Stateless Versus Stateful Hash-Based Authorisation Inside a Private-Transaction STARK

Phexora Cash — unreleased clarifications to the Quantum 2-input/2-output protocol

Phexora AI · phexora.ai

2026-11-08

Abstract

This manuscript defines the pre-registration draft for the complete-transaction experiment in the Quantum research programme: a private note transaction with exactly two inputs and two outputs, including commitment openings, membership, nullifiers, encrypted-note binding, 64-bit integer conservation, and full SLH-DSA-SHAKE-256f/256s verification inside a transparent zero-knowledge STARK. It treats TzEL's public note/nullifier/ML-KEM/one-time-signature/STARK prototype as the closest engineering baseline identified by the scoped review completed on 23 July 2026 and rejects a generic post-quantum private-payment novelty claim. The remaining question is a comparative one: whether exact standardised stateless authorisation provides a material security, interoperability, or state-management benefit while meeting direct-layer-1 and constrained-client budgets. The protocol fixes the statement and witness boundary and specifies reproducibility, security-analysis, measurement, and stop/go requirements. It reports no implementation or result.

Keywords: post-quantum private payments; SLH-DSA; stateful hash-based signatures; STARK; comparative feasibility; reproducible benchmarking.

1. Status and claim boundary

Programme identity: This study belongs to Phexora Cash. Quantum remains the historical protocol codename used in this manuscript and stable filenames. The display-name change does not change the transaction relation, comparison arms, protocol identifiers, pre-registration or evidence requirements.

This is a **research protocol**, not a result paper. This working copy contains unreleased clarifications dated 2026-11-08 to the 0.4.0-research-protocol baseline. Campaigns must pin exact source hashes; this update is not an approved pre-registration. The status remains:

- no complete Quantum transaction prover or verifier exists in this repository;
- no TzEL-shaped or NIST stateful authorisation baseline has been independently implemented for comparison;
- no proof-system, commitment, or note-encryption profile has been frozen;
- no benchmark threshold has been approved under task T005;
- no benchmark described here has been run;
- no composed post-quantum security proof has been completed; and
- no feasibility, throughput, anonymity, audit, testnet, or production claim is made.

The manuscript exists to make the experiment falsifiable before implementation begins. Empty threshold fields, missing evidence, or an unimplemented relation are **not** passing results.

The authoritative protocol requirements remain the Quantum research design. The authoritative task dependencies and evidence gates remain the verification plan. If this protocol conflicts with either document, the conflict must be resolved explicitly; an implementation must not choose the more convenient interpretation.

1.1 Decision owned by this paper

The paper owns one decision:

> Does either exact FIPS 205 stateless profile, SLH-DSA-SHAKE-256f or SLH-DSA-SHAKE-256s, inside the complete Quantum 2-input/2-output relation meet every pre-registered security, latency, memory, proof-size, wire-size, and aggregation threshold, and does it provide a material benefit over a TzEL-shaped one-time baseline and an applicable NIST stateful baseline?

A positive answer may retain the stateless profile. Selecting 256s or a reviewed stateful profile requires a versioned profile decision before integration. T006 must first return NOT_RULED_OUT for receiving and transport; actual T305 formats and sizes must remain inside its frozen envelope before wallet, node and consensus integration may proceed. If no arm meets the frozen requirements, the signature, commitment, transaction relation, proof profile, or system requirements return to research, or the project stops. Failure is not relabelled as a future node optimisation.

1.2 Decisions not owned by this paper

This experiment does not establish:

- scalable or private wallet note discovery;
- network-origin anonymity;
- deterministic private-state application under GHOSTDAG;
- full-node storage, pruning, recovery, or denial-of-service resistance;
- sustained end-to-end throughput of 1,000 accepted layer-1 transactions per second;
- production security; or
- legal, economic, or market readiness.

Those questions have different adversaries, measurements, and failure modes. They are therefore separate manuscripts or later system evidence, not extra sections appended to a successful proving benchmark.

It also does not establish that note commitments, nullifiers, ML-KEM note delivery, in-proof hash-based authorisation, delegated proving, or recursive STARK private payments are new. TzEL already combines those architectural elements in an experimental public testnet system [TZEL].

2. Why a separate paper is necessary

The transaction relation is the earliest point where the selected standards, open cryptographic choices, privacy model, and client-performance target must work together. Testing the components separately cannot answer the decision in Section 1.1.

The representative relation includes all of the following at once:

1. two committed input notes and their openings;
2. two authenticated Merkle membership paths;
3. two correctly derived, locally distinct public nullifiers;
4. two public output commitments and their private openings;
5. binding between each output and its public encrypted-note payload;
6. complete verification of input authorisation under every retained comparison profile, including exact SLH-DSA-SHAKE-256f and SLH-DSA-SHAKE-256s;
7. canonical 64-bit values, limb ranges, carries, a public fee, and exact integer conservation;
8. one frozen transparent STARK transcript and zero-knowledge profile; and
9. the proposed individual-proof and, if retained, aggregate-proof wire modes.

Omitting any item answers an easier question and does not satisfy Quantum task T305.

3. Prior work and non-duplication boundary

3.1 Search scope

The novelty screen was refreshed on 23 July 2026. It reviewed primary material from NIST, the IACR Cryptology ePrint Archive and linked proceedings, peer-reviewed publications, protocol whitepapers, public implementation repositories, the Zcash Improvement Proposal repository, and the original PHANTOM/GHOSTDAG work. Search themes included:

- post-quantum confidential transactions and private payments;
- lattice RingCT with multiple inputs and outputs;
- post-quantum note/nullifier ledgers and public testnet implementations;
- transparent proofs and hash-signature verification or aggregation;
- anonymous KEMs and encrypted-note delivery;
- oblivious payment detection and light-client note discovery;
- transaction-origin anonymity; and
- GHOSTDAG ordering and state consistency.

This is a scoped novelty review, not proof that no related manuscript exists. It must be repeated before submission and recorded with queries, dates, and screening decisions in the evidence manifest.

A targeted source check on 8 November 2026 added SophOMR [SOPHOMR] and InstantOMR [INSTANTOMR] and checked the FIPS 203/205 parameter distinctions. This supplements the July novelty screen; it is not a new exhaustive search or a change to the T305 adopt/adapt/replicate decision.

3.2 Work that already addresses adjacent problems

Prior work	What it establishes or investigates	Why this paper must not repeat its claim
TzEL [TZEL] [TZELCODE]	An experimental Tezos rollup and public testnet combining note commitments, position-bound nullifiers, ML-KEM-768 note delivery, WOTS-like one-time spend authorisation under an XMSS-style tree, and recursive STARKs	The complete architectural combination closest to Quantum already exists as a public prototype; neither "post-quantum shielded payments" nor "hash-signature authorisation inside a private-payment STARK" remains available as a generic novelty claim
Lattice RingCT v1.0 [LRCT1] and v2.0 [LRCT2]	Post-quantum linkable-ring-signature transactions; v2.0 adds multiple inputs, multiple outputs, and a balance model	A generic claim to be the first post-quantum private or multi-input/multi-output payment protocol is unavailable
MatRiCT [MATRIX] and MatRiCT+ [MATRIXPLUS]	Concrete lattice-based RingCT designs and implementation measurements	Quantum must compare against their model, proof sizes, latency, anonymity scope, and assumptions rather than presenting post-quantum RingCT as an open invention
Gao et al. [GAO]	More efficient lattice-based zero-knowledge balance proofs and ring signatures, applied to RingCT	A new balance proof or ring-signature efficiency claim requires direct comparison and an actual construction

Prior work	What it establishes or investigates	Why this paper must not repeat its claim
LACT+ [LACTPLUS]	Aggregable lattice confidential transactions, many-input/many-output scaling, and a public implementation referenced by the paper	A lattice confidential-value or aggregation claim requires comparison; LACT+ does not itself supply Quantum's complete sender, recipient, network, or note-delivery privacy model
ZK-STARK [STARK], DEEP-FRI [DEEPFRI], and later zero-knowledge analysis [STARKZK]	Transparent proof foundations, proximity testing, and explicit zero-knowledge treatment	Selecting the STARK family does not itself constitute novelty or prove that the Quantum profile is sound or zero knowledge
Khaburzaniya et al. [HASHAGG]	STARK-based aggregation and thresholdisation of hash-based signatures with concrete experiments	"Hash signatures can be put behind a STARK" is already established in another relation and parameter regime
Sphinx-in-the-Head [SPHINXHEAD]	Modifies a SPHINCS+-style credential to make its verification practical in a different zero-knowledge group-signature setting	Its redesign reinforces that exact standard SLH-DSA verification is a real measurement gate; results for a custom credential cannot be reused as Quantum evidence
Post-Quantum Privacy Pass [PQPP]	A STARK-friendly post-quantum credential design with concrete token trade-offs	General-purpose post-quantum proofs of authorisation are not unique to Quantum
GeT a CAKE [KEMANON]	Uses KEM anonymity and fuzziness in black-box KEM-to-PAKE transformations and analyses Kyber for those properties	Ordinary KEM confidentiality and ML-KEM standardisation alone cannot support a receiver-anonymity claim
Oblivious Message Retrieval [OMR], PerfOMR [PERFOMR], and OCash [OCASH]	Private recipient detection/retrieval and anonymous light-client payments	Scalable private note discovery is substantial prior work and remains a separate Quantum integration question
SophOMR [SOPHOMR] and InstantOMR [INSTANTOMR]	Homomorphic retrieval improvements in throughput, latency and parallelism	Provider work, malicious-provider completeness, exact PQ parameters and integration costs remain separate gates; latency reduction is not equivalent to less total CPU work
Dandelion++ [DANDELION] and later P2P-anonymity analysis [P2PANON]	Transaction-origin routing and attacks/limits in cryptocurrency P2P networks	A generic transaction-origin anonymity paper would duplicate a mature line of work
PHANTOM/GHOSTDAG [GHOSTDAG]	A blockDAG ordering protocol and security analysis	GHOSTDAG selection alone does not solve Quantum's nullifier, anchor, reward, and private-state semantics

3.3 Closest-work delta: TzEL

TzEL changes the decision boundary more than the earlier RingCT work because it uses the same high-level shielded-note architecture as Quantum. Its public whitepaper describes note commitments, historical Merkle roots, nullifiers, ML-KEM recipient delivery, in-proof hash-based

authorisation, and recursive STARK validity proofs. Its repository reports a live experimental testnet path, while explicitly warning that the scheme and implementation are not audited or safe for real value [TZELCODE].

Surface	TzEL public baseline at the search cut-off	Quantum incumbent	Consequence for this paper
Spend model	One to seven input notes; recipient, change, and producer-fee outputs	Fixed two-input/two-output representative relation	Note membership, nullifiers, and balance are prior art; Quantum must justify only its exact relation and system constraints
Authorisation	WOTS-like one-time signatures, base 4 with 133 chains, under a depth-16 XMSS-style tree; verified inside the STARK	Stateless FIPS 205 SLH-DSA-SHAKE-256f verified inside the STARK	The publishable question is the standardisation/state-management/performance trade-off, not the idea of in-proof hash-signature authorisation
Recipient path	ML-KEM-768 plus authenticated encryption, separate detection and viewing material	ML-KEM-1024 candidate plus an unfrozen authenticated composition and receiver-key-privacy game	Generic post-quantum note delivery is prior art; parameter rationale and a stronger measured privacy definition may still differ
Proof and carriage	Cairo AIR, Stwo recursion, and Tezos DAL	Unfrozen transparent STARK carried by a native layer 1	Quantum must demonstrate a direct-layer-1 size and aggregation result rather than relying on a separate high-bandwidth availability layer
Reported 2-input transfer	About 289.5 KiB proof, 5.235 s proving, 16.66 GiB peak RSS, and 32 ms verification on AWS c8g.16xlarge	No implementation or measurement	TzEL is the minimum engineering comparator; its self-reported result is not a Quantum benchmark or a constrained-client pass
Proving trust boundary	Proving may be delegated; the prover receives sensitive witness material and can infer address-associated spent state	Current experiment requires a constrained-client local-proving measurement	Local proving without witness disclosure is a material systems question if its budget is frozen and met
Evidence	Public whitepaper, source, testnet path, and self-reported benchmarks; no production-security claim	Two independent implementations and pre-registered acceptance thresholds required	Quantum may contribute independent comparative evidence, not a generic construction claim

Surface	TzEL public baseline at the search cut-off	Quantum incumbent	Consequence for this paper
Reuse terms	Repository states "All rights reserved" at snapshot 0288c1c…	Future implementation licence unresolved	No TzEL source code may be copied without compatible permission and documented legal review

These figures are context, not a head-to-head result: the hardware, proof system, transaction shape, security parameters, and deployment model differ. The TzEL metrics must be reproduced or clearly labelled author-reported in any comparison.

3.4 Candidate contribution

The candidate contribution is now a comparative decision rather than a new private-payment architecture:

1. exact FIPS 205 SLH-DSA-SHAKE-256f and SLH-DSA-SHAKE-256s as standardised stateless challengers to TzEL-shaped one-time authorisation;
2. an applicable NIST SP 800-208 XMSS/XMSSMT or LMS/HSS profile as a separate stateful standards comparator, if its hardware and state requirements can be met [SP800208];
3. one complete, identical note/nullifier/encryption/balance relation around each retained authorisation arm so that only measured differences are attributed to authorisation;
4. pre-registered direct-layer-1, constrained-client, verifier, proof, transaction, and aggregation budgets;
5. explicit comparison with TzEL's delegated-proving and DAL trade-offs; and
6. two independently implemented, interoperable evidence pipelines.

The resulting manuscript is a comparative feasibility, standardisation, or negative-result paper unless it supplies and proves a genuinely new technique. It must not use "first," "novel protocol," or equivalent wording merely because Quantum changes parameter sets, consensus, or deployment.

3.5 Non-duplication gate before implementation

The versioned T305 prior-art and reuse decision records the initial comparison and chooses **ADAPT + REPLICATE**, not a blank construction or direct code adoption. Named owner and reviewer signatures remain mandatory before implementation. Before the T301–T305 profile is frozen, the record must remain current and compare at least TzEL, Lattice RingCT v2.0, MatRiCT, MatRiCT+, Gao et al., LACT+, and the proposed Quantum relation across:

- sender, recipient, amount, and transaction-graph privacy definitions;
- input/output multiplicity, balance, range, and double-spend models;
- post-quantum assumptions and use of standardised primitives;
- setup, commitment, authorisation, and disclosure properties;
- proof, signature, key, and complete wire sizes;
- proving, verification, memory, and batching measurements;
- note delivery, recipient discovery, and light-client implications; and
- available implementations, vectors, audits, licences, and reproducibility.

Every later revision must preserve one of three honest paths:

1. adopt or adapt a published construction where it meets the Quantum requirements;
2. retain the STARK relation and identify the exact missing property or measured trade-off that justifies new work; or
3. treat the work as a replication/comparative feasibility study rather than a new protocol contribution.

If the comparison cannot identify a material Quantum-specific research question, the correct action is to reuse prior work and not write a redundant protocol paper.

4. Research questions and hypotheses

4.1 Research questions

RQ0 — Prior-art delta. Under one complete and otherwise identical transaction relation, does standardised stateless authorisation provide a pre-registered material security, interoperability, or state-management benefit over the closest one-time/stateful baselines?

RQ1 — Completeness. Can the exact public statement and private witness in Section 6 express every required Quantum relation without an unchecked value, key, index, payload, or context?

RQ2 — Authorisation cost. What trace length, proving time, peak memory, proof-size contribution, and verifier cost result from two complete authorisations for each retained arm in Section 7.2 inside the selected STARK?

RQ3 — Composition. Can commitment openings, membership, nullifiers, authorisation, encrypted-note binding, and carry-safe balance be composed without lowering the declared post-quantum security target below 128 bits or introducing witness leakage?

RQ4 — Client feasibility. Does the complete prover meet every frozen single-wallet p50, p95, and p99 latency and memory threshold on both reference hardware profiles?

RQ5 — Wire feasibility. Do the individual and proposed aggregate modes meet their frozen proof-size, transaction-size, verifier, latency, and amortisation thresholds without transmitting the proofs that an aggregate claims to replace?

RQ6 — Reproducibility. Can an independent implementation reproduce the statement encoding, accepted/rejected vectors, proofs, and reported metrics within the pre-registered tolerance?

4.2 Falsifiable hypotheses

Threshold symbols below are intentionally unresolved until T005 freezes their numeric values. T004 first pins the measurement method, artifact harness, and benchmark environment. Every threshold must then be replaced by a signed, versioned T005 value before the first measured feasibility run.

ID	Hypothesis	Required owner(s)
H0	At least one authorisation arm meets all frozen requirements; a stateless arm is retained only if its pre-registered material benefit justifies its measured overhead over the qualifying stateful arms	T001/T202/T305; T005 for any numeric benefit threshold
H1	Both implementations accept every valid canonical vector and reject every invalid vector identically	T003/T004
H2	Complete desktop proving p95 and p99 do not exceed L_desktop_p95 and L_desktop_p99	T005
H3	Complete constrained-client proving p95 and p99 do not exceed L_client_p95 and L_client_p99	T005

ID	Hypothesis	Required owner(s)
H4	Peak and steady resident memory remain below the frozen hardware-specific budgets	T005
H5a	Individual proof bytes remain below B_proof	T005
H5b	Consensus-transaction bytes remain below B_consensus_tx	T005
H5c	External encrypted payload and total payment bytes remain below B_external_payload and B_total_payment	T005
H5d	Payload-provider and retrieval traffic remain below B_payload_provider	T005
H6	Verifier latency, memory, and malformed-proof work remain below their frozen budgets	T005
H7	Aggregate mode meets size and latency budgets and omits replaced individual proofs	T304/T005
H8	Concrete composed soundness and privacy analysis reaches at least 128 post-quantum bits under the declared model	T001/T303/T701
H9	The second implementation reproduces metrics within the frozen tolerance and no semantic mismatch remains	T004/T005/T305

A hypothesis with an unset threshold is **not evaluated**. A run performed before threshold freeze is exploratory and cannot be relabelled as the acceptance run.

5. Fixed design pressures from selected standards

The following figures are standard encodings, not benchmark results.

FIPS 205 specifies a 49,856-byte signature for SLH-DSA-SHAKE-256f. Two independently authorised inputs therefore place 99,712 signature bytes in the private witness before keys, notes, membership paths, or proving-system data are counted. These bytes need not be on the public transaction wire if verification is proved in zero knowledge, but all required hash work remains part of the proved computation.

For SLH-DSA-SHAKE-256s, FIPS 205 gives 29,792 signature bytes, or 59,584 bytes for two independently authorised inputs. Arm D must measure complete signing and proving alongside Arm C; fewer witness bytes alone do not establish lower latency, smaller public proofs, or the composed security target. FIPS 205 Table 2 is the source of both sizes.

FIPS 203 specifies a 1,568-byte ciphertext for ML-KEM-1024. A one-ciphertext-per-output profile would place at least 3,136 KEM ciphertext bytes on a 2-output transaction before authenticated payloads, commitments, nullifiers, proof data, and framing. At 1,000 such transactions per second, the KEM ciphertexts alone would be 270,950,400,000 bytes per day. This is a design-pressure calculation, not a prediction of the final wire format.

These values make three shortcuts invalid:

- measuring a mock or smaller signature profile;
- reporting only prover throughput while hiding single-wallet latency; or
- treating note delivery and wallet scanning as negligible because the proof itself is compact.

TzEL's repository reports about 289.5 KiB for its recursive proof and, for its two-input transfer workload on AWS c8g.16xlarge, 5.235 seconds proving, 16.66 GiB peak resident memory, and 32 milliseconds verification [TZELCODE]. Those author-reported figures are not directly comparable with Quantum's unfrozen proof system or constrained-client profile. They do show that the closest public implementation identified in the scoped review is already well above the illustrative 25 KB direct-layer-1 transaction budget in the Quantum design unless a different aggregation, carriage, or system requirement is approved.

The 25 KB figure is an illustrative topology budget, not a frozen T005 limit. T305 must distinguish amortised proof bytes from complete transaction bytes: the latter still include the selected ciphertext/payload carriage, commitments, nullifiers and framing. The specification's Section 4.5 also requires the wallet-to-validator path, including any aggregators, to be budgeted before wide distribution; omitting individual proofs from a block does not erase earlier gossip traffic.

SophOMR's USENIX prepublication Table 4 reports 157 seconds for 65,536 612-byte payloads, with at most 50 pertinent payloads, on one thread [SOPHOMR]. A linear extrapolation to 2,000 outputs/s gives about 115 thread-hours per recipient per day of history. This is an author-reported benchmark extrapolation, not a measured Quantum cost, an end-to-end provider budget or a STOP decision. Exact payloads, security, preprocessing, service concurrency and completeness protection require their own T006 accounting. Its passive receiver-privacy model does not establish malicious-provider completeness. InstantOMR's latency/parallelism gains likewise do not establish lower total work [INSTANTOMR]. Both remain comparison inputs to T006.

6. Frozen experiment statement

6.1 Public inputs

The canonical public input must contain exactly:

```
PublicTransaction {
    protocol_version
    chain_id
    anchor_root
    anchor_context
    nullifiers[2]
    output_commitments[2]
    encrypted_note_payloads_or_digests[2]
    fee_u64
    fee_context
    expiry_context
    transaction_flags
    authorisation_profile
    proof_policy
}
```

Every integer and byte string must have one canonical encoding and one maximum length. Vector counts are fixed to two for this experiment. The statement must reject unknown versions or authorisation profiles, trailing data, duplicate nullifiers, duplicate output commitments, and non-canonical encodings before expensive verification work.

The encrypted_note_payloads_or_digests choice is not a byte- accounting escape hatch. If only digests are consensus-carried, the experiment must also produce the complete encrypted note payloads and report separately:

- consensus-transaction bytes;
- external encrypted-note payload bytes;
- total generated bytes per payment;
- payload-provider and retrieval bandwidth; and
- availability and retention assumptions.

Every payload digest binds the resulting ciphertext. The pre-encryption associated data binds output index, commitment and semantic context; the final effects ID then binds that payload digest under specification Section 6.1. The ciphertext must not depend on the final ID. An external payload required for receipt, scanning, recovery, or spending remains part of client and system feasibility even when it is not part of the layer-1 transaction encoding.

The experiment reuses T402's canonical effects codec, pre-encryption context, effects ID, authorisation digest and proof-policy interface. T304 supplies a separate bounded proof envelope; it is not an input to the signature it proves. The explicit fee context uses the selected static or dynamic profile, including epoch/weight fields when applicable. Eventual proof size or aggregation cannot reprice an already signed transaction. T305 may use a frozen static-fee instance; a later dynamic T506 profile must revalidate the exact relation and costs before integration claims. The codec cannot be replaced with a prototype-only format that omits semantic fields.

6.2 Private witness

For each input:

```
InputWitness {
    note_plaintext
    commitment_randomness
    membership_path
    membership_index
    nullifier_secret
    authorisation_public_key
    authorisation_witness
}
```

For each output:

```
OutputWitness {
    note_plaintext
    commitment_randomness
    note_encryption_witness
}
```

The witness also contains only the proof-system randomness and auxiliary trace data required by the frozen profile. Hidden preprocessing, trusted witness services, and omitted input or output openings are forbidden.

