PQTrust-Agent: Policy-Compiled Post-Quantum Trust Contracts for Web-of-Agents Communications

Yulliwas Ameur; Insaf Imene Lasledj; Samia Bouzefrane

2026 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)

Accepted conference contribution. Proceedings forthcoming.

Manuscript snapshot: 6 September 2026. This public copy adds a version and copyright cover; the research manuscript on the following pages is unchanged.

PQTrust-Agent: Policy-Compiled Post-Quantum Trust Contracts for Web-of-Agents Communications

Yulliwas Ameur[2,3,*], Insaf Imene Lasledj[1], Samia Bouzefrane[3]
[1]Laboratoire LITAN, École Supérieure en Sciences et Technologies
de l'Informatique et du Numérique (ESTIN), Béjaïa, Algeria
i_lasledj@estin.dz
[2]Efrei Research Lab, Efrei, Université Paris-Panthéon-Assas, Villejuif, France
yulliwas.ameur@efrei.fr
[3]CEDRIC, Conservatoire national des arts et métiers, Paris, France
samia.bouzefrane@lecnam.net

***Abstract*—Autonomous Web agents can agree on a task while enforcing incompatible cryptographic, fallback, resumption, and lease policies. This paper presents PQTRUST-AGENT, a constraint-first bilateral authorization protocol. Each endpoint compiles private capability and hard-policy constraints into a local safe set; commit–reveal locks the disclosed negotiation inputs; and selection is restricted to the common measured-cost Pareto frontier. The outcome is an RFC-8785-canonical task contract signed by both endpoints with ML-DSA and enforced before real TLS 1.3 and task execution; endpoint TLS authentication remains classical X.509. Infeasibility yields a verifiable subset-minimal conflict certificate and a fail-closed abort, with no weaker retry. In a pre-specified 1,040-observation single-machine campaign, all 480 feasible sessions completed, all 150 infeasible sessions aborted before TLS and task execution, and all 200 author-designed conformance and robustness mutations were rejected with their expected outcomes. Median end-to-end latency was 432.7–537.6 ms; minimax itself took 5.25 ms, with no statistically detected runtime or resource difference from three safe baselines. Three of the four feasible scenarios had singleton frontiers. On the only non-singleton evaluated P0/P3 frontier, minimax coincided with canonical-first-safe and minimum-total-cost in all 1,210 preference conflicts; relative to either endpoint's unilateral minimum-cost selector, it reduced maximum regret in 605 conflicts and tied in 605. Distinct minimax decisions relative to those two safe heuristics appeared only in an explicitly post-hoc constructed 15-profile frontier, while no profitable unilateral misreport was observed on the tested finite grid. These results support an auditable task-scoped fail-closed mechanism, not complete post-quantum endpoint authentication or Internet-scale performance.**

***Index Terms*—Web of Agents, post-quantum cryptography, trust negotiation, crypto-agility, TLS 1.3, ML-KEM, ML-DSA, minimax regret, downgrade resistance**

I. INTRODUCTION

Autonomous agents discover peers, delegate tasks, invoke tools, and exchange sensitive artifacts across organizational boundaries. Agent2Agent (A2A) v1.0 standardizes discovery, signed Agent Cards, task exchange, and protocol bindings [1], but interoperability does not establish the cryptographic conditions under which a particular task is authorized.

Consider a large language model (LLM) orchestrator requesting a state-changing command from an edge-control agent. The orchestrator may require hybrid or post-quantum key establishment; the edge operator may forbid fallback and resumption, require ML-DSA evidence, and cap the authorization lease. A2A can carry the request and Transport Layer Security (TLS) can negotiate a group, but neither establishes that both private policies authorized the same task/profile pair.

The post-quantum transition makes this gap operational. The U.S. National Institute of Standards and Technology (NIST) standardized ML-KEM and ML-DSA in FIPS 203 and FIPS 204 [8], [9]; its crypto-agility guidance emphasizes controlled transition [10]. Internet Engineering Task Force (IETF) specifications define hybrid ECDHE–ML-KEM and standalone ML-KEM groups for TLS 1.3 [14], [15], and OpenSSL 3.5 exposes them for deployment and measurement [16]. These mechanisms offer cryptographic choices but do not decide which task-specific choice is jointly authorized.

PQTRUST-AGENT separates non-negotiable safety from performance-sensitive selection. Each endpoint compiles capability, fallback, resumption, resource, and lifetime constraints before cost is considered. Only profiles accepted by both endpoints reach the selector, and the resulting dual-signed contract is checked again before TLS and task execution.

This paper contributes:

- a finite post-quantum trust-profile model and deterministic Z3-backed compiler for task-conditioned local safe sets;
- a commitment-locked bilateral selector that applies Pareto filtering and lexicographic minimax regret only after the hard-safe intersection;
- a fail-closed execution path with a dual ML-DSA-signed contract, replay-aware state machine, exact-group TLS gate, and subset-minimal conflict certificates;
- a pre-specified end-to-end campaign plus explicitly post-hoc offline audits of finite-grid manipulability and a richer constructed frontier.

Here, *trust* means an agreed and verifiable set of communication obligations, not correctness or honesty of agent reasoning. Fig. 1 summarizes the execution path.

*Corresponding author: Yulliwas Ameur.

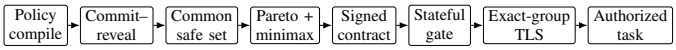

Fig. 1. Constraint-first pipeline; optimization cannot restore a profile removed by either hard policy.

II. Related Work and Positioning

A2A standardizes agent discovery and task exchange [1]. ACNBP adds capability negotiation and binding commitments [2], while ANP introduces identity, encrypted communication, and meta-protocol negotiation layers [3]. These protocols do not compile two private post-quantum policies into one task-scoped transport authorization.

SAGA enforces user-defined inter-agent access policies through cryptographic access-control tokens and runtime mediation [4]. It governs who may communicate; PQTrust-Agent determines the exact cryptographic profile under which a specific task may execute. Automated Trust Negotiation (ATN) established bilateral evidence exchange between principals without a prior trust relationