*Evidence status: established (attacks), heuristic (deployment implication).* The non-interactivity layer carries two distinct soundness obligations, and the post-quantum question is only the second. The first is *classical* Fiat-Shamir soundness: that instantiating the

round challenges with a concrete hash, and binding every soundness-relevant value into the transcript before the challenge is derived, does not itself introduce an attack. The second is the ROM-to-QROM lift analyzed in the remainder of this subsection. The two are independent: a transform can be QROM-sound in the idealized model yet unsound once instantiated, or correctly instantiated yet lack a QROM proof.

Until recently the classical obligation was treated as discharged for natural protocols, with Fiat-Shamir instantiation counterexamples regarded as contrived. That view no longer holds. Khovratovich, Rothblum, and Soukhanov give the first Fiat-Shamir attack on a natural, widely studied protocol, the GKR interactive proof: for every concrete hash there is a circuit family on which the non-interactive GKR argument proves false statements [KRS25]. The attack is classical, not quantum, and its scope is bounded, requiring an adversarially constructed circuit that internally recomputes the hash, so deployments that verify a fixed, audited circuit are not directly broken. The companion positive direction defines a white-box attack model and proves a variant transform ("XFS") secure within it [AY25], bounding how far the attack generalizes, though it is not yet benchmarked for deployed multi-round IOP pipelines.

The deployment-grade form of the same failure class is transcript binding. A 2026 disclosure broke the soundness of six deployed zkVMs (Jolt, Nexus, Cairo-M, Ceno, Expander, Binius64) by exploiting prover-controlled public values that were not absorbed into the Fiat-Shamir transcript before the gating challenge was sampled, letting a prover see challenges in advance and prove impossible statements [Ott26]. These were classical soundness breaks at L4, not L2 commitment failures and not QROM issues, and they reinforce that the modularity test's "re-instantiable Fiat-Shamir layer" (definition 5) hides a soundness-critical correctness obligation orthogonal to the post-quantum question.

### 5.2.2 Why ROM rewinding does not automatically lift

Classical Fiat-Shamir security and extractability arguments are typically rewinding-based: the extractor rewinds the prover to extract a witness from multiple accepting transcripts. In the QROM, two obstacles arise: (i) oracle queries in superposition prevent the extractor from observing classical query strings without disturbing the adversary's state, and (ii) quantum no-cloning constrains rewinding. A ROM proof relying on classical rewinding is therefore not evidence of QROM security.

Unruh develops quantum-aware proof techniques that replace naive rewinding [Unr14, Unr17]. DFMS-style results [DFMS19], measure-and-reprogram techniques [DFM20], and Zhandry's compressed-oracle framework [Zha18] provide the modern QROM proof toolkit.

### 5.2.3 Concrete QROM reduction losses

QROM reductions incur quantifiable security losses. For the well-studied case of Fiat-Shamir on $\Sigma$-protocols, the DFMS result [DFMS19] establishes QROM security with a reduction loss quadratic in the number of quantum oracle queries $q_H$. Let $\kappa$ denote the soundness error of the underlying interactive protocol prior to Fiat-Shamir compilation, and let $|C|$ denote the challenge-space size. The compiled NIZK has soundness error on the order of $\kappa + q_H^2/|C|$ in the QROM (simplified; the full DFMS expression includes additional terms; see [DFMS19] for the precise bound), compared to $\kappa + q_H/|C|$ in the ROM. The difference is that the reduction loss scales *quadratically* in $q_H$ rather than linearly. This DFMS-style bound applies under structural conditions such as special soundness and honest-verifier zero-knowledge for the underlying $\Sigma$-protocol.

For multi-round IOP compilation (the pattern used by STARK-style systems), the situation is more involved. Measure-and-reprogram techniques [DFM20] can handle multi-round protocols, but the reduction loss depends on the number of rounds $r$ and the transcript structure. Losses polynomial in $r$ and $q_H$ are typical, making tight bounds more

**Table 8:** Indicative QROM applicability map for common Fiat-Shamir compilation patterns.

| Compilation pattern | QROM technique | Approx. loss behavior | Conditions |
|---|---|---|---|
| FS on Σ-protocols | DFMS | Typically quadratic in $q_H$ | Special soundness, honest-verifier ZK (HVZK), and a transcript structure matching the proof framework in [DFMS19]. |
| Multi-round IOP-to-non-interactive (NI) compilation | Measure-and-reprogram | Typically polynomial in $r$ and $q_H$ | Round structure and extractor/soundness properties must support the measure-and-reprogram argument; tightness is system-dependent. |
| ROM-only extractor arguments | None by default | No direct QROM guarantee | A classical ROM proof alone is not evidence of QROM security; explicit adaptation is required. |

important for systems with many rounds of interaction (as in deep FRI). For the specific case of FRI-based pipelines, Block et al. prove Fiat-Shamir security, including against quantum query adversaries in the QROM, for FRI, batched FRI, and a class of low-degree-test protocols covering ethSTARK, Plonky2, RedShift, and RISC Zero [BGK+23], via round-by-round soundness of the underlying IOP. This places the Fiat-Shamir *compilation* step for these systems on firm footing; what remains pipeline-specific is the conjectured-versus-proven *interactive* FRI soundness and the concrete per-pipeline QROM bit-loss, on which the ethSTARK assessment in §5.5 depends. Table 8 summarizes QROM applicability for common compilation patterns.

*Evidence status: derived and heuristic.* These losses have implications for parameter adjustment in deployed systems; the following observations are illustrative, not prescriptive recommendations for any specific pipeline. Hash output lengths for challenge derivation should account for the $q_H^2$ QROM loss, not just the $q_H$ ROM loss. Soundness margins in multi-round IOPs should include the round-dependent QROM overhead. To illustrate the scale: targeting 128-bit PQ security against adversaries making up to $2^{64}$ quantum queries, the QROM penalty beyond the ROM loss (the extra factor of $q_H = 2^{64}$ in $q_H^2$ versus $q_H$) accounts for on the order of 64 bits of the classical margin. The actual figure is compiler- and implementation-dependent, and a full pipeline-specific analysis is required before translating this observation into deployed parameters.

### 5.2.4 Applicability map

FS on Σ-protocols is covered by DFMS under specific conditions [DFMS19]. Multi-round IOP-to-NI compilation requires measure-and-reprogram reasoning; details depend on transcript structure [DFM20]. ROM-only extractor arguments are insufficient evidence for PQ security unless explicitly adapted to the QROM [YZ20]. An alternative compilation path is the Fischlin transform, which Majenz and Sharma [MS26] prove to be straight-line extractable in the QROM via compressed-oracle techniques, avoiding the rewinding difficulties of standard Fiat-Shamir extraction. Whether the Fischlin transform is practical for the multi-round IOP pipelines used in deployed proof systems remains to be evaluated; the transform trades extraction difficulty for larger proofs, and the concrete overhead has not been benchmarked for production compilers.

### 5.3 Extractability under quantum adversaries

*Evidence status: established and derived.* Classical no-go obstacles and the Chiesa et al. 2025 positive result are cited; per-system applicability is derived.

Many practical systems target arguments of knowledge: acceptance implies existence of an efficient extractor for a witness [BCCT11]. Quantum adversaries complicate extraction because naive rewinding interacts poorly with superposition access and no-cloning: a classical extractor rewinds the prover to an earlier internal state and re-runs it with a fresh challenge, but measuring a quantum prover's superposition irreversibly collapses it, and the no-cloning theorem prevents saving a copy of the pre-measurement state for reuse.

Recent work by Chiesa et al. (TCC 2025) studies quantum rewinding in IOP-based succinct arguments and establishes conditions under which post-quantum extraction can be obtained [CDD+25]. This partially addresses the open problem of post-quantum knowledge soundness for IOP-based systems, though the results apply under specific structural conditions (state-restoration soundness, particular round structures) that do not automatically cover every deployed compiler. For deployment, L4 must still be analyzed on a per-system basis.

### 5.4 Zero-knowledge property under quantum adversaries

*Evidence status: established and derived.* Watrous-style ZK-against-quantum results and Unruh's QROM simulation techniques are established; system-level implications are derived.

The preceding subsections (§5.1, §5.2, and §5.3) focus on soundness and knowledge-soundness. However, the *zero-knowledge property itself* can also be affected by quantum adversaries.

In the standard simulation paradigm, the ZK property requires that for every (possibly malicious) verifier, there exists an efficient simulator that produces indistinguishable transcripts without access to the witness. When the verifier (or distinguisher) is quantum, the simulator must produce transcripts that are indistinguishable to a quantum distinguisher. For honest-verifier ZK (HVZK), this is typically unproblematic: the simulator's output is a classical transcript, and computational indistinguishability against quantum distinguishers follows if the underlying hiding property (e.g., commitment hiding) holds against quantum adversaries.

For *malicious-verifier* ZK, the situation is more subtle. Classical simulation techniques that rewind the verifier encounter the same quantum obstacles as extraction: no-cloning and measurement disturbance. Building on Watrous's quantum rewinding lemma [Wat06], Unruh develops quantum-aware simulation strategies that work in the QROM [Unr14]. For most deployed non-interactive systems (where the verifier is deterministic given the proof), the simulation concern reduces to the hiding property of the commitment scheme at L2 and the zero-knowledge simulator for the underlying IOP at L3.

For hash-based systems where commitment hiding relies on hash properties, the ZK property is structurally salvageable (no Shor-class break at the primitive level) and degrades in parallel with soundness; the magnitude of that degradation remains pipeline-dependent. For pairing-based systems, the ZK property is often less consequential than soundness once a quantum break occurs. Hiding often relies on Decisional Diffie–Hellman (DDH)-type assumptions that are also Shor-vulnerable; however, once soundness collapses (allowing the adversary to forge proofs for false statements), the ZK property becomes moot for new proofs: the adversary gains nothing additional from breaking hiding when they can already fabricate convincing proofs without a valid witness. For previously generated honest proofs, hiding may still matter: an adversary who breaks DDH could extract witness information from existing proof transcripts, which is relevant in CNFL scenarios (§6.1) where retained proofs carry sensitive data. In either case, a complete PQ analysis should explicitly address

both soundness and zero-knowledge. For simulation-extractable non-interactive arguments, the quantum story is even less settled; this remains an open problem for applications that require stronger composition guarantees.

## 5.5 Hash-based IOP systems: STARK/FRI-style constructions

STARK-style systems [BBHR18] avoid trusted setup and discrete-log-based commitments by relying on hash-based Merkle commitments and FRI proximity testing [BGKS19]. At the primitive level, the dominant quantum effect is quantitative (hash security and soundness parameters), so these systems fall under quantitative degradation, subject to full pipeline-specific QROM analysis confirming that no additional structural vulnerability arises in the compiled protocol.

### 5.5.1 FRI-specific soundness under quantum adversaries

*Evidence status: established and derived.* FRI achieves low-degree testing through recursive folding: the prover repeatedly halves the polynomial's degree by sending Merkle-committed evaluations, and the verifier checks consistency at randomly sampled positions. Soundness depends on two components: the *proximity gap* (how far a non-low-degree function must be from low-degree, guaranteeing detection with bounded queries) and the collision resistance of the Merkle commitments binding the prover to its claimed evaluations.

Three effects arise under quantum adversaries. The first is collision bounds in Merkle layers. Each round of FRI involves Merkle commitment, so collision resistance is load-bearing. For $n$-bit hash outputs, quantum collision-finding query complexity is at most $\approx 2^{n/3}$ under the BHT bound (§4.1). This bound assumes quantum random-access memory (QRACM); whether QRACM is practically realizable at cryptographic scales remains debated. Without QRACM, the effective complexity may be closer to $2^{2n/5}$ or higher [CNS17], making $2^{n/3}$ a conservative (worst-case) estimate for deployment planning. The access-model distinction matters for practitioners interpreting collision-security margins: parameters sized for the $2^{n/3}$ bound may be more conservative than necessary if QRACM proves impractical. With $n = 256$, the worst-case $2^{n/3}$ bound gives $\approx 85$ bits of collision security. Achieving 128-bit PQ collision security requires $n \approx 384$. The second is proximity-gap interaction. The proximity gap analysis of FRI [BGKS19] is information-theoretic: it does not depend on computational assumptions but bounds the probability that a function far from low-degree passes the proximity test given random queries. This analysis is therefore unaffected by quantum adversaries. The quantum pressure enters through the Merkle binding (can the prover equivocate on committed evaluations?) and through the QROM compilation of the interactive protocol. The third is soundness error composition. FRI soundness error compounds across $\log d$ folding rounds (for degree-$d$ polynomials). Each round contributes soundness error from the proximity gap and from the Merkle commitment. Under quantum adversaries, the Merkle contribution per round weakens by the collision-security reduction, and the Fiat-Shamir compilation adds QROM loss per round. Conservative parameter selection should account for both effects multiplicatively across rounds.

### 5.5.2 Grinding (proof-of-work) under Grover

*Evidence status: derived.* Some STARK implementations include a grinding step: the prover must find a nonce such that a hash of the proof transcript has a specified number of leading zeros. This serves as a proof-of-work to amplify soundness. Under Grover, a $k$-bit grinding target provides only $\approx k/2$ bits of PQ security rather than the classical $k$ bits.