6.3 Required constraints

The AIR or equivalent relation must establish:

1. canonical version, chain, asset, and domain context;
2. both input commitment openings;
3. both complete membership paths against the public anchor;
4. both nullifiers bound to the opened notes, keys, authenticated positions and immutable creation profiles under specification Section 5.6.1;
5. local nullifier uniqueness;
6. both authorisation keys bound by their opened input notes;
7. two complete verifications for the declared authorisation profile over the same canonical transaction-authorisation digest, with no mixed-profile transaction;
8. both output commitment openings;
9. both encrypted-note bindings to the same output plaintext, output index, transaction context, chain, and version;
10. canonical unsigned 64-bit ranges for all values and the fee;
11. carry-constrained integer equality input_0 + input_1 = output_0 + output_1 + fee;
12. output-commitment uniqueness; and

13. agreement of counts, ordering, flags, expiry, fee context, authorisation profile, and proof policy across the statement, signed digest, encryption binding, and proof.

Global nullifier freshness, current-anchor admissibility, expiry, and version-activation checks remain external state checks. Their absence from the AIR must not be represented as proof that the transaction is globally admissible.

7. Cryptographic profiles that must be frozen

T001 must justify every exact parameter set against the composed R3 target. NIST categories are comparative security benchmarks, not direct statements of 128 post-quantum bits after multi-user and lifetime losses. For example, SLH-DSA-128s belongs to category 1 and ML-KEM-768 to category 3; their smaller encodings do not demonstrate security-equivalent replacements for the current category-5 candidates [FIPS203, FIPS205]. A changed security or hash profile requires a separately reviewed campaign, with the same R3 acceptance target.

7.1 Commitment

The report must identify one reviewed construction, exact parameters, setup rules, randomness sampler, encodings, hiding and binding definitions, and concrete post-quantum estimates. A family name such as Module-LWE/Module-SIS or BDLOP is insufficient.

Hash-based and lattice-based commitments are eligible under specification 4.4 and T103/T104. Freeze the construction before comparing authorisation arms. For commitments, Merkle paths and proof-system hashes, identify the hash, digest length and concrete security argument, and measure its trace cost. An algebraic hash requires its own review and standardisation-status statement; an existing SHAKE gadget does not make additional SHAKE evaluations free.

If the commitment cannot efficiently bind every note field required by the specification inside the selected proof system, the experiment fails its completeness condition.

7.2 Authorisation

Every retained arm must use the same note, membership, nullifier, encryption, balance, statement, proof, hardware, and measurement profile. The only intended experimental variable is spend authorisation. The report must pin for every arm:

- exact construction, parameter set, standard or source revision, and errata state;
- interface, context string, state-transition rules, and signing randomness;
- key and signature encodings;
- transaction-authorisation digest;
- in-circuit hash and address/domain encodings;
- malformed-key and malformed-signature behaviour.

The experiment starts with four arms:

1. **Arm A — TzEL-shaped one-time baseline:** an independently specified WOTS/XMSS-style relation that reproduces the public one-time and tree-state design questions without copying TzEL code;
2. **Arm B — NIST stateful comparator:** one exact XMSS/XMSSMT or LMS/HSS profile from NIST SP 800-208, retained only if its controlled key-generation and state requirements are applicable to the intended wallet model;
3. **Arm C — Quantum incumbent:** exact FIPS 205 SLH-DSA-SHAKE-256f, including the FIPS API, message-bound signature caveat, official KATs, and randomised or deterministic signing choice; and
4. **Arm D — Standardised stateless comparator:** exact FIPS 205 SLH-DSA-SHAKE-256s under the same complete relation, KAT, message-bound-signature, security and measurement requirements as Arm C.

For Arms A and B, the security and operational analysis must cover key-index allocation, crash consistency, backup/restore rollback, concurrent signing, key exhaustion, device loss, recovery, and state desynchronisation. For Arms C and D, it must quantify the statelessness benefit and the full additional proving cost. NIST SP 800-230 is an initial public draft at the evidence cut-off, not a final source of protocol parameter sets [SP800230].

The scoped NIST update checked on 2026-09-05 still lists SP 800-230 as an initial public draft, titled Additional SLH-DSA Parameter Sets for Limited-Signature Use Cases. A separately preregistered exploratory arm must pin the exact draft and enforce its 2^24-signatures-per-key lifetime ceiling across devices, retries and backup restores. It cannot silently replace either required FIPS 205 arm or be described as a final standard. Signature generation, not merely accepted on-ledger spends, counts toward that limit.

A proof-friendly research signature may be measured only as a separately labelled exploratory arm. It cannot silently replace any arm or become normative without a revised specification, threat model, prior-art decision, and human review.

7.3 Recipient encryption and binding

The report must pin the ML-KEM interface, KDF, authenticated encryption, nonce rules, associated data, decapsulation-failure behaviour, replay rules, key separation, and the receiver-anonymity game. The binding proved inside the transaction must cover the actual output plaintext and public payload or payload digest.

ML-KEM IND-CCA security does not by itself imply that a public ciphertext hides which eligible recipient key was used.

7.4 STARK

The report must pin:

- base and extension fields and encodings;
- AIR columns, constraints, degrees, and trace padding;
- commitment hashes and digest lengths;
- Fiat–Shamir transcript order and domain tags;
- challenge sampling and rejection rules;
- FRI or DEEP-FRI variant, blow-up, queries, grinding, and batching;
- zero-knowledge masks and leakage argument;
- quantum-random-oracle and multi-proof soundness accounting;
- proof and public-input encoding; and
- verifier resource limits before attacker-controlled allocation.

The analysis must account for the full composed experiment. A nominal field size, a count of queries multiplied by “bits”, or a failure-free test run is not a soundness proof.

8. Independent implementation protocol

8.1 Separation

Two teams must implement the frozen profile from the public documents and canonical vectors. They may share:

- specifications;
- generated external-standard KATs with provenance;
- the canonical wire schema; and
- a language-neutral vector corpus.