In practice, deployed systems typically use grinding parameters in the range of 16–20 bits (as of the 2021 ethSTARK specification [Ben21]). A 16-bit grinding target contributes

16 bits of classical soundness amplification but only $\approx 8$ bits under Grover-type adversaries; a 20-bit target contributes $\approx 10$ PQ bits rather than 20. To restore the original soundness contribution under quantum adversaries, the grinding parameter must be doubled (e.g., from 16 to 32 bits). However, doubling the grinding parameter squares the prover's expected work for the grinding step, which may be acceptable for moderately sized parameters (e.g., 32-bit grinding adds $\approx 2^{32}$ hash evaluations) but becomes a meaningful prover cost at higher targets. Implementations that rely on grinding for $\geq$20% of their total soundness budget should either double the parameter or redistribute the soundness margin to other mechanisms (e.g., additional FRI query rounds), depending on which approach has lower overall cost impact.

#### 5.5.3 ethSTARK parameter assessment

*Evidence status: heuristic engineering warning.* The analysis below is an order-of-magnitude directional argument, not a derived PQ security bound for any specific ethSTARK deployment.

ethSTARK [Ben21] targets 80-bit conjectured security in its standard configuration (as of the 2021 specification). Under PQ assumptions with 256-bit hash outputs, the generic quantum collision bound of $\approx 2^{n/3}$ query complexity [BHT98] puts Merkle collision security at approximately 85 bits, leaving roughly 5 bits of margin above the 80-bit target before further reductions are applied.

Several additional factors further erode this thin margin. First, FRI's multi-round folding structure requires the verifier's random challenges to remain binding under quantum superposition queries; the QROM composition overhead across these rounds incurs a security loss whose magnitude depends on the number of folding rounds and the assumed query bound (see §5.2.3). Second, the grinding parameter's PQ contribution is halved (as analyzed above), reducing its soundness share. The combination of narrow collision margins, cumulative multi-round QROM overhead, and grinding reduction under Grover together place effective PQ security meaningfully below the 80-bit classical target. We state this as a qualitative directional claim, not a computed number: the exact magnitude of the degradation depends on the QROM composition model applied to FRI's full pipeline, which remains an open question (§7). Without a per-pipeline derivation that accounts for all interacting reductions (proximity gap, Merkle binding, measure-and-reprogram across folding rounds, grinding contribution), any specific PQ bit-security figure for ethSTARK would be premature. The point we are making is structural: an 80-bit classical margin that relies on 256-bit Merkle outputs and 16–20-bit grinding has very little slack left once generic quantum losses are applied, so parameter choices that appeared conservative in the classical analysis may not remain conservative under quantum adversaries.

One possible engineering response, illustrative rather than prescriptive, would involve: (i) moving collision-critical Merkle commitments to larger hash outputs (e.g., 384-bit, yielding $\approx 128$ bits of quantum collision security), and (ii) adding FRI query rounds to widen the soundness margin and absorb QROM overhead. The exact parameter choices are compiler- and implementation-dependent; these figures should not be read as formal recommendations but as order-of-magnitude indicators of the adjustment scale involved. The margin pressure is compounded for small-field instantiations (e.g., Plonky2/3 over Goldilocks): Fenzi and Sanso [FS25] show that proximity-gap and list-decoding bounds in small-field hash-based SNARGs are weaker than previously conjectured, leaving less classical headroom to absorb PQ overhead. This margin pressure is not specific to small fields: Crites and Stewart disprove the up-to-capacity Reed–Solomon proximity-gap conjectures for all field sizes, showing that the achievable proximity radius is bounded by list-decoding capacity rather than the conjectured rate bound [CS25]. The result is purely information-theoretic and does not break FRI soundness, but FRI- and WHIR-style parameter sets chosen on the capacity conjecture lose some classical proximity margin

before any post-quantum overhead is applied. More query-efficient hash-based proximity tests have since been proposed, notably STIR [ACFY24a] and WHIR [ACFY24b]; both rely only on collision-resistant hashing and inherit the same quantitative-degradation profile as FRI, so a post-quantum L2 replacement for a PLONK+KZG stack can target FRI, STIR, or WHIR.

## 5.6 System-specific PQ notes

Several system families share structural similarity with previously analyzed constructions. Table 9 summarizes their PQ-specific concerns; we discuss the two families with the most distinctive issues below.

The classifications in this subsection and in §5.10 apply our taxonomy to named deployed systems and inherit a different evidentiary footing from the family-level analysis in §5.1 and §5.5. The family-level results rely on structural properties of the underlying primitives (Shor on discrete log, Grover on hash preimage, DFMS on Fiat-Shamir), which are established or derived from cited formal results. The system-specific entries instead encode a best-effort classification of specific deployed systems from public documentation of their commitment layer, transcript structure, and deployment architecture. These entries are judgments, not theorems, and may shift as systems publish updated analyses or change their implementations.

**Table 9:** System-specific PQ notes for families structurally similar to previously analyzed constructions. Entries are classification judgments under our taxonomy, not proved end states; "pending" QROM-status entries reflect the state of the literature to our knowledge and may be addressed in recent preprints we have not surveyed. Evidence status derived or heuristic unless stated otherwise.

| System | Structural similarity | PQ-specific concern | QROM status |
|---|---|---|---|
| Plonky2/3 | STARK/FRI | Aggressive parameter margins may leave insufficient room for PQ overhead [FS25]; Goldilocks field does not introduce Shor-vulnerability. | Pending, to our knowledge |
| Circle STARKs | STARK/FRI | Circle-group QROM transfer: transcript structure is similar to standard FRI, but formal verification for the circle-group variant appears open. | Pending, to our knowledge |
| Halo 2 | Bulletproofs (DLP) | Accumulation propagates Shor-vulnerability through recursion chain; no PQ replacement for IPA's homomorphic interface. | N/A (structural break) |
| Binius | STARK (hash-based) | Binary-tower QROM compilation: multilinear extensions and different evaluation/proximity arguments require independent QROM verification. | Pending, to our knowledge |

### 5.6.1 Halo 2: accumulation compounds Shor-vulnerability

*Evidence status: established.* The Halo 2 system, building on Halo [BGH19], uses an IPA over a prime-order elliptic-curve group and achieves recursive proof composition through an accumulation scheme. Despite being transparent, its commitment binding relies entirely on discrete-log hardness: the Pedersen-style vector commitments are Shor-vulnerable. The accumulation scheme compounds the problem: accumulated verification claims carry forward discrete-log-dependent commitments, so a Shor-capable adversary

can forge accumulated proofs at any point in the recursion chain. Replacing IPA with a PQ-plausible commitment would require redesigning the accumulation scheme's algebraic interface, since Halo 2's recursion is tightly coupled to IPA's linear homomorphism and efficient commitment updates. This accumulation pattern shares the same fundamental concern as folding-based recursion schemes such as Nova (§5.8).

### 5.6.2 Binius: binary-tower QROM verification

*Evidence status: derived and heuristic.* Binius [DP23] operates over binary tower fields with hash-based Merkle commitments, placing it in the quantitative-degradation category. However, its polynomial IOP structure differs from standard FRI (it uses multilinear extensions and different evaluation/proximity arguments). Its polynomial commitment has since evolved from the original Brakedown-style scheme to a binary-tower FRI variant (FRI-Binius) [DP24], so the object requiring independent QROM analysis is now this FRI-based commitment rather than the original code-based one; the quantitative-degradation classification is unchanged. The QROM compilation pattern must be verified independently: measure-and-reprogram results do not automatically transfer across different IOP structures without checking that structural conditions (e.g., state-restoration soundness) are met. The quantitative-degradation classification is consistent with available evidence but carries the caveat that Binius-specific QROM analysis has not been carried out to our knowledge.

## 5.7 Discrete-log transparent systems: Bulletproofs and IPA

*Evidence status: established and derived.* Bulletproofs [BBB+17] and IPA-style constructions rely on discrete-log-based binding commitments. Shor implies a structural break. The structural coupling is tighter than in PLONK+KZG: the inner-product argument's verification performs recursive halving with multi-exponentiations that are integral to the verification equation, not opaque calls to a replaceable PCS interface. The verifier checks $g^a h^b = C$ (schematically) for Pedersen-type commitments at each recursion step; these group operations are fused into verification semantics, paralleling Groth16's pairing-equation coupling. No known PQ construction preserves the same homomorphic vector-commitment interface at comparable efficiency, so deployed stacks are best classified as structural-break systems at the family level; as with other families, per-deployment outcomes depend on verifier governance and upgrade architecture.

## 5.8 Folding schemes and recursion: Nova as a post-quantum hinge

*Evidence status: established and derived.* Folding schemes such as Nova [KST21] build recursion by repeatedly folding constraint instances. In common instantiations, Nova uses group-based commitments inside the folding step, inheriting discrete-log assumptions at L2.

Recursion is not a Shor escape hatch: it embeds an inner verifier as arithmetic constraints in an outer system. If the inner verifier relies on Shor-vulnerable group operations, recursion *arithmetizes* those operations rather than eliminating them. To illustrate: in a Nova folding verifier circuit, the commitment-update step involves multi-scalar multiplication (MSM) over the base group. These group operations become arithmetic constraints in the outer circuit. A Shor-capable adversary who can produce group elements satisfying the MSM relation without knowing the corresponding discrete logs can satisfy these constraints with fabricated values that do not correspond to a valid inner proof, breaking folding soundness. By contrast, recursion for hash-based IOP systems stays within the "hash + field arithmetic" envelope, so the dominant effect remains quantitative.

## 5.9 Lookup arguments as L3 components

*Evidence status: derived.* The reduction to host-system PQ status is a structural consequence of the layered analysis, not a separate formal result.

Lookup arguments (plookup [GW20], LogUp [Hab22], Lasso/Jolt [STW23, AST23]) rarely introduce a new PQ fault line by themselves. Their soundness typically reduces to L2 (the lookup-table commitments use the same PCS as the rest of the system) and L4 (Fiat-Shamir compilation of the lookup protocol). The lookup-specific algebraic checks (permutation arguments, logarithmic derivatives) are L3 operations that do not introduce additional hardness assumptions beyond those already present in the host system. Consequently, the PQ classification of the host proof system carries through to its lookup components: if the host's L2 is Shor-vulnerable, so are the lookup commitments; if the host is hash-based, lookups inherit the same quantitative degradation profile.

## 5.10 Consolidated system map

Table 10 classifies the architectural properties and PQ impact for each system family. Table 11 covers deployed systems with measured or well-established numbers; table 12 covers preliminary and research-stage constructions. The two tables are cross-referenced by system name.

Systems classified as "quantitative degradation" face PQ pressure (hash parameter increases, QROM reduction overhead) but avoid a Shor-style structural break. Systems classified as "structural break" require architectural replacement, not parameter tuning. "Modular collapse" indicates that a PCS swap plus verifier upgrade constitutes a viable migration path if the deployment passes the modularity test (definition 5). All FS-based systems carry an L4 QROM obligation regardless of their L2 classification.

**Table 10:** System map (Part A): architectural classification and PQ impact. The cryptographic-effect and migration columns record classification judgments under the two-axis taxonomy of §2.4; they are not proved end states and individual entries may shift as the literature or deployment architectures evolve.

| System | Setup | L2 mechanism | Uses FS? | Crypto. effect | Migration |
|---|---|---|---|---|---|
| Groth16 | CRS | Pairing verification | No | Structural break | Upgrade-constrained |
| PLONK+KZG | Univ. SRS | KZG PCS | Yes | Modular collapse | Deployment-dep.§ |
| Halo 2 | Transparent | IPA (DL-based) | Yes | Structural break | Upgrade-constrained |
| Bulletproofs | Transparent | IPA / DL commits | Yes | Structural break | Upgrade-constrained |
| STARK-style | Transparent | Merkle + FRI | Yes | Quant. degrad. | Upgrade-ready¶ |
| Plonky2/3 | Transparent | FRI (small fields) | Yes | Quant. degrad. | Upgrade-ready¶ |
| Circle STARK | Transparent | Circle-FRI + Merkle | Yes | Quant. degrad.† | Upgrade-ready¶ |
| Binius | Transparent | Hash-based (binary) | Yes | Quant. degrad.† | Upgrade-ready¶ |
| Nova | Transparent | Group commits in fold | Yes | Structural break‡ | Upgrade-constrained |
| Lattice PCS | Varies | Lattice assumptions | Yes | PQ-plausible | Maturity-limited |

† QROM verification of the deployed pipeline is pending; classification is conditional on this analysis.

‡ Nova's folding step uses group-based commitments inheriting DLP assumptions. The structural break propagates through the recursion chain (§5.8).

§ PLONK+KZG passes the modularity test architecturally (the PCS is separable), so the migration outcome depends on the specific deployment's verifier upgradeability: upgrade-ready where proxy governance exists, upgrade-constrained otherwise.

¶ "Upgrade-ready" at the family level reflects an architectural tendency: the proof system admits PCS replacement without changing statement semantics. Whether a specific deployment is upgrade-ready depends on its verifier governance and contract architecture; the modularity test of definition 5 must be applied per deployment.

**Table 11:** System map (Part B.1): deployed systems with measured or well-established performance numbers. Ethereum gas figures and published asymptotic bounds come from specifications and production deployments. Entries are order-of-magnitude indicators for architectural comparison and remain configuration-, circuit-, and implementation-dependent. Numbers in this table should not be compared directly against the preliminary and research-stage entries in table 12.

| System | Recursive comp.? | Proof size | Verif. cost | Notes / source |
|---|---|---|---|---|
| Groth16 | No | ≈128–192 bytes | ∼230k gas | 3 group elements [Gro16]; gas via EIP-197/1108 [BR17, Car18] |
| PLONK+KZG | Optional | ≈0.8–2 KB | ≲300k gas | Ethereum gas, implementation-dependent; dominated by EIP-197/1108 pairing precompile [BR17, Car18] |
| Halo 2 | Yes | ≈5–10 KB | $O(n)$ group ops | Asymptotic [BGH19]; deployed in Zcash Orchard [HBHW25, Zca21] |
| Bulletproofs | Limited | ≈0.7 KB | $O(n)$ group ops | Asymptotic [BBB+17] |
| STARK-style | Possible | ≈50–200 KB | ∼1–5M gas | Ethereum gas [Tha22]; varies by batching/blobs |
| Plonky2/3 | Yes | ≈40–100 KB | Recursion-opt. | Typically verified via outer proof |

*Units.* "gas" = Ethereum gas units; "group ops" = elliptic-curve operations (asymptotic). Proof sizes in bytes (B) or kilobytes (KB).
*Abbreviations.* "Recursion-opt." = verification cost is optimized via recursive proof composition and not directly comparable to single-proof costs.

**Table 12:** System map (Part B.2): preliminary and research-stage estimates. These entries are not measured production numbers. Circle STARK, Binius, and Nova figures come from early implementations and design papers and are subject to revision as implementations mature. The Lattice PCS row reflects research-stage constructions with no production deployment. *These numbers must not be compared directly against the deployed-systems entries in table 11*; doing so mixes measured and estimated quantities and will produce misleading impressions of relative efficiency.

| System | Recursive comp.? | Proof size | Verif. cost | Maturity / source |
|---|---|---|---|---|
| Circle STARK | Possible | ≈50–150 KB | Similar to STARK | Preliminary; based on published estimates |
| Binius | Emerging | ≈30–100 KB | Binary-opt. | Preliminary; implementation-dependent |
| Nova | Yes (central) | Folding-dep. | Recursion-dep. | Preliminary; performance inherits from inner scheme |
| Lattice PCS | Unclear | KB–MB range | Heavy | Research-stage; proof size is construction-dependent (MB-scale in [HSS24]; tens of KB in [NS24, NOZ26]) |

*Abbreviations.* "Binary-opt." = verification optimized for binary-tower arithmetic; "Folding-dep." = cost depends on the inner scheme used in folding.

# 6 Present-day consequences

Architectural choices made today determine whether a deployed proof system can migrate later, and whether future quantum capabilities enable forgery via broken assumptions in live verification logic.

## 6.1 "Collect now, forge later"

"Harvest now, decrypt later" (HNDL) refers to collecting ciphertext today and decrypting it once public-key encryption breaks [CSD17]. For proof-system deployments, the analogous risk is future proof forgery. The core threat is not that an adversary must secretly harvest artifacts; verification keys, on-chain verifier code, and proof transcripts are typically public and persistent on-chain. Rather, the risk is that vulnerable verification infrastructure remains in place: as long as a Shor-vulnerable verifier continues to accept proofs, a future quantum-capable adversary can generate fraudulent proofs that pass legacy verification.

The analogy with HNDL is imperfect and the difference matters for risk assessment: HNDL threatens confidentiality of past data (collected ciphertext can be decrypted retroactively), while "collect now, forge later" (CNFL) threatens future integrity (new fraudulent proofs can be generated that pass legacy verification).

> *CNFL threatens future integrity (verification infrastructure), not confidentiality of previously proven statements.*

In pairing-based zk-SNARK deployments, CNFL specializes into two operational risks. *Forward forgery* is the risk that an adversary with access to a quantum computer generates new proofs for false statements and has them accepted by a legacy (Shor-vulnerable) verifier still in operation. *Retroactive trust erosion* is the risk that long-retained attestations lose evidentiary weight once forged proofs become indistinguishable from genuine ones under the legacy verifier: the historical proofs themselves are not invalidated, but their probative value degrades once the verification infrastructure can no longer distinguish honest from forged proofs. These are consequence concepts that only become operational once CNFL is introduced; the framework definitions in §2.4 (structural break, modularly replaceable break, quantitative degradation) remain the root taxonomy.

For blockchains specifically, a future attacker cannot retroactively rewrite finalized history without breaking consensus. The operational risk is continued acceptance: if the protocol keeps accepting proofs under a Shor-vulnerable verifier, a quantum-capable adversary can create new fraudulent transitions that verify and become canonical state. This argument scopes the attack to the proof-system layer and assumes the underlying consensus mechanism (signatures, VRFs, randomness, and block-production primitives) remains secure against quantum adversaries. If the chain's own consensus primitives are Shor-vulnerable (as most production blockchains' signature schemes currently are), the attack surface extends beyond proof forgery to direct chain-level compromise, which requires its own migration track orthogonal to proof-system verifier upgrades; we treat that case as out of scope here.

## 6.2 Implications for blockchains, rollups, and on-chain verifiers

Many systems choose pairing-based SNARKs because proofs are small and verification is cheap [LHW+25]. This efficiency comes with a structural dependency on Shor-vulnerable verification semantics.

Several migration barriers attach to this dependency. On-chain verification logic is a protocol surface: verifiers can be hard-coded in contracts or consensus rules; proving/verification keys are coupled to circuit and protocol versions; bridges, light clients, and indexers depend on stable formats; and long-lived state commitments must remain interpretable across upgrades.

Verification cost provides a concrete anchor for comparing stacks. For pairing-based verifiers on Ethereum, the pairing-check precompile (EIP-197, repriced in EIP-1108) costs $45{,}000 + 34{,}000 \cdot k$ gas for $k$ pairing inputs [BR17, Car18]. Groth16 verification typically costs ∼230k gas (dominated by the EIP-197/1108 pairing precompile [BR17, Car18], plus

calldata and public-input overhead); PLONK-family verifiers $\lesssim$300k gas (implementation-dependent); deployed STARK verification $\sim$1–5M gas depending on batching, calldata, and blob usage [Tha22]. Ethereum's fee structure has changed since EIP-4844 [BFL+22], whose blob transactions reduce calldata costs for data-heavy proofs; this narrows the on-chain data-cost component for hash-heavy STARK-style proofs. Blob economics and intrinsic verifier execution cost should be distinguished: EIP-4844 changes data-availability pricing, not the base cost of executing pairing checks or other verification logic inside the Ethereum Virtual Machine (EVM).

## 6.3 Case studies: applying the modularity test

We now apply the PQ modularity test (definition 5) to three real deployment patterns to illustrate whether the framework produces actionable guidance.

The conclusions in each case study split into two kinds. *Architectural conclusions* follow from the cryptographic structure of the proof system and the taxonomy of §2.4: they identify which layer carries the Shor-vulnerable assumption, whether the commitment interface is separable, and what migration category the system falls into. These conclusions are durable under ordinary deployment changes. *Operational conclusions* depend on the current state of a specific deployment's verifier contract, governance mechanism, and ecosystem coordination. These may change with contract upgrades, governance decisions, or protocol forks, and should be read as a snapshot of early 2026 rather than as stable findings. We flag which conclusions are durable and which are time-sensitive inline as they arise.

Each case study follows the same evaluation checklist:

Q1: What is the exact L2 mechanism, and which assumption is load-bearing for binding/evaluation soundness?

Q2: Is the FS layer explicit, and what is its oracle-model status (ROM-only, QROM-proven, or QROM-pending)?

Q3: Does verifier upgradeability exist at the protocol level, contract level, or governance level?

Q4: Are historical commitments/proofs still interpretable after a verifier upgrade?

Q5: What remains unproved for a full deployed-QROM claim?

The modularity test conditions (1)–(3) from definition 5 are then evaluated, the taxonomy classification is derived, and the practical migration outlook is stated.

### 6.3.1 Case study 1: Zcash Sapling (Groth16)

**Evidence basis (as of early 2026).** Zcash Protocol Specification [HBHW25] and ZIP-224 for Orchard [Zca21] (normative specifications; both are living documents whose content may evolve across protocol versions), plus architectural inference from the cryptographic structure of Groth16 over BLS12-381.

Zcash Sapling uses Groth16 proofs over BLS12-381 for shielded transactions.

Q1 (L2 mechanism): Pairing-based verification; binding and evaluation soundness rely on discrete-log and pairing assumptions in BLS12-381, broken by Shor.

Q2 (FS / oracle model): N/A; Groth16 achieves non-interactivity through algebraic NIZK structure, not Fiat-Shamir.

Q3 (Verifier upgradeability): Consensus-layer embedded. Upgrading requires a network-wide protocol change (hard fork), coordinated across all full nodes.

Q4 (Historical interpretability): Shielded notes and nullifiers would need re-proving or explicit legacy-trust policy after migration.

Q5 (QROM residual): N/A (no FS layer).

**Modularity test.**

1. Replaceable commitment interface? *No.* Groth16's verification semantics are intrinsically pairing equations. There is no separable PCS module.
2. Replaceable Fiat-Shamir layer? *N/A.*
3. Upgradable verification logic? *Upgrade-constrained.* Zcash can change consensus rules through network upgrades, but this requires hard-fork-level coordination.

**Verdict.** Fails condition (1). Applying definition 2: the soundness of Groth16 verification relies on pairing-group assumptions (broken by Shor in quantum polynomial time), and these assumptions are not cleanly replaceable without changing the system's verification interface. Classification: *total collapse.* As long as the Sapling verifier remains in consensus rules, the system is exposed to forward forgery and retroactive trust erosion (see §6.1). Migration requires deploying a successor proof system (e.g., a STARK-based shielded pool) and transitioning users and state through a carefully coordinated network upgrade. The Zcash Orchard upgrade [HBHW25, Zca21] (moving to Halo 2, building on Halo [BGH19]) addresses trusted-setup concerns but remains Shor-vulnerable due to IPA's DLP dependence. The Zcash community prioritized eliminating trusted setup (a concrete, present-day trust concern) over PQ resilience, which addresses a future and currently non-operational threat. This means that a subsequent PQ migration will require another full proof-system replacement, since Halo 2 is no closer to PQ resilience than Groth16. The Sapling and Orchard shielded pools coexist on-chain: the Sapling Groth16 verifier remains part of the active consensus rules even after Orchard's deployment, so the Shor-vulnerable verification path remains live at the protocol level and accepts Sapling-format transactions whenever they are submitted. We make no claim here about the current volume of Sapling versus Orchard traffic, which varies over time and is outside the scope of this architectural analysis; the point is that the verification path cannot be retired simply by deploying a successor pool. For deployments making architectural choices today, PQ resilience and setup-model preferences should be evaluated together rather than sequentially.

### 6.3.2 Case study 2: zkSync Era (composite STARK+SNARK rollup)

**Evidence basis.** The zkSync Era Boojum upgrade announcement [ZKs23] (project blog) and community analyses of the deployed verifier contracts, supplemented by third-party deployment tracking.[3] Architectural claims about the inner-FRI / outer-wrapper composition are engineering-level inference from these sources, not direct consequences of a primary technical specification.

We illustrate the composite-wrapper pathway using zkSync Era [ZKs23], a ZK-rollup on Ethereum whose proving architecture changed with the Boojum upgrade (July 2023). Prior to Boojum, zkSync Era used a PLONK-style proof system with KZG polynomial commitments. Post-Boojum, the inner proving system is FRI-based, operating over a Goldilocks-field arithmetization in a STARK-style configuration [ZKs23]. Based on public zkSync documentation, the Boojum upgrade announcement, and community analyses of the deployed verifier contracts, the inner STARK proof is, to our knowledge, compressed by an outer pairing-based SNARK wrapper before on-chain verification, because raw STARK proofs are too large for cost-effective L1 verification; we flag the architectural claim as engineering-level inference from public sources rather than as a direct consequence of a primary technical specification. The on-chain verifier is deployed behind an upgradeable proxy contract governed by a multisig structure (as of early 2026; evidence status: heuristic,

[3] L2BEAT (https://l2beat.com/scaling/projects/zksync-era) provides contract addresses and risk snapshots but is not a primary technical source.

as governance configurations depend on organizational decisions rather than formal proofs).[4]

This composite architecture creates a split PQ profile: the inner proof system (L1 arithmetization through L2 commitment) is hash-based and Shor-resistant, while the outer compression wrapper reintroduces a pairing-based Shor-vulnerable component at the on-chain verification boundary (see §6.6).

Q1 (L2 mechanism): Composite. The inner PCS is FRI-based (hash commitments, no DLP dependence); the outer compression wrapper uses a pairing-based SNARK. Inner binding relies on collision resistance (quantitative degradation under Grover). Outer wrapper binding relies on pairing assumptions broken by Shor, but the wrapper is separable, so this is a modularly replaceable break rather than a structural one.

Q2 (FS / oracle model): Both the inner STARK and the outer SNARK use Fiat-Shamir compilation. ROM proofs exist for generic STARK pipelines; end-to-end QROM verification of the full composite pipeline (inner FRI + outer wrapper) is pending.

Q3 (Verifier upgradeability): The verifier contract is behind a proxy with multisig governance, providing an explicit upgrade path. However, governance coordination and security-council approval introduce latency.

Q4 (Historical interpretability): State roots are Keccak-based commitments to account state, not intrinsically tied to either the inner or outer PCS. A wrapper replacement would change the on-chain proof format but can preserve state-root continuity.

Q5 (QROM residual): End-to-end QROM proof for the composite pipeline (inner FRI prover + outer pairing-based compressor + Fiat-Shamir transcript) is not available. The inner and outer components would each require separate QROM analysis.

**Modularity test.**

1. Replaceable commitment interface? *Yes, demonstrated in practice.* The Boojum upgrade itself replaced the original KZG-based PCS with FRI, confirming that the PCS layer is architecturally separable. The remaining Shor-vulnerable component is the outer compression wrapper, which is a distinct module from the inner commitment scheme.

2. Replaceable Fiat-Shamir layer? *Yes.* Transcript hashing and challenge derivation are configurable parameters in both the inner and outer proof systems.

3. Upgradable verification logic? *Yes, via proxy governance.* zkSync Era's on-chain verifier sits behind an upgradeable proxy controlled by a security council multisig. This provides a concrete upgrade mechanism, though governance latency and coordination remain practical constraints.

**Verdict.** Passes all three conditions, with the Boojum upgrade providing empirical evidence for condition (1). The PQ profile is composite rather than uniform. The inner proof system (FRI over Goldilocks) falls under definition 4: Grover-type speedups reduce concrete hash security, but parameter increases restore it without structural redesign. The outer pairing-based compression wrapper, however, is Shor-vulnerable and falls under definition 3: it must be replaced, not merely re-parameterized. Because the wrapper is architecturally separated from the inner prover and the verifier contract is proxy-upgradeable, this replacement is feasible through the same governance mechanism that executed the Boojum migration. Until the wrapper is replaced, the on-chain verification layer remains exposed to forward forgery. Classification: *quantitative degradation* (inner system) composed with *modular collapse* (outer wrapper), with a demonstrated migration path. The practical migration involves deploying a PQ-compatible compression layer (e.g., recursive STARK folding or a PQ-SNARK wrapper), updating the verifier contract via

[4]Specific governance parameters (multisig threshold, timelock duration, proxy type) are configuration-dependent and subject to change; see the zkSync Era documentation for current values.

governance, running a dual-verifier epoch window (§6.6), and sunsetting the pairing-based wrapper.

### 6.3.3 Case study 3: StarkNet (STARK-based)

**Evidence basis.** StarkWare SHARP [Sta24] and Starknet's published decentralization roadmap [Sta22] (project docs and governance blog). Current governance parameters are tracked at `https://governance.starknet.io`; our claims describe the documented framework as of early 2026 rather than the exact state of deployed production contracts.

StarkNet [Sta24, Sta22] uses STARK proofs with hash-based Merkle commitments and FRI proximity testing.

Q1 (L2 mechanism): Hash-based Merkle commitments + FRI proximity testing. No Shor-vulnerable assumptions.

Q2 (FS / oracle model): Fiat-Shamir compilation; ROM-based arguments exist, end-to-end QROM verification of the full deployed pipeline is pending.

Q3 (Verifier upgradeability): StarkNet's on-chain verifier sits under a documented upgrade framework and a governance-controlled upgrade path. Starknet's published decentralization roadmap [Sta22] set the direction toward community governance and token-weighted voting; the current framework comprises the Starknet Foundation's upgrade process, Starknet Improvement Proposals (SNIPs) with STRK token-weighted voting, and a Security Council for emergency response (established via SNIP-25 in late 2024). We report this as a description of the documented framework rather than as a direct inspection of current production contracts, which may have been updated since the cited sources (evidence status: heuristic, as governance structures depend on organizational decisions).[5]

Q4 (Historical interpretability): State roots are hash-based and not tied to a specific curve; parameter changes preserve interpretability.

Q5 (QROM residual): Full QROM proof for the multi-round FRI pipeline as deployed has not been formally completed.

**Modularity test.**

1. Replaceable commitment interface? *Yes, though tightly optimized.* The commitment layer is hash-based (Merkle + FRI). No structural Shor-vulnerability exists. Hash function and output size are configurable parameters. Migration means parameter tightening, not component replacement.
2. Replaceable Fiat-Shamir layer? *Yes.* Hash selection and transcript conventions can be updated between versions.
3. Upgradable verification logic? *Yes, per the documented upgrade framework.* Upgrades to StarkNet's on-chain verifier proceed through SNIP-based governance (STRK token voting) or, in emergencies, through the multi-sig Security Council (as of early 2026). Within this documented framework, verifier replacement does not require redeploying dependent contracts (as of early 2026).

**Verdict.** Passes all three conditions. Applying definition 4: quantum attacks reduce concrete security via Grover-type speedups on hash-based commitments and QROM reduction losses on Fiat-Shamir compilation, but security can be restored through parameter increases without replacing the core commitment or verification interface. Classification: *quantitative degradation* with upgrade-ready migration feasibility. PQ hardening is expected to involve increasing hash output sizes for collision-critical paths, widening soundness margins to absorb QROM overhead, and formally verifying the QROM compilation of the

[5] Governance specifics as of early 2026; the Starknet governance framework continues to evolve. Consult `https://governance.starknet.io` for current parameters.

deployed pipeline. The exact parameter changes are implementation-dependent (see §5.5). The cost is proportional increases in proof size and verification gas. Even for systems that are mathematically upgrade-ready, governance-based upgrade mechanisms (token-weighted voting, security council multisigs) introduce social consensus delays and coordination friction that extend the effective migration timeline beyond the purely technical changeover window.

## 6.4 Migration timeline framing

The modularity test determines whether migration is feasible; the question of when requires a separate timeline analysis. Migration difficulty and urgency together drive actual decisions.

Let $T_{\text{CRQC}}$ denote the year a cryptographically relevant quantum computer becomes operational, and let $T_{\text{mig}}$ denote the engineering time required to complete a PQ migration (design, implement, audit, deploy, and sunset legacy acceptance). The critical constraint is:

*Systems with structural breaks must begin migration by* $T_{CRQC} - T_{mig}$.

For representative values: if $T_{\text{mig}} \approx$ 3–5 years (an author estimate encompassing specification, implementation, auditing, and coordinated deployment, consistent with the multi-year migration timelines in NIST IR 8547 [MPR+24]), then structural-break systems need to initiate migration 3–5 years before CRQC arrival. Systems with quantitative degradation face a softer deadline, since parameter adjustments can be deployed more incrementally.

The uncertainty in $T_{\text{CRQC}}$ itself is large. Policy anchors across jurisdictions (UK NCSC 2025, NIST IR 8547 initial public draft, EU Commission Recommendation 2024/1101) converge on a common 2030–2035 migration window (see §4.2 for the full discussion), while the most aggressive recent resource estimate places 256-bit ECDLP attacks within reach of 1,200–1,450 logical qubits and under $5 \times 10^5$ physical qubits (at a $10^{-3}$ physical error rate) [BZG+26]. Neither the policy window nor the technical estimate settles the specific year; together they bound the relevant planning horizon. The cost of late migration (operating with broken verification semantics) exceeds the cost of early migration (engineering effort spent before CRQC arrives). For systems whose migration difficulty is "High" (table 13), the conservative strategy is to treat migration as a current design priority regardless of specific hardware forecasts.

## 6.5 Migration surfaces: what actually changes

Even if the high-level application remains the same, proof-system migration usually touches: verifier semantics (pairing equations vs. hash/IOP checks), artifacts and trust model (circuit-specific CRS vs. transparent setup), proof format and transcript rules (domain separation, serialization), and state binding (how proofs bind to roots/commitments across upgrade epochs).

## 6.6 Hybrid migration patterns

A practical migration pattern is *hybrid verification*: use a PQ-plausible successor system to enforce transition rules while legacy proofs are phased out.

Two patterns recur. A *dual-verifier window* accepts both legacy and successor proofs for a defined epoch range, with explicit versioning. A *wrapper (outer proof) pattern* has a successor proof system prove that "the legacy verifier would accept this legacy proof", and the chain then verifies only the successor proof after cutover.

As a concrete rollup-style example, a rollup using Groth16 for withdrawals can: (i) deploy a successor verifier (e.g., STARK-style) alongside the legacy verifier; (ii) accept both for $N$ epochs, binding each to the same state root; (iii) after the window, accept only successor proofs; (iv) keep the legacy verifier callable only for historical auditing (or remove it).

Some deployed rollup architectures already use wrapper-style compression in practice. For example, a STARK-style proof may be compressed by an outer pairing-based SNARK for on-chain efficiency. This improves verification cost, but if the outer wrapper remains Shor-vulnerable, the resulting stack is still not PQ-plausible in the relevant sense.

Wrapping does not make the legacy proof system PQ-plausible. It only changes what the chain accepts going forward. If the outer wrapper itself is pairing-based or otherwise Shor-vulnerable, the wrapped system remains non-post-quantum in the relevant sense; the wrapper buys compatibility or efficiency, not PQ security.

## 6.7 Migration playbook

Table 13 summarizes representative migration directions and pain points.

**Table 13:** Migration playbook: typical direction, difficulty, and dominant pain point.

| Current | Migration direction | Difficulty | Pain point |
|---|---|---|---|
| Groth16 | Replace verifier; deploy successor, version formats, sunset legacy via governance. | High | Protocol upgrade + re-attestation; losing tiny proofs. |
| PLONK+KZG | Swap PCS (e.g., to FRI); upgrade verifiers; dual-verifier window. | Med–High | Bigger proofs; QROM obligations. |
| Halo 2 | No PQ-plausible IPA replacement; redesign proof primitive. | High | Replacing homomorphism-heavy interfaces. |
| Bulletproofs | Same as Halo 2; no drop-in PQ analogue for IPA. | High | Same as Halo 2. |
| STARK-style | Tighten parameters (digest length, query counts, margins); document QROM. | Low–Med | Proof size + cost increase. |
| Nova / folding | Redesign folding commitments to avoid DLP. | High | Recursion amplifies Shor-vulnerability. |

## 6.8 Stakeholder-specific recommendations

**Protocol designers.** Design for modularity at L2 and upgradeability at L4. For new rollups choosing between Groth16 and PLONKish stacks, PLONKish offers cleaner PCS separation, making PCS swap a feasible migration plan.

**Library maintainers.** For arkworks, halo2, Plonky2/3, and similar libraries, make commitment and transcript APIs explicit and versioned, and require version identifiers and domain separators in Fiat-Shamir hashing, so that migrating from 256-bit to 384-bit digests does not create ambiguous transcripts.

**Rollup and on-chain deployers.** Treat verifier replacement as a governance feature, not an emergency patch. The concrete pattern is to deploy successor verification alongside legacy, run a dual-verifier epoch window, then sunset legacy acceptance.

**Auditors.** Trace the L2 assumption chain and record oracle-model status as a standing audit item, including an explicit line for "FS compilation proven in ROM only" versus "QROM proof applies under specific conditions."

**Governance and compliance owners.** Define how long-term evidence is interpreted across verifier versions: "valid under verifier v1 at issuance" versus "must remain valid under current verifier" are materially different policies.

**Regulators and standards bodies.** PQ migration guidance for proof-system deployments should follow the broader PQ transition windows documented in §4.2 (UK NCSC, NIST IR 8547 ipd, EU Commission Recommendation 2024/1101 and the 2025 coordinated roadmap). Proof-system verifiers should be treated as a distinct asset class from classical signature/KEM deployments: a jurisdiction's PQ migration inventory should enumerate on-chain and off-chain proof-system verifiers separately, since their migration paths (dual-verifier epochs, wrapper replacement, parameter tightening) differ structurally from certificate-authority rotation or PKI reissuance.

# 7 Open problems and research directions

The layered analysis surfaces concrete open problems, concentrated at L2 (where Shor-vulnerable algebra is embedded) and L4 (where ROM-to-QROM transfer and tight bounds remain compiler-dependent). The problems below fall into two categories. *Theoretical* problems require new cryptographic constructions or proofs: designing PQ-plausible folding schemes, closing the lattice-PCS proof-size gap, establishing stronger zero-knowledge guarantees under quantum adversaries. *Engineering and audit* problems require applying or specializing existing QROM toolkits to deployed pipelines that differ from the idealized models: pipeline-faithful QROM proofs for ethSTARK, system-specific verification of Binius and Plonky2/3. We mark each subsection with its category and with a priority tag: three subsections (*end-to-end QROM security*, *post-quantum folding*, and *practical PQ commitments*) are flagged as Priority 1 because they gate the paper's most direct classifications. The remaining subsections are important but derivative. A scope-boundary statement closes the section in §7.7. For each problem, we indicate which taxonomy outcome it affects and why resolution matters for deployment decisions.

## 7.1 End-to-end QROM security for deployed pipelines

*Priority 1; engineering / audit.*

Many deployed STARK-style systems rely on Fiat-Shamir, and their security arguments combine proximity testing, Merkle commitments, and extraction reasoning. Establishing end-to-end QROM claims for the exact pipeline used in practice, not just idealized components in isolation, remains a gap. The QROM proof toolkit [BDF+10, Zha18, DFMS19] provides the building blocks, but composing these results across multi-round FRI folding, Merkle commitment layers, and pipeline-specific transcript conventions has not been carried out for any concrete deployed system.

Three concrete subproblems stand out. The first is to construct *pipeline-faithful models* that formalize engineering choices (transcript structure, batching, query scheduling, domain separation, hash instantiation) and prove the compiled scheme in the QROM. The gap between idealized IOP models and deployed compilers is nontrivial: batching strategies, recursive aggregation steps, and domain-separation conventions all affect the transcript structure that QROM proofs must cover. The second is to close the *extractor realism* gap: QROM extractors may be non-tight or assume conditions that differ from deployed compilers. The Chiesa et al. result on quantum rewinding for IOP-based arguments [CDD+25]

partially addresses this by establishing conditions under which post-quantum extraction succeeds, but applicability depends on whether the deployed compiler satisfies the structural conditions (e.g., state-restoration soundness) assumed by the result. Characterizing precisely which deployed pipelines meet these conditions is a near-term open problem. The third is to bound *concrete soundness margins* by translating FRI soundness error composition into conservative PQ margins, accounting for both quantum collision bounds and QROM reduction losses across $\log d$ folding rounds (§5.5). The multi-round structure of FRI [BGKS19, GMW25] means that per-round security losses accumulate, and QROM overhead at each round interacts with the proximity-gap analysis in ways that have not been bounded end-to-end.

Recent work narrows this gap from the theory side. Chiesa et al. prove that a hash-based succinct non-interactive argument obtained via the BCS transformation is QROM-secure whenever the underlying IOP satisfies classical round-by-round soundness [CDHZ25], composing the Merkle-commitment and Fiat-Shamir layers in a single modular argument. This is a general framework rather than a per-pipeline result: it reduces the remaining work to verifying that each deployed FRI-based pipeline, including circle-STARK and binary-tower variants, satisfies round-by-round soundness, which is the pipeline-faithful modeling problem above.

Resolution would confirm or refine the "quantitative degradation" classification for hash-based systems and provide concrete parameter guidance for PQ hardening.

## 7.2 Tight QROM bounds for common proof-system compilers

*Priority 2; theoretical and engineering.*

Even when QROM techniques exist, security losses can be non-trivial (§5.2.3). The DFMS result [DFMS19] establishes QROM security for Fiat-Shamir on Σ-protocols with a quadratic loss in $q_H$, and Don et al. [DFM20] extend this to multi-round protocols via measure-and-reprogram, showing that the $O(q_H^2)$ loss is essentially tight for general protocols (via a Grover-based attack matching the bound). Yamakawa and Zhandry [YZ20] provide lifting theorems from ROM to QROM, but these apply under structural conditions that not all deployed compilers satisfy.

A practical research direction is tighter, construction-relevant bounds translated into concrete parameter guidance. Two needs recur. The first is to treat *tightness as an engineering input*: a principled mapping from QROM loss terms to parameter changes, rather than ad hoc safety factors. Current practice amounts to adding conservative slack (e.g., 64 additional bits for $2^{64}$ quantum queries), but the relationship between loss tightness and actual parameter overhead is not well understood for multi-round compiled systems. The second is to develop *compiler-specific checklists*: reusable criteria for when a given compilation pattern is defensible in the QROM, reducing audit friction. For Unruh's framework [Unr17], applicable conditions include special soundness and honest-verifier ZK, but deployed compilers use varied transcript structures that do not always map cleanly onto these conditions.

Tighter bounds directly affect whether "quantitative degradation" systems require modest or substantial parameter increases, with corresponding proof-size and verification-cost implications.

## 7.3 Post-quantum folding schemes and recursion

*Priority 1; theoretical.*

Folding-based IVC (e.g., Nova [KST21], SuperNova [KS22]) is a widely used recursion pattern, but common instantiations inherit DLP assumptions through their commitment mechanisms (§5.8). Designing folding schemes whose commitment mechanisms are PQ-plausible without losing recursion efficiency remains open, though recent work has begun

to address it: LatticeFold [BC24] (preprint, 2024) introduces the first lattice-based folding scheme, building on Module-SIS and claiming performance comparable to group-based schemes, but the construction is recent and deployment competitiveness remains undemonstrated. Follow-up work has narrowed the gap: LatticeFold+ [BC25] (CRYPTO 2025) removes the bit-decomposition range proof that bottlenecked the original prover, and Neo and SuperNeo [NS25, NS26] extend lattice-based folding to CCS over small fields such as Goldilocks, the regime used by Plonky2/3, while retaining post-quantum (Module-SIS) commitments.

Three subdirections matter here. The first is building PQ-plausible commitments that preserve the folding interface. Nova's folding step relies on the linear homomorphism of Pedersen-style vector commitments; lattice-based commitments offer additive homomorphism but with substantially larger parameters, and it is unclear whether the folding interface can be preserved at competitive concrete costs. The second is recursion within a hash and field envelope. Fractal [COS19] demonstrates PQ-plausible recursion via holographic IOPs with hash-based commitments, but its concrete performance (proof size, prover time) has not reached deployment competitiveness. The third is competitive concrete performance: any PQ folding scheme must be benchmarked against the DLP-based schemes it would replace. The current cost differential (group operations vs. hash-and-field arithmetic for folding updates) is large enough that a pure efficiency argument may not suffice; amortization strategies or new algebraic structures may be needed.

Folding-based systems currently face "structural break" under Shor. A PQ-plausible folding scheme would reclassify them as "quantitative degradation" or "PQ-plausible."

## 7.4 Practical PQ commitment and PCS designs

*Priority 1; theoretical and engineering.*

Replacing Shor-vulnerable PCS at L2 is the primary migration bottleneck. The proof-size gap between lattice-based and pairing-based PCS remains large but is narrowing: Hwang et al. [HSS24] report 8.93 MB at degree $2^{20}$ for their lattice PCS, compared to constant-size (≈48-byte) openings for KZG [KZG10]. More recent constructions such as Jindo [HLSS26] (preprint, 2026) achieve 4.8× smaller proofs than prior work at comparable degrees, alongside the Greyhound and Hachi results discussed in §4.2. Ishai et al. [ISW21] construct lattice-based designated-verifier zkSNARKs, but these target a different setting (designated verifier) and do not directly address the publicly-verifiable PCS interface needed by most deployed systems.

Three problems block "PCS swap" paths. Closing the proof-size gap asks whether MB-scale lattice PCS proofs can be compressed to KB-scale through new techniques, amortization, or fundamentally different algebraic approaches. Developing multi-opening and batching interfaces is critical for deployed systems that batch many polynomial evaluations into a single proof; lattice-based analogues of these interfaces are underdeveloped. Making lattice PCS recursion-friendly requires in-circuit verification costs that make PQ recursion practical. Lattice arithmetic in-circuit is expensive, and current lattice PCS verification does not appear amenable to efficient arithmetization.

Progress here directly enables the "modular collapse → PCS swap" migration path for PLONK-family systems and reduces the practical barrier to PQ migration.

## 7.5 System-specific QROM verification

*Priority 2; audit and engineering.*

Our analysis identifies several systems where the PQ classification is "quantitative degradation" but formal QROM verification of the deployed pipeline has not been completed. Binius [DP23] uses a binary-tower polynomial IOP compilation whose multilinear extension structure and different evaluation and proximity arguments require independent

QROM verification (§5.6.2). Circle STARKs [HLP24, CGH+26] are a circle-group FRI variant whose transcript structure is similar to standard FRI but whose group-theoretic underpinning differs (§5.6). Plonky2/3 uses aggressive parameter margins over small fields; whether the chosen soundness margins provide adequate room for PQ overhead is an open quantitative question (§5.6). Recent analysis by Fenzi and Sanso [FS25] (2025 preprint) suggests that small-field hash-based SNARGs may be less sound than conjectured, which raises further concern about thin margins in this regime; the analysis is recent and further independent verification would strengthen the conclusions. For ethSTARK, the parameter assessment in §5.5 identifies thin PQ margins, but a formal per-pipeline QROM derivation accounting for all interacting reductions has not been completed [Ben21].

Each is a concrete verification target whose resolution would either confirm or necessitate revision of the current "quantitative degradation" classification.

## 7.6 Stronger zero-knowledge guarantees under quantum adversaries

*Priority 2; theoretical.*

Our discussion in §5.4 focuses on the baseline zero-knowledge property, but many applications require stronger guarantees such as simulation-extractability, robust composability, or security under repeated composition with auxiliary protocols. For these stronger notions, the quantum picture is less settled.

Unruh [Unr17] establishes that Fiat-Shamir compilation yields simulation-sound proofs in the QROM, but the result does not extend to simulation-extractability (where the simulator must also extract witnesses from adversarial proofs). Very recent work by Majenz and Sharma [MS26] (2026 preprint) proves QROM security of the Fischlin transform (an alternative to Fiat-Shamir with straight-line extraction). The result is recent and its applicability to deployed NIZK compilers has not been systematically evaluated, and the gap between simulation-soundness and simulation-extractability remains. Watrous [Wat06] shows that interactive proofs can achieve ZK against quantum verifiers via quantum rewinding, but the non-interactive setting and composition with other quantum-aware protocols remain less understood.

Three concrete open questions follow. Identifying simulation-extractable NIZKs in quantum-aware models requires determining which existing constructions retain simulation-extractability against quantum adversaries and under what assumptions. The gap between simulation-soundness (established) and simulation-extractability (open) is particularly relevant for applications like anonymous credentials and privacy-preserving smart contracts. Understanding composition with deployed protocol stacks means characterizing how quantum-aware ZK guarantees behave when proofs are embedded in larger systems such as bridges, recursive proof pipelines, or multi-stage rollups. Determining practical proof obligations means deciding what additional arguments auditors and protocol designers should require when applications need more than baseline soundness and zero-knowledge.

Stronger composition guarantees are orthogonal to the L2 structural/quantitative distinction but affect whether a "PQ-plausible" classification suffices for a given application's security requirements.

## 7.7 Limitations of this work

The analysis is structural and architecture-driven. It identifies where quantum risk enters a proof system, what kind of migration is implied by the L1–L4 decomposition, and which proof burdens remain open for the deployed pipelines we examine. Several caveats bound the scope of the conclusions.

**Scope exclusions.** The paper focuses on IOP/PCS-based and algebraic proof-system families that dominate current succinct proof deployments. MPC-in-the-head constructions (e.g., KKW, Ligero, Limbo) and purely symmetric-key-based ZK approaches are excluded (§1.4): these are arguably the most naturally post-quantum constructions, but a systematic PQ analysis would require a different structural decomposition and is left for future work. We also do not treat "quantum ZK" constructions that use quantum communication or quantum verifiers, and we do not make hardware timeline predictions beyond citing the resource-estimate literature.

**Deployment-specificity of quantitative claims.** Parameter recommendations (hash output lengths, grinding targets, QROM slack bits) are compiler- and implementation-dependent. The paper does not derive pipeline-specific numeric bounds for any deployed system; the ethSTARK thin-margin assessment in §5.5 is explicitly framed as a qualitative directional claim rather than a computed security level. A full formal derivation that accounts for all interacting reductions (proximity gap, Merkle binding, multi-round measure-and-reprogram, grinding) for a specific deployed pipeline is an open problem (§7).

**Evidence basis for the case studies.** The three case studies in §6.3 apply the modularity test to real deployments, but the supporting evidence is heterogeneous. Zcash Sapling rests on normative protocol specifications. zkSync Era's inner-FRI / outer-pairing-wrapper composition is engineering-level inference from public documentation and community analyses rather than a direct consequence of a primary technical specification. StarkNet governance claims describe the documented framework rather than an inspection of current production contract state. We have flagged each of these inline; readers should treat the case-study conclusions as architectural rather than contract-level assertions.

**Snapshot nature of the classifications.** The system-specific PQ notes (table 9) and the system map (tables 10 to 12) reflect the state of the literature and deployed systems as of early 2026. New preprints may establish QROM security for systems currently marked as "pending"; protocol upgrades may change verifier contract upgradeability; and new proof systems may emerge that require reclassification. The framework (L1–L4 decomposition, two-axis taxonomy, modularity test) is intended to be durable; individual entries are not.

**QROM verification is not certification.** A "quantitative degradation" classification does not certify that a concrete deployed compiler has a complete QROM argument or that its parameter choices achieve a specified PQ security level. It indicates that the system's structural properties make such an argument plausible and such parameter choices tractable; the actual verification remains the responsibility of the system's maintainers and auditors.

# 8 Conclusion

We developed a four-layer decomposition (L1–L4) and a two-axis taxonomy for analyzing the post-quantum security of deployed zero-knowledge proof systems. Post-quantum migration is an architectural problem: parameter tuning does not recover security when the underlying assumption collapses.

The L1–L4 framework decomposes each proof system into arithmetization (L1), polynomial commitment (L2), protocol logic (L3), and non-interactive compilation (L4). This decomposition localizes quantum risk: L2 carries the dominant post-quantum exposure for most systems, while L4 introduces a near-universal QROM proof obligation that remains underserved by the current literature.

Shor's algorithm induces a structural break for pairing- and discrete-log-based commitments at L2, including KZG, IPA (Halo 2, Bulletproofs), and group-based folding (Nova). Recent resource estimates place the attack cost at 1,200–1,450 logical qubits and under half a million physical qubits on superconducting architectures for 256-bit elliptic-curve targets [BZG+26]; under these estimates, hardware at this scale could emerge within the design lifetime of systems being deployed today (table 7), though its actual arrival date is not predicted here. Hash-dominated transparent systems (STARKs, Plonky2/3, circle STARKs, Binius) face quantitative degradation: Grover-type speedups and QROM reduction losses require parameter increases but preserve architectural viability. The ethSTARK parameter assessment (§5.5) suggests that deployed 80-bit configurations may fall materially below their classical security targets without parameter adjustment, though the exact magnitude depends on the QROM composition model applied to the full pipeline.

The modularity test (definition 5), applied to Zcash Sapling, zkSync Era, and StarkNet, demonstrates that practical outcomes depend as much on deployment governance as on cryptographic properties. Systems that embed Shor-vulnerable verification semantics should assume an eventual structural break unless they have an explicit upgrade path; even hash-based pipelines carry QROM proof obligations at L4. The migration timeline framing (§6.4) shows that systems with structural breaks and multi-year migration timelines need to begin PQ migration now.

For practitioners, the analysis yields concrete architectural guidance. New deployments should design for L2 modularity from the outset: separating the polynomial commitment interface from the verification semantics so that a PCS swap does not require a full system redesign. All Fiat-Shamir-based systems carry an L4 QROM proof obligation that should be treated as a first-class audit item, not deferred indefinitely. Verifier upgrade mechanisms (proxy contracts, governance-gated replacement) should be built as standard infrastructure rather than emergency patches. Policy bodies across jurisdictions (UK NCSC 2025, NIST IR 8547 initial public draft, and the 2025 EU NIS Cooperation Group roadmap [Nat25, MPR+24, NIS25]) converge on a 2030–2035 migration window as the institutional reference point for planning; systems whose migration difficulty is high should begin PQ preparation within this window regardless of specific hardware forecasts. Proof-system verifier infrastructure should be treated as a distinct asset class in national post-quantum migration plans: the upgrade pathways for on-chain or embedded verifiers differ fundamentally from standard PKI certificate rotations, and the governance and coordination requirements documented in the case studies (§6.3) have no direct analogue in conventional cryptographic migration.

The unexpected collapse of SIDH/SIKE to a classical attack in 2022 [CD23] illustrates that no single cryptographic assumption should be treated as permanently secure. The modularity test developed here applies beyond the quantum setting; it captures whether a system's architecture can absorb *any* future assumption failure at a specific layer without requiring a full redesign.

Several limitations bound this analysis. The paper does not cover MPC-in-the-Head constructions, which face a different set of quantum considerations. The QROM security estimates rely on known reduction techniques and may shift as tighter bounds emerge (see §7 for the specific open problems). Hardware timelines are deliberately excluded: our classifications describe what happens if a cryptographically relevant quantum computer arrives, not when.

The principal open problems (§7) are pipeline-faithful QROM analyses for deployed compilers, practical PQ commitment replacements at L2, and recursion-friendly designs that do not depend on Shor-vulnerable group structure. The gap between hash-based systems that degrade gracefully and algebraic systems that suffer structural breaks determines which deployments need a full proof-system replacement and which need only parameter increases. Whether the field converges on lattice-based commitments, symmetric-key

techniques, or hybrid architectures remains open, but the layered analysis presented here provides a framework for evaluating each path.

For protocol designers, the post-quantum design question has two parts: which cryptographic assumption class a system relies on, and whether the live verification boundary (the code path that accepts proofs into canonical state) can be replaced when that assumption eventually fails. Assumption choice determines which systems suffer which failure mode; replaceability of the verification boundary determines which of them can survive the transition.

# References

[AAA+25] Rajeev Acharya, Dmitry A. Abanin, Laleh Aghababaie-Beni, Igor Aleiner, Trond I. Andersen, Markus Ansmann, Frank Arute, Kunal Arya, Abraham Asfaw, Nikita Astrakhantsev, Juan Atalaya, Ryan Babbush, Dave Bacon, Brian Ballard, Joseph C. Bardin, Johannes Bausch, Andreas Bengtsson, Alexander Bilmes, Sam Blackwell, Sergio Boixo, Gina Bortoli, Alexandre Bourassa, Jenna Bovaird, Leon Brill, Michael Broughton, David A. Browne, Brett Buchea, Bob B. Buckley, David A. Buell, Tim Burger, Brian Burkett, Nicholas Bushnell, Anthony Cabrera, Juan Campero, Hung-Shen Chang, Yu Chen, Zijun Chen, Ben Chiaro, Desmond Chik, Charina Chou, Jahan Claes, Agnetta Y. Cleland, Josh Cogan, Roberto Collins, Paul Conner, William Courtney, Alexander L. Crook, Ben Curtin, Sayan Das, Alex Davies, Laura De Lorenzo, Dripto M. Debroy, Sean Demura, Michel Devoret, Agustin Di Paolo, Paul Donohoe, Ilya Drozdov, Andrew Dunsworth, Clint Earle, Thomas Edlich, Alec Eickbusch, Aviv Moshe Elbag, Mahmoud Elzouka, Catherine Erickson, Lara Faoro, Edward Farhi, Vinicius S. Ferreira, Leslie Flores Burgos, Ebrahim Forati, Austin G. Fowler, Brooks Foxen, Suhas Ganjam, Gonzalo Garcia, Robert Gasca, Élie Genois, William Giang, Craig Gidney, Dar Gilboa, Raja Gosula, Alejandro Grajales Dau, Dietrich Graumann, Alex Greene, Jonathan A. Gross, Steve Habegger, John Hall, Michael C. Hamilton, Monica Hansen, Matthew P. Harrigan, Sean D. Harrington, Francisco J. H. Heras, Stephen Heslin, Paula Heu, Oscar Higgott, Gordon Hill, Jeremy Hilton, George Holland, Sabrina Hong, Hsin-Yuan Huang, Ashley Huff, William J. Huggins, Lev B. Ioffe, Sergei V. Isakov, Justin Iveland, Evan Jeffrey, Zhang Jiang, Cody Jones, Stephen Jordan, Chaitali Joshi, Pavol Juhas, Dvir Kafri, Hui Kang, Amir H. Karamlou, Kostyantyn Kechedzhi, Julian Kelly, Trupti Khaire, Tanuj Khattar, Mostafa Khezri, Seon Kim, Paul V. Klimov, Andrey R. Klots, Bryce Kobrin, Pushmeet Kohli, Alexander N. Korotkov, Fedor Kostritsa, Robin Kothari, Borislav Kozlovskii, John Mark Kreikebaum, Vladislav D. Kurilovich, Nathan Lacroix, David Landhuis, Tiano Lange-Dei, Brandon W. Langley, Pavel Laptev, Kim-Ming Lau, Loïck Le Guevel, Justin Ledford, Joonho Lee, Kenny Lee, Yuri D. Lensky, Shannon Leon, Brian J. Lester, Wing Yan Li, Yin Li, Alexander T. Lill, Wayne Liu, William P. Livingston, Aditya Locharla, Erik Lucero, Daniel Lundahl, Aaron Lunt, Sid Madhuk, Fionn D. Malone, Ashley Maloney, Salvatore Mandrà, James Manyika, Leigh S. Martin, Orion Martin, Steven Martin, Cameron Maxfield, Jarrod R. McClean, Matt McEwen, Seneca Meeks, Anthony Megrant, Xiao Mi, Kevin C. Miao, Amanda Mieszala, Reza Molavi, Sebastian Molina, Shirin Montazeri, Alexis Morvan, Ramis Movassagh, Wojciech Mruczkiewicz, Ofer Naaman, Matthew Neeley, Charles Neill, Ani Nersisyan, Hartmut Neven, Michael Newman, Jiun How Ng, Anthony Nguyen, Murray Nguyen, Chia-Hung Ni, Murphy Yuezhen Niu, Thomas E. O'Brien, William D. Oliver, Alex Opremcak, Kristoffer Ottosson, Andre Petukhov, Alex

Pizzuto, John Platt, Rebecca Potter, Orion Pritchard, Leonid P. Pryadko, Chris Quintana, Ganesh Ramachandran, Matthew J. Reagor, John Redding, David M. Rhodes, Gabrielle Roberts, Eliott Rosenberg, Emma Rosenfeld, Pedram Roushan, Nicholas C. Rubin, Negar Saei, Daniel Sank, Kannan Sankaragomathi, Kevin J. Satzinger, Henry F. Schurkus, Christopher Schuster, Andrew W. Senior, Michael J. Shearn, Aaron Shorter, Noah Shutty, Vladimir Shvarts, Shraddha Singh, Volodymyr Sivak, Jindra Skruzny, Spencer Small, Vadim Smelyanskiy, W. Clarke Smith, Rolando D. Somma, Sofia Springer, George Sterling, Doug Strain, Jordan Suchard, Aaron Szasz, Alex Sztein, Douglas Thor, Alfredo Torres, M. Mert Torunbalci, Abeer Vaishnav, Justin Vargas, Sergey Vdovichev, Guifre Vidal, Benjamin Villalonga, Catherine Vollgraff Heidweiller, Steven Waltman, Shannon X. Wang, Brayden Ware, Kate Weber, Travis Weidel, Theodore White, Kristi Wong, Bryan W. K. Woo, Cheng Xing, Z. Jamie Yao, Ping Yeh, Bicheng Ying, Juhwan Yoo, Noureldin Yosri, Grayson Young, Adam Zalcman, Yaxing Zhang, Ningfeng Zhu, Nicholas Zobrist, and Google Quantum AI and Collaborators. Quantum error correction below the surface code threshold. *Nature*, 638(8052):920–926, February 2025. URL: https://www.nature.com/articles/s41586-024-08449-y, doi:10.1038/s41586-024-08449-y.

[AAC+22] Gorjan Alagic, Daniel Apon, David Cooper, Quynh Dang, Thinh Dang, John Kelsey, Jacob Lichtinger, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, Daniel Smith-Tone, and Yi-Kai Liu. Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process. Technical Report NIST Internal or Interagency Report (NISTIR) 8413, National Institute of Standards and Technology, September 2022. URL: https://csrc.nist.gov/pubs/ir/8413/upd1/final, doi:10.6028/NIST.IR.8413-upd1.

[ABC+25] Gorjan Alagic, Maxime Bros, Pierre Ciadoux, David Cooper, Quynh Dang, Thinh Dang, John Kelsey, Jacob Lichtinger, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, Hamilton Silberg, Daniel Smith-Tone, and Noah Waller. Status report on the fourth round of the NIST post-quantum cryptography standardization process. Technical Report NIST IR 8545, National Institute of Standards and Technology (U.S.), Gaithersburg, MD, March 2025. URL: https://nvlpubs.nist.gov/nistpubs/ir/2025/NIST.IR.8545.pdf, doi:10.6028/NIST.IR.8545.

[ACFY24a] Gal Arnon, Alessandro Chiesa, Giacomo Fenzi, and Eylon Yogev. STIR: Reed–Solomon Proximity Testing with Fewer Queries, 2024. URL: https://eprint.iacr.org/2024/390.

[ACFY24b] Gal Arnon, Alessandro Chiesa, Giacomo Fenzi, and Eylon Yogev. WHIR: Reed–Solomon Proximity Testing with Super-Fast Verification, 2024. URL: https://eprint.iacr.org/2024/1586.

[AHIV22] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight Sublinear Arguments Without a Trusted Setup. In *CCS 2017*, 2022. URL: https://eprint.iacr.org/2022/1608, doi:10.1145/3133956.3134104.

[AST23] Arasu Arun, Srinath Setty, and Justin Thaler. Jolt: SNARKs for Virtual Machines via Lookups. In *EUROCRYPT 2024*, 2023. URL: https://eprint.iacr.org/2023/1217, doi:10.1007/978-3-031-58751-1_1.

[AY25] Gal Arnon and Eylon Yogev. Towards a White-Box Secure Fiat-Shamir Transformation, 2025. URL: https://eprint.iacr.org/2025/329.

[BBB+17] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short Proofs for Confidential Transactions and More. In *IEEE S&P 2018*, 2017. URL: https://eprint.iacr.org/2017/1066, doi:10.1109/SP.2018.00020.

[BBD09] Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen, editors. *Post-Quantum Cryptography*. Springer, Berlin, Heidelberg, 2009. URL: http://link.springer.com/10.1007/978-3-540-88702-7, doi:10.1007/978-3-540-88702-7.

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable, transparent, and post-quantum secure computational integrity, 2018. URL: https://eprint.iacr.org/2018/046.

[BBHT98] Michel Boyer, Gilles Brassard, Peter Hoeyer, and Alain Tapp. Tight bounds on quantum searching. *Fortschritte der Physik*, 46(4-5):493–505, June 1998. URL: http://arxiv.org/abs/quant-ph/9605034, arXiv:quant-ph/9605034, doi:10.1002/(SICI)1521-3978(199806)46:4/5<493::AID-PROP493>3.0.CO;2-P.

[BC24] Dan Boneh and Binyi Chen. LatticeFold: A Lattice-based Folding Scheme and its Applications to Succinct Proof Systems, 2024. URL: https://eprint.iacr.org/2024/257.

[BC25] Dan Boneh and Binyi Chen. LatticeFold+: Faster, Simpler, Shorter Lattice-Based Folding for Succinct Proof Systems, 2025. URL: https://eprint.iacr.org/2025/247.

[BCCT11] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. From Extractable Collision Resistance to Succinct Non-Interactive Arguments of Knowledge, and Back Again. In *ITCS 2012*, 2011. URL: https://eprint.iacr.org/2011/443, doi:10.1145/2090236.2090263.

[BDF+10] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random Oracles in a Quantum World. In *ASIACRYPT 2011*, 2010. URL: https://eprint.iacr.org/2010/428, doi:10.1007/978-3-642-25385-0_3.

[Ben21] Eli Ben-Sasson. ethSTARK Documentation, 2021. URL: https://eprint.iacr.org/2021/582.

[BFL+22] Vitalik Buterin, Dankrad Feist, Diederik Loekrakker, George Kadianakis, Matt Garnett, Mofi Taiwo, and Ansgar Dietrichs. EIP-4844: Shard Blob Transactions, 2022. URL: https://eips.ethereum.org/EIPS/eip-4844.

[BGH19] Sean Bowe, Jack Grigg, and Daira Hopwood. Recursive Proof Composition without a Trusted Setup, 2019. URL: https://eprint.iacr.org/2019/1021.

[BGK+23] Alexander R. Block, Albert Garreta, Jonathan Katz, Justin Thaler, Pratyush Ranjan Tiwari, and Michał Zając. Fiat-Shamir Security of FRI and Related SNARKs, 2023. URL: https://eprint.iacr.org/2023/1071.