They must not share the same prover/verifier core, parser implementation, AIR generator, or expected-value code path. Wrapping the same library in two languages does not create independent implementations.

The teams must not copy TzEL source at the recorded all-rights-reserved snapshot. They may implement an independently written baseline from published descriptions only after the experiment owner records provenance and legal review of the intended use. Until then, TzEL measurements remain author-reported context rather than reproduced results.

8.2 Required vectors

The corpus must include:

- minimum, typical, and maximum valid 64-bit values;
- zero and maximum permitted fees;
- carry boundaries including 65,535 + 1;
- distinct valid membership positions;
- deterministic output-encryption test fixtures;
- every retained authorisation profile and its state transitions;
- key-index allocation, exhaustion, crash, restore, rollback, and concurrent signing cases for stateful profiles;
- both proof modes, if aggregate mode remains;
- every malformed canonical encoding class; and
- every negative relation in Section 8.3.

Every vector must record its schema version, generator revision, provenance, expected result, and independent confirmation.

8.3 Required negative relations

At minimum, each implementation must reject a proof attempt or proof for:

1. an altered input value with the original commitment;
2. an altered input commitment with the original path;
3. an unconstrained or wrong membership index;
4. a nullifier derived from a different note or key;
5. duplicate input nullifiers;
6. an authorisation key not bound by the input note;
7. a valid signature over a partial or differently ordered transaction;
8. a malformed key, signature, state index, or profile identifier;
9. an altered output plaintext with the original commitment;
10. an encrypted payload bound to another output index or transaction;
11. values equal only after proof-field wraparound;
12. an unconstrained 16-bit limb or carry;
13. an omitted or altered fee, chain identifier, version, or expiry;
14. duplicate output commitments;
15. transcript domain or proof-mode confusion; and
16. an aggregate proof bound to a different ordered transaction set;
17. reuse of a one-time key or state index;
18. state rollback after backup restore or interrupted signing;
19. a signature from an exhausted or different authorisation tree;
20. a transaction that mixes authorisation profiles across its inputs;
21. a nullifier bound to a different leaf position or creation profile;
22. a changed ciphertext/fee context with the old effects ID or signature; and
23. an envelope carrying effects not authorised by its admitted proofs.

Integration vectors must additionally cover identical notes accepted in separate transactions, old-note spending across active-version changes, reorged positions, and exactly-once migration. These use T403/T510 state contexts; a T305 AIR-only pass does not prove global freshness. Positive vectors must show that policy-permitted proof replacement preserves txid and fee. Builder tests reject self-referential digest dependencies and out-of-profile proof carriage; no fixed-point

search is an accepted builder. The full corpus includes A039–A052 from the verification guide where relevant to each task, following its scenario-to-requirement mapping. T305 covers the assigned aspects of A049 (proof carriage), A050 (helper-service costs and failure assumptions), and A052 (stateful signing after crash/restore). Receiving-completeness and state-recovery scenarios retain their mapped owners; this cross-reference does not add them as transaction-proof constraints.

9. Benchmark protocol

T006 must have a signed NOT_RULED_OUT result before this campaign. The report must link its envelope and compare measured output, proof and packet sizes against it. A mismatch stops integration and requires a new reviewed screen; T305 cannot promote simulation into receiving or anonymity assurance.

9.1 Pre-registration

Before acceptance measurements, T004 must publish and sign the benchmark method, artifact harness, and environment:

- exact hardware identifiers, CPU features, memory, storage, and power mode;
- operating systems, kernels, compilers, dependencies, and revisions;
- clock source and measurement tooling;
- cold/warm cache policy and preprocessing policy;
- workload seeds and number of samples;
- allowed parallelism and thread affinity;
- rules for exclusions, crashes, retries, and failed proofs.

T202 and T305 must publish and sign the rule for excluding an inapplicable NIST SP 800-208 arm before results are viewed, including reviewer rationale, and the material-benefit rule for retaining each stateless arm (C and D). T005 must then publish and sign every numerical acceptance threshold, including:

- p95 and p99 latency, memory, verifier, proof, wire, and aggregate budgets;
- separate consensus-transaction, external encrypted-payload, total-payment, and payload-provider/retrieval budgets;
- any minimum acceptable material-benefit delta; and
- the reproduction tolerance.

Any change after viewing acceptance results creates a new experiment version and requires a new pre-registration.

9.2 Measurements

The raw record for every run must include:

- authorisation arm and exact parameter revision;
- end-to-end proof-generation latency;
- per-component trace rows/cycles, time and memory for authorisation, commitments, Merkle paths, nullifiers, ML-KEM/KDF/AEAD binding, ranges/balance and proof overhead, with shared work accounted once;
- peak and steady resident memory;
- proof bytes, full canonical consensus-transaction bytes, external encrypted- note payload bytes, and total generated bytes per payment;
- payload-provider/retrieval bandwidth and the availability and retention assumptions under which delivery succeeds;
- verifier wall time, CPU time, and peak memory;
- malformed-proof rejection time and allocation;

- parallel-prover scaling and saturation;
- aggregate creation latency, size, verifier cost, and amortisation;
- per-hop admission/distribution bytes and fanout, batch-wait latency, aggregator capacity, retries, withholding and outage behaviour;
- preprocessing time and bytes; and
- all warnings, failures, retries, and outliers.

Single-wallet measurements run with only the frozen wallet parallelism. Separate throughput experiments may use additional workers, but their results must not be substituted for wallet latency.

Analytical hash-call and Keccak-permutation counts must state the operation, message sizes, average/worst-case assumptions and excluded work. They do not measure AIR constraints, trace padding or end-to-end proving time. Neither signature nor encryption dominance may be reported as a result without the component measurements above.

9.3 Environments

At least two named classes are required:

1. **reference desktop**, representing the intended proving environment; and
2. **constrained client**, representing the weakest client that the release requirements claim can construct a private transaction locally.

Cloud hardware may provide supplemental throughput data but cannot silently replace either frozen client profile.

10. Security-analysis deliverables

The feasibility report must be accompanied by:

1. a formal relation and traceability map to every constraint in Section 6.3;
2. commitment correctness, hiding, and binding analyses;
3. transaction unforgeability and prevention-of-unauthorised-issuance arguments under the selected monetary-policy invariant;
4. recipient-key privacy and encrypted-note integrity analyses;
5. STARK completeness, knowledge soundness, and zero-knowledge analyses;
6. Fiat–Shamir/QROM treatment for the exact adaptive and multi-proof setting;
7. concrete post-quantum security accounting across all relevant targets and the intended protocol lifetime;
8. side-channel scope and constant-time review for secret-dependent work; and
9. a list of assumptions and properties not covered.

The benchmark may show that an implementation is fast enough. It cannot prove any item in this list by measurement alone.

11. Evidence package

A complete package contains:

```
evidence/t305/<experiment-id>/
  preregistration/
      thresholds.json
      hardware.json
      signatures/
  profile/
      relation.md
      encodings.md
      cryptographic-profile.md
      dependency-locks/
```

```
vectors/
    manifest.json
    positive/
    negative/
implementation-a/
implementation-b/
raw/
reports/
    security-analysis.pdf
    benchmark-report.pdf
    interoperability-report.md
    unresolved-risks.md
decision/
    stop-go.md
    approvals/
```

The real implementation repository may choose a different root, but the manifest must map every logical artifact above to an immutable digest and revision. This documentation repository does not pretend that those artifacts already exist.

12. Stop/go rules

The experiment first applies the common evidence gate. An arm qualifies only if all of the following are true:

- the frozen relation contains every required constraint;
- a signed T006 screen covers the actual output, proof and packet sizes;
- two genuinely independent implementations interoperate;
- every positive and negative vector has the expected result;
- every pre-registered desktop and constrained-client threshold passes;
- individual and aggregate wire modes, external payload carriage, total payment bytes, and provider traffic meet their separate thresholds;
- the composed security analysis reaches the required target;
- no critical or high unresolved review finding remains;
- all raw artifacts and reproduction commands are public; and
- accountable human reviewers sign their respective boundaries.

The signed result then uses exactly one classification:

- **GO — retain stateless:** Arm C or D qualifies and satisfies the frozen material-benefit rule relative to every qualifying stateful arm. Selection of Arm D changes the normative profile before integration.
- **ADAPT — select stateful:** a reviewed Arm A or B qualifies while neither stateless arm provides a qualifying material benefit. The normative specification changes before integration.
- **REPLICATE — publish comparison or negative result:** the evidence is reproducible and useful but supports no new construction or qualifying deployment profile.
- **STOP — redesign:** no arm satisfies the common direct-layer-1, constrained-client, security, and privacy requirements.

Any non-GO outcome must identify which of these design surfaces returns to research:

- authorisation profile;
- commitment construction;
- transaction relation;
- note-encryption composition;
- proof-system profile;
- client-support claim; or
- aggregate mode.

Removing a required relation, weakening privacy, adding a classical fallback, or changing a threshold after measurement is not an optimisation.

13. Interfaces to the remaining research papers

If T305 passes, three separate research questions remain:

1. **Private post-quantum note discovery:** use the frozen output format and measured output rate to evaluate local scanning, compact broadcast, OMR, OCash-like, or another explicitly modelled design.
2. **Network-origin anonymity:** use the frozen transaction size and target rate to evaluate a named observer model, cover traffic, delay, reliability, and denial-of-service behavior.
3. **GHOSTDAG private-state semantics:** define deterministic anchor, nullifier, commitment, reward, reorganisation, and finality behavior under the exact selected order.

Templates for these manuscripts live in the research-template directory. They are not results and must not be promoted to active papers until their entry criteria are met.

14. Limitations

This protocol fixes the shape of a decisive experiment but does not supply the missing cryptographic profiles or implementations. Its 2-input/2-output shape is representative, not proof that all future vector lengths or payment policies have the same costs. A successful laboratory result does not establish side-channel security, network anonymity, consensus safety, operational stability, or production readiness.

The literature screen may miss unpublished, non-indexed, differently terminologised, or newly released work. Before external submission, authors must refresh the search, obtain specialist review, and narrow all originality claims to demonstrated differences.

TzEL's whitepaper, repository, and benchmark tables are public engineering evidence, not a peer-reviewed proof of production security or an independent Quantum measurement. The implementation explicitly warns that it is experimental and unaudited. Its source terms also prevent treating public availability as permission to reuse code. These limitations must remain visible in any comparison.

References

- [**SOPHOMR**] Lee and Yeo, SophOMR: Improved Oblivious Message Retrieval from SIMD-Aware Homomorphic Compression, ePrint 2024/1814, USENIX Security 2026; consulted 2026-11-08. The 157-second figure is from Table 4 of the USENIX prepublication, not a Quantum reproduction.
- [**INSTANTOMR**] Liang et al., InstantOMR: Oblivious Message Retrieval with Low Latency and Optimal Parallelizability, ePrint 2025/2317, revision 2026-05-02, USENIX Security 2026; consulted 2026-11-08. The latency/throughput distinction is discussed in Section 6.5 of the USENIX prepublication.
- [**FIPS203**] NIST, FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard, 2024.
- [**FIPS205**] NIST, FIPS 205: Stateless Hash-Based Digital Signature Standard, 2024.
- [**SP800208**] NIST, SP 800-208: Recommendation for Stateful Hash-Based Signature Schemes, 2020.
- [**SP800230**] NIST, SP 800-230 initial public draft: Additional SLH-DSA Parameter Sets for Limited-Signature Use Cases, 2026; non-final at the evidence cut-off.
- [**TZEL**] TzEL contributors, TzEL whitepaper, 2026.
- [**TZELCODE**] Trilitech, TzEL source snapshot, accessed 2026-07-23.