[BGKS19] Eli Ben-Sasson, Lior Goldberg, Swastik Kopparty, and Shubhangi Saraf. DEEP-FRI: Sampling Outside the Box Improves Soundness. In *ITCS 2020*, 2019. URL: https://eprint.iacr.org/2019/336, doi:10.4230/LIPIcs.ITCS.2020.5.

[BHT98] Gilles Brassard, Peter Hoyer, and Alain Tapp. Quantum Algorithm for the Collision Problem. In *LATIN '98: Theoretical Informatics*, volume 1380, pages 163–169. Springer, 1998. URL: http://arxiv.org/abs/quant-ph/9705002, arXiv:quant-ph/9705002, doi:10.1007/BFb0054319.

[BR17] Vitalik Buterin and Christian Reitwiessner. EIP-197: Precompiled contracts for optimal ate pairing check on the elliptic curve alt_bn128, 2017. URL: https://eips.ethereum.org/EIPS/eip-197.

[BS23] Dan Boneh and Victor Shoup. *A Graduate Course in Applied Cryptography*. Self-Published, January 2023. URL: https://toc.cryptobook.us/.

[BZG+26] Ryan Babbush, Adam Zalcman, Craig Gidney, Michael Broughton, Tanuj Khattar, Hartmut Neven, Thiago Bergamaschi, Justin Drake, and Dan Boneh. Securing Elliptic Curve Cryptocurrencies against Quantum Vulnerabilities: Resource Estimates and Mitigations. Technical report, Google Quantum AI, 2026. URL: https://quantumai.google/static/site-assets/downloads/cryptocurrency-whitepaper.pdf.

[Car18] Antonio Salazar Cardozo. EIP-1108: Reduce alt_bn128 precompile gas costs, 2018. URL: https://eips.ethereum.org/EIPS/eip-1108.

[CD23] Wouter Castryck and Thomas Decru. An Efficient Key Recovery Attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023*, volume 14008, pages 423–447. Springer Nature Switzerland, Cham, 2023. URL: https://link.springer.com/10.1007/978-3-031-30589-4_15, doi:10.1007/978-3-031-30589-4_15.

[CDD+25] Alessandro Chiesa, Marcel Dall'Agnol, Zijing Di, Ziyi Guan, and Nicholas Spooner. Quantum Rewinding for IOP-Based Succinct Arguments. In *TCC 2025*, 2025. URL: https://eprint.iacr.org/2025/947, doi:10.1007/978-3-032-12296-4_16.

[CDHZ25] Alessandro Chiesa, Zijing Di, Zihan Hu, and Yuxi Zheng. How to Prove Post-Quantum Security for Succinct Non-Interactive Reductions, 2025. URL: https://eprint.iacr.org/2025/2166.

[CGH+26] Dan Carmon, Lior Goldberg, Ulrich Haböck, Leonardo Lerer, Ilya Lesokhin, Shahar Papini, and Shahar Samocha. S-two Whitepaper, 2026. URL: https://eprint.iacr.org/2026/532.

[CMNW24] Valerio Cini, Giulio Malavolta, Ngoc Khanh Nguyen, and Hoeteck Wee. Polynomial Commitments from Lattices: Post-Quantum Security, Fast Verification and Transparent Setup. In *CRYPTO 2024*, 2024. URL: https://eprint.iacr.org/2024/281, doi:10.1007/978-3-031-68403-6_7.

[CNS17] André Chailloux, María Naya-Plasencia, and André Schrottenloher. An Efficient Quantum Collision Search Algorithm and Implications on Symmetric Cryptography. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017*, volume 10625, pages 211–240. Springer International Publishing, Cham, 2017. URL: https://link.springer.com/10.1007/978-3-319-70697-9_8, doi:10.1007/978-3-319-70697-9_8.

[COS19] Alessandro Chiesa, Dev Ojha, and Nicholas Spooner. Fractal: Post-Quantum and Transparent Recursive Proofs from Holography. In *EUROCRYPT 2020*, 2019. URL: https://eprint.iacr.org/2019/1076, doi:10.1007/978-3-030-45721-1_27.

[CS25] Elizabeth Crites and Alistair Stewart. On Reed–Solomon Proximity Gaps Conjectures, 2025. URL: https://eprint.iacr.org/2025/2046.

[CSD17] Information Technology Laboratory Computer Security Division. Post-Quantum Cryptography | CSRC | CSRC, January 2017. URL: https://csrc.nist.gov/projects/post-quantum-cryptography.

[DFM20] Jelle Don, Serge Fehr, and Christian Majenz. The Measure-and-Reprogram Technique 2.0: Multi-Round Fiat-Shamir and More. In *CRYPTO 2020*, 2020. URL: https://eprint.iacr.org/2020/282, doi:10.1007/978-3-030-56877-1_21.

[DFMS19] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the Fiat-Shamir Transformation in the Quantum Random-Oracle Model. In *CRYPTO 2019*, 2019. URL: https://eprint.iacr.org/2019/190, doi:10.1007/978-3-030-26951-7_13.

[DP23] Benjamin E. Diamond and Jim Posen. Succinct Arguments over Towers of Binary Fields. In *EUROCRYPT 2025*, 2023. URL: https://eprint.iacr.org/2023/1784, doi:10.1007/978-3-031-91134-7_4.

[DP24] Benjamin E. Diamond and Jim Posen. Polylogarithmic Proofs for Multilinears over Binary Towers, 2024. URL: https://eprint.iacr.org/2024/504.

[Fed21] Federal Office for Information Security (BSI). Migration to Post Quantum Cryptography. Technical report, Federal Office for Information Security (BSI), May 2021. URL: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Crypto/Migration_to_Post_Quantum_Cryptography.pdf?__blob=publicationFile&v=2.

[FMMC12] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. Surface codes: Towards practical large-scale quantum computation. *Physical Review A*, 86(3):032324, September 2012. URL: http://arxiv.org/abs/1208.0928, arXiv:1208.0928, doi:10.1103/PhysRevA.86.032324.

[FS87] Amos Fiat and Adi Shamir. How To Prove Yourself: Practical Solutions to Identification and Signature Problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology — CRYPTO' 86*, pages 186–194, Berlin, Heidelberg, 1987. Springer. doi:10.1007/3-540-47721-7_12.

[FS25] Giacomo Fenzi and Antonio Sanso. Small-field hash-based SNARGs are less sound than conjectured, 2025. URL: https://eprint.iacr.org/2025/2197.

[GE21] Craig Gidney and Martin Ekerå. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. *Quantum*, 5:433, April 2021. URL: http://arxiv.org/abs/1905.09749, arXiv:1905.09749, doi:10.22331/q-2021-04-15-433.

[Gid25] Craig Gidney. How to factor 2048 bit RSA integers with less than a million noisy qubits, May 2025. URL: http://arxiv.org/abs/2505.15917, arXiv:2505.15917, doi:10.48550/arXiv.2505.15917.

[GMW25] Albert Garreta, Nicolas Mohnblatt, and Benedikt Wagner. A Simplified Round-by-round Soundness Proof of FRI, 2025. URL: https://eprint.iacr.org/2025/1993.

[GOT21] Cyprien Delpech de Saint Guilhem, Emmanuela Orsini, and Titouan Tanguy. Limbo: Efficient Zero-knowledge MPCitH-based Arguments. In *CCS 2021*, 2021. URL: https://eprint.iacr.org/2021/215, doi:10.1145/3460120.3484595.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC 1996)*. arXiv, 1996. URL: http://arxiv.org/abs/quant-ph/9605043, arXiv:quant-ph/9605043, doi:10.1145/237814.237866.

[Gro16] Jens Groth. On the Size of Pairing-based Non-interactive Arguments. In *EUROCRYPT 2016*, 2016. URL: https://eprint.iacr.org/2016/260, doi:10.1007/978-3-662-49896-5_11.

[GW20] Ariel Gabizon and Zachary J. Williamson. Plookup: A simplified polynomial protocol for lookup tables, 2020. URL: https://eprint.iacr.org/2020/315.

[GWC19] Ariel Gabizon, Zachary J. Williamson, and Oana Ciobotaru. PLONK: Permutations over Lagrange-bases for Oecumenical Noninteractive arguments of Knowledge, 2019. URL: https://eprint.iacr.org/2019/953.

[Hab22] Ulrich Haböck. Multivariate lookups based on logarithmic derivatives, 2022. URL: https://eprint.iacr.org/2022/1530.

[HBHW25] Daira-Emma Hopwood, Sean Bowe, Taylor Hornby, and Nathan Wilcox. Zcash Protocol Specification, Version 2025.6.3 [NU6.1]. Technical report, Electric Coin Company / Zcash Foundation, June 2025. URL: https://zips.z.cash/protocol/protocol.pdf.

[HLP24] Ulrich Haböck, David Levit, and Shahar Papini. Circle STARKs, 2024. URL: https://eprint.iacr.org/2024/278.

[HLSS26] Intak Hwang, Hyeonbum Lee, Jinyeong Seo, and Yongsoo Song. Jindo: Practical Lattice-Based Polynomial Commitment for Zero-Knowledge Arguments, 2026. URL: https://eprint.iacr.org/2026/044.

[HSS24] Intak Hwang, Jinyeong Seo, and Yongsoo Song. Concretely Efficient Lattice-based Polynomial Commitment from Standard Assumptions. In *CRYPTO 2024*, 2024. URL: https://eprint.iacr.org/2024/306, doi:10.1007/978-3-031-68403-6_13.

[ISW21] Yuval Ishai, Hang Su, and David J. Wu. Shorter and Faster Post-Quantum Designated-Verifier zkSNARKs from Lattices. In *CCS 2021*, 2021. URL: https://eprint.iacr.org/2021/977, doi:10.1145/3460120.3484572.

[KKW18] Jonathan Katz, Vladimir Kolesnikov, and Xiao Wang. Improved Non-Interactive Zero Knowledge with Applications to Post-Quantum Signatures. In *CCS 2018*, 2018. URL: https://eprint.iacr.org/2018/475, doi:10.1145/3243734.3243805.

[KRS25] Dmitry Khovratovich, Ron D. Rothblum, and Lev Soukhanov. How to Prove False Statements: Practical Attacks on Fiat-Shamir. In *Advances in Cryptology – CRYPTO 2025*, Lecture Notes in Computer Science. Springer, 2025. URL: https://eprint.iacr.org/2025/118, doi:10.1007/978-3-032-01887-8_1.

[KS22] Abhiram Kothapalli and Srinath Setty. SuperNova: Proving universal machine executions without universal circuits, 2022. URL: https://eprint.iacr.org/2022/1758.

[KST21] Abhiram Kothapalli, Srinath Setty, and Ioanna Tzialla. Nova: Recursive Zero-Knowledge Arguments from Folding Schemes. In *CRYPTO 2022*, 2021. URL: https://eprint.iacr.org/2021/370, doi:10.1007/978-3-031-15985-5_13.

[KZG10] Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-Size Commitments to Polynomials and Their Applications. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, and Masayuki Abe, editors, *Advances in Cryptology - ASIACRYPT 2010*, volume 6477, pages 177–194. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. URL: http://link.springer.com/10.1007/978-3-642-17373-8_11, doi:10.1007/978-3-642-17373-8_11.

[LHW+25] Junkai Liang, Daqi Hu, Pengfei Wu, Yunbo Yang, Qingni Shen, and Zhonghai Wu. SoK: Understanding zk-SNARKs: The Gap Between Research and Practice. In *34th USENIX Security Symposium (USENIX Security 25)*, pages 2085–2104, 2025. URL: https://www.usenix.org/conference/usenixsecurity25/presentation/liang-sok.

[Lit23] Daniel Litinski. How to compute a 256-bit elliptic curve private key with only 50 million Toffoli gates, June 2023. URL: http://arxiv.org/abs/2306.08585, arXiv:2306.08585, doi:10.48550/arXiv.2306.08585.

[Lyu11] Vadim Lyubashevsky. Lattice Signatures Without Trapdoors. In *EUROCRYPT 2012*, 2011. URL: https://eprint.iacr.org/2011/537, doi:10.1007/978-3-642-29011-4_43.

[MPR+24] Dustin Moody, Ray Perlner, Andrew Regenscheid, Angela Robinson, and David Cooper. Transition to Post-Quantum Cryptography Standards. Technical Report NIST Internal or Interagency Report (NISTIR) 8547 (Draft), National Institute of Standards and Technology, November 2024. URL: https://csrc.nist.gov/pubs/ir/8547/ipd, doi:10.6028/NIST.IR.8547.ipd.

[MS26] Christian Majenz and Jaya Sharma. Security of the Fischlin Transform in Quantum Random Oracle Model, 2026. URL: https://eprint.iacr.org/2026/311.

[Nat24a] National Institute of Standards and Technology. Module-Lattice-Based Digital Signature Standard. Technical Report Federal Information Processing Standard (FIPS) 204, U.S. Department of Commerce, August 2024. URL: https://csrc.nist.gov/pubs/fips/204/final, doi:10.6028/NIST.FIPS.204.

[Nat24b] National Institute of Standards and Technology. Module-Lattice-Based Key-Encapsulation Mechanism Standard. Technical Report Federal Information

Processing Standard (FIPS) 203, U.S. Department of Commerce, August 2024. URL: https://csrc.nist.gov/pubs/fips/203/final, doi:10.6028/NIST.FIPS.203.

[Nat24c] National Institute of Standards and Technology. Stateless Hash-Based Digital Signature Standard. Technical Report Federal Information Processing Standard (FIPS) 205, U.S. Department of Commerce, August 2024. URL: https://csrc.nist.gov/pubs/fips/205/final, doi:10.6028/NIST.FIPS.205.

[Nat25] National Cyber Security Centre. Timelines for migration to post-quantum cryptography | National Cyber Security Centre - NCSC.GOV.UK, March 2025. URL: https://www.ncsc.gov.uk/guidance/pqc-migration-timelines.

[NIS25] NIS Cooperation Group PQC Work Stream. A Coordinated Implementation Roadmap for the Transition to Post-Quantum Cryptography, Part 1. Coordinated Implementation Roadmap, Part 1, Version 1.1, European Union, NIS Cooperation Group, June 2025. URL: https://digital-strategy.ec.europa.eu/en/library/coordinated-implementation-roadmap-transition-post-quantum-cryptography.

[NOZ26] Ngoc Khanh Nguyen, George O'Rourke, and Jiapeng Zhang. Hachi: Efficient Lattice-Based Multilinear Polynomial Commitments over Extension Fields, 2026. URL: https://eprint.iacr.org/2026/156.

[NS24] Ngoc Khanh Nguyen and Gregor Seiler. Greyhound: Fast Polynomial Commitments from Lattices. In *CRYPTO 2024*, 2024. URL: https://eprint.iacr.org/2024/1293, doi:10.1007/978-3-031-68403-6_8.

[NS25] Wilson Nguyen and Srinath Setty. Neo: Lattice-based folding scheme for CCS over small fields and pay-per-bit commitments, 2025. URL: https://eprint.iacr.org/2025/294.

[NS26] Wilson Nguyen and Srinath Setty. Neo and SuperNeo: Post-quantum folding with pay-per-bit costs over small fields, 2026. URL: https://eprint.iacr.org/2026/242.

[ORES24] Bjorn Oude Roelink, Mohammed El-Hajj, and Dipti Sarmah. Systematic review: Comparing zk-SNARK, zk-STARK, and bulletproof protocols for privacy-preserving authentication. *SECURITY AND PRIVACY*, 7(5):e401, 2024. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/spy2.401, doi:10.1002/spy2.401.

[Ott26] OtterSec. Unfaithful Claims: Breaking 6 zkVMs, 2026. URL: https://osec.io/blog/2026-03-03-zkvms-unfaithful-claims/.

[Pol22] Polygon Zero Team. Plonky2: Fast Recursive Composition of ZK-SNARKs, 2022. URL: https://github.com/0xPolygonZero/plonky2/blob/main/plonky2/plonky2.pdf.

[Pol26] Polygon Zero Team. Plonky3/Plonky3. Plonky3, June 2026. URL: https://github.com/Plonky3/Plonky3.

[PPG24] Christof Paar, Jan Pelzl, and Tim Güneysu. *Understanding Cryptography: From Established Symmetric and Asymmetric Ciphers to Post-Quantum Algorithms*. Springer, Berlin, Heidelberg, 2024. URL: https://link.springer.com/10.1007/978-3-662-69007-9, doi:10.1007/978-3-662-69007-9.

[Pre18] John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, August 2018. URL: http://arxiv.org/abs/1801.00862, arXiv:1801.00862, doi:10.22331/q-2018-08-06-79.

[RNSL17] Martin Roetteler, Michael Naehrig, Krysta M. Svore, and Kristin Lauter. Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms. In *ASIACRYPT 2017*, 2017. URL: https://eprint.iacr.org/2017/598, doi:10.1007/978-3-319-70697-9_9.

[SAKK25] Nojan Sheybani, Anees Ahmed, Michel Kinsy, and Farinaz Koushanfar. Zero-Knowledge Proof Frameworks: A Systematic Survey, April 2025. URL: http://arxiv.org/abs/2502.07063, arXiv:2502.07063, doi:10.48550/arXiv.2502.07063.

[Sho97] Peter W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing*, 26(5):1484–1509, October 1997. URL: http://arxiv.org/abs/quant-ph/9508027, arXiv:quant-ph/9508027, doi:10.1137/S0097539795293172.

[Sta22] Starknet. Part 2: A Decentralization and governance proposal for Starknet, July 2022. URL: https://www.starknet.io/blog/part-2-a-decentralization-and-governance-proposal-for-starknet/.

[Sta24] StarkWare Industries. SHARP: Shared Prover Architecture, 2024. URL: https://docs.starknet.io/learn/protocol/sharp.

[STW23] Srinath Setty, Justin Thaler, and Riad Wahby. Unlocking the lookup singularity with Lasso. In *EUROCRYPT 2024*, 2023. URL: https://eprint.iacr.org/2023/1216.

[Tha22] Justin Thaler. SNARK Security and Performance, September 2022. URL: https://a16zcrypto.com/posts/article/snark-security-and-performance/.

[Unr14] Dominique Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. In *EUROCRYPT 2015*, 2014. URL: https://eprint.iacr.org/2014/587, doi:10.1007/978-3-662-46803-6_25.

[Unr17] Dominique Unruh. Post-Quantum Security of Fiat-Shamir. In *ASIACRYPT 2017*, 2017. URL: https://eprint.iacr.org/2017/398, doi:10.1007/978-3-319-70694-8_3.

[Wat06] John Watrous. Zero-knowledge against quantum attacks. In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '06, pages 296–305, New York, NY, USA, May 2006. Association for Computing Machinery. URL: https://dl.acm.org/doi/10.1145/1132516.1132560, doi:10.1145/1132516.1132560.

[YZ20] Takashi Yamakawa and Mark Zhandry. Classical vs Quantum Random Oracles. In *EUROCRYPT 2021*, 2020. URL: https://eprint.iacr.org/2020/1270, doi:10.1007/978-3-030-77886-6_20.

[Zca21] Zcash Foundation. ZIP 224: Orchard Shielded Protocol, 2021. URL: https://zips.z.cash/zip-0224.

[Zha15] Mark Zhandry. A Note on the Quantum Collision and Set Equality Problems. *Quantum Information and Computation*, 15(7-8), 2015. URL: http://arxiv.org/abs/1312.1027, arXiv:1312.1027, doi:10.26421/qic15.7-8-2.

[Zha18] Mark Zhandry. How to Record Quantum Queries, and Applications to Quantum Indifferentiability. In *CRYPTO 2019*, 2018. URL: https://eprint.iacr.org/2018/276, doi:10.1007/978-3-030-26951-7_9.

[ZKs23] ZKsync. Boojum Upgrade: zkSync Era's New High-performance Proof System for Radical Decentralization, July 2023. URL: https://paragraph.com/@zksync/boojum-upgrade-zksync-era-s-new-high-performance-proof-system-for-radical-decentralization.

# A Glossary and abbreviations

Abbreviations used throughout the paper, sorted alphabetically. Each entry expands the acronym and gives a one-line meaning in the context of post-quantum (PQ) analysis of proof systems.

| Term | Meaning (in this paper) |
|---|---|
| AIR | Algebraic Intermediate Representation; a constraint format based on execution traces and transition polynomials, used in STARK-style systems. |
| Binius | Proof system over binary tower fields ($\mathbb{F}_{2^k}$); hash-based commitments, PQ-plausible. |
| BLS12-381 | A pairing-friendly elliptic curve with ≈381-bit base field and ≈255-bit scalar field (group order), used in Zcash, Ethereum beacon chain, and several proof systems. |
| BN254 | A pairing-friendly elliptic curve with ≈254-bit base-field size, widely used in Ethereum proof-system deployments (also known as alt_bn128). |
| BQP | Bounded-Error Quantum Polynomial Time; standard class for efficient quantum computation. |
| Circle STARKs | STARK-family constructions using circle-domain techniques over prime fields; structurally hash-based, expected to exhibit quantitative degradation, pending formal QROM verification of the circle-group variant. |
| CNFL | "Collect now, forge later"; the proof-system analogue of HNDL, threatening future proof forgery rather than past confidentiality. |
| CRQC | Cryptographically Relevant Quantum Computer; a fault-tolerant quantum machine capable of running Shor at practical cryptographic sizes. |
| CRS | Common Reference String; a public string generated during setup and shared between prover and verifier; more general than SRS (a CRS need not have algebraic structure). |
| DDH | Decisional Diffie–Hellman; a hardness assumption on group elements, broken by quantum adversaries via Shor's algorithm. |
| DLP | Discrete Logarithm Problem; hardness assumption broken by Shor's algorithm. |
| ECDLP | Elliptic Curve Discrete Logarithm Problem; the DLP restricted to elliptic-curve groups, broken by Shor's algorithm in polynomial time. |
| $d$-SDH | $d$-Strong Diffie–Hellman; a $q$-type hardness assumption over a pairing or prime-order group underlying KZG polynomial commitment binding; solvable given a discrete-log oracle and therefore broken by Shor. |
| EIP | Ethereum Improvement Proposal; a design document proposing changes to the Ethereum protocol; referenced here for precompiled-contract gas costs (EIP-197, EIP-1108) and data availability (EIP-4844). |
| EVM | Ethereum Virtual Machine; the execution environment for smart contracts on Ethereum, relevant here for on-chain verification cost analysis. |

*(continued on next page)*

| Term | Meaning (in this paper) |
|---|---|
| FIPS | Federal Information Processing Standard; U.S. standards published by NIST; referenced here for the finalized PQ standards (FIPS 203, 204, 205). |
| FRI | Fast Reed–Solomon Interactive Oracle Proof of Proximity; proximity test central to many STARK pipelines. |
| FS | Fiat-Shamir heuristic; compilation that replaces verifier randomness with hash outputs derived from the transcript. |
| Halo 2 | IPA-based proof system with recursion via accumulation; transparent but DLP-dependent (Shor-vulnerable). |
| HNDL | "Harvest now, decrypt later"; the standard post-quantum risk that ciphertext collected today may be decrypted once quantum attacks become practical. |
| HVZK | Honest-Verifier Zero-Knowledge; a zero-knowledge guarantee that holds when the verifier follows the protocol honestly (i.e., generates challenges uniformly at random). |
| IOP | Interactive Oracle Proof; proof framework where the prover provides oracle access to large objects and the verifier makes few queries. |
| IPA | Inner-Product Argument; proof technique used with vector commitments and transparent proof systems (often DLP-based). |
| IVC | Incremental Verifiable Computation; proving a long computation via incremental steps, often relying on recursion/folding. |
| KZG | Kate–Zaverucha–Goldberg polynomial commitment; pairing-based PCS with constant-size openings but Shor-vulnerable assumptions. |
| MLWE | Module Learning With Errors; lattice-based hardness assumption underlying NIST-standardized PQ primitives (ML-KEM / ML-DSA) and several lattice-based commitment schemes; conjectured quantum-hard. |
| MPCitH | MPC-in-the-Head; a paradigm for constructing zero-knowledge proofs by simulating a multi-party computation internally; explicitly out of scope in this paper. |
| MSM | Multi-Scalar Multiplication; core performance kernel for elliptic-curve-based proof systems. |
| NI | Non-Interactive; refers to the compilation of an interactive protocol into one where the prover sends a single message (typically via Fiat-Shamir). |
| NISQ | Noisy Intermediate-Scale Quantum; current era of non-fault-tolerant quantum devices with limited circuit depth. |
| NIST | National Institute of Standards and Technology; U.S. federal agency responsible for the PQ cryptography standardization process (FIPS 203–205). |
| NIZK | Non-Interactive Zero-Knowledge; a zero-knowledge proof system where the prover sends a single message to the verifier (no interaction). |
| Nova | A folding-based recursion/IVC scheme; common instantiations use group commitments (Shor-vulnerable unless redesigned). |
| PCS | Polynomial Commitment Scheme; commits to a polynomial and proves evaluation claims with a short opening proof. |
| PLONK | A family of SNARK protocols with PLONKish arithmetization; commonly instantiated with KZG or other PCS backends. |
| Plonky2/3 | Recursive-friendly proof systems using FRI over small fields (Goldilocks); Plonky3 is a modular generalization [Pol26]. |
| PPT | Probabilistic Polynomial Time; the standard complexity class for efficient classical adversaries. |
| PQC | Post-Quantum Cryptography; classical cryptography designed to remain secure against polynomial-time quantum adversaries. |

*(continued on next page)*

| Term | Meaning (in this paper) |
|---|---|
| PQ-plausible | No known polynomial-time quantum attack; security rests on conjectured quantum-hard assumptions (default term in this paper). |
| PQ-secure | Formal reduction to a specific assumption conjectured quantum-hard, with proof in a quantum-aware model. |
| QAP | Quadratic Arithmetic Program; a constraint representation used in pairing-based SNARKs (e.g., Groth16) where computation is encoded as polynomial divisibility checks. |
| QRACM | Quantum Random Access Classical Memory; a quantum access model for classical data tables; required as an assumption for some quantum collision-finding bounds (e.g., BHT's $2^{n/3}$ complexity). |
| QROM | Quantum Random Oracle Model; ROM where an adversary may query the oracle in superposition. |
| R1CS | Rank-1 Constraint System; a constraint format expressing computation as a system of bilinear equations over a finite field. |
| ROM | Random Oracle Model; idealized model where a hash function is treated as a random function. |
| SNARK | Succinct Non-interactive Argument of Knowledge; succinct (small) proofs with fast verification, typically computational soundness. |
| SRS | Structured Reference String; setup artifact with algebraic structure (e.g., powers of a secret trapdoor). |
| STARK | Scalable Transparent Argument of Knowledge; transparent (no trusted setup) proofs, typically hash/IOP based and larger than SNARKs. |
| ZK / ZKP | Zero-Knowledge / Zero-Knowledge Proof; proofs that convince a verifier without revealing the witness beyond validity. |