- [**LRCT1**] Torres et al., Post-Quantum One-Time Linkable Ring Signature and Application to Ring Confidential Transactions, 2018.
- [**LRCT2**] Torres et al., Lattice RingCT v2.0 with Multiple Input and Multiple Output Wallets, 2019.
- [**MATRIX**] Esgin, Zhao, Steinfeld, Liu and Liu, MatRiCT: Efficient, Scalable and Post-Quantum Blockchain Confidential Transactions Protocol, 2019.
- [**MATRIXPLUS**] Esgin, Steinfeld and Zhao, MatRiCT+: More Efficient Post-Quantum Private Blockchain Payments, 2022.
- [**GAO**] Gao et al., Lattice-based Zero-knowledge Proofs for Blockchain Confidential Transactions, revised 2025.
- [**LACTPLUS**] Alupotha, Boyen and McKague, LACT+: Efficient Lattice-Based Aggregatable Confidential Transactions with Hidden Amounts, 2023.
- [**STARK**] Ben-Sasson et al., Scalable, Transparent, and Post-Quantum Secure Computational Integrity, 2018.
- [**DEEPFRI**] Ben-Sasson et al., DEEP-FRI: Sampling Outside the Box Improves Soundness, 2019.
- [**STARKZK**] Haböck and Al Kindi, A Note on Adding Zero Knowledge to STARKs, 2024.
- [**HASHAGG**] Khaburzaniya, Chalkias, Lewi and Malvai, Aggregating and Thresholdizing Hash-based Signatures Using STARKs, 2022.
- [**SPHINXHEAD**] Chen, Dong, Newton and Wang, Sphinx-in-the-Head: Group Signatures from Symmetric Primitives, 2024.
- [**PQPP**] Policharla, Westerbaan, Faz-Hernández and Wood, Post-Quantum Privacy Pass via Post-Quantum Anonymous Credentials, 2023.
- [**BDLOP**] Baum et al., More Efficient Commitments from Structured Lattice Assumptions, 2016.
- [**KEMANON**] Beguinet, Chevalier, Pointcheval, Ricosset and Rossi, GeT a CAKE: Generic Transformations from Key Encapsulation Mechanisms to Password Authenticated Key Exchanges,

2023.

- [**OMR**] Liu and Tromer, Oblivious Message Retrieval, 2021.
- [**PERFOMR**] Liu et al., PerfOMR: Oblivious Message Retrieval with Reduced Communication and Computation, 2024.
- [**OCASH**] Hansen, Nielsen and Simkin, OCash: Fully Anonymous Payments between Blockchain Light Clients, revised 2025.
- [**ZIP307**] Zcash Improvement Proposals, ZIP 307: Light Client Protocol for Payment Detection, current specification.
- [**DANDELION**] Fanti et al., Dandelion++: Lightweight Cryptocurrency Networking with Formal Anonymity Guarantees, 2018.
- [**P2PANON**] Sharma, Gosain and Diaz, On the Anonymity of Peer-To-Peer Network Anonymity Schemes Used by Cryptocurrencies, 2022.
- [**GHOSTDAG**] Sompolinsky, Wyborski and Zohar, PHANTOM and GHOSTDAG, 2018.

Appendix A — Revision record

Unreleased — 2026-11-08

- identified the programme as Phexora Cash while retaining the historical Quantum protocol codename and the comparative experiment contract;
- added a targeted SophOMR/InstantOMR source update, explicitly separate from the earlier literature screen and from protocol measurements;
- clarified parameter justification, component-cost measurement and the full proof-distribution path, including helper-service costs;

- extended the scenario cross-reference through A052 with task-specific applicability; and
- retained the published baseline identity without recording a new release, approved campaign, benchmark result or gate pass.

0.4.0-research-protocol — 2026-09-05

- added standardised 256s Arm D and bounded any SP 800-230 exploratory arm;
- reused the canonical T402 effects codec and staged digest schedule, added fee context, and bound input nullifiers to authenticated note instances;
- required the T006 receiving/transport screen and an actual-size envelope check before integration; this is not wallet or anonymity assurance;
- added cross-transaction duplicate-note, version-continuity, digest/envelope and fee-carriage negative cases; no experiment or approval is recorded.

0.3.2-research-protocol — 2026-08-09

- required digest-only layer-1 note carriage to report external encrypted payload bytes, total generated bytes per payment, provider/retrieval traffic, and availability/retention assumptions rather than hiding them from the wire feasibility budget; and
- updated the normative cross-reference to the 0.5.2 specification and verification plan without reporting a benchmark result.

0.3.1-research-protocol — 2026-08-09

- aligned security-analysis terminology with the selected monetary-policy invariant and prevention of unauthorised issuance; and
- updated the normative cross-reference to the 0.5.1 specification and verification plan without changing the registered experiment relation or threshold ownership.

0.3.0-research-protocol — 2026-08-09

- assigned the benchmark method, artifact harness, and pinned environment to T004 while assigning all numerical acceptance thresholds to T005;
- separated the qualitative authorisation-selection rule owned by T202/T305 from any numerical material-benefit delta frozen by T005; and
- aligned the feasibility protocol with the 0.5.0 specification and verification task graph without reporting a benchmark result.

0.2.0-research-protocol — 2026-07-23

- identified TzEL as the closest public engineering baseline found by the scoped review and rejected a generic post-quantum private-payment novelty claim;
- reframed the experiment as a same-relation comparison of a TzEL-shaped one-time baseline, an applicable NIST SP 800-208 stateful profile, and exact FIPS 205 SLH-DSA-SHAKE-256f;
- added state-management failure cases, a requirement to pre-register the material-benefit rule, source-reuse restrictions, and GO/ADAPT/REPLICATE/STOP outcomes.

0.1.0-research-protocol — 2026-07-23

- defined the complete two-input/two-output T305 relation, independent implementations, benchmark fields, security deliverables, and stop/go gate;
- separated the three downstream questions into future-paper templates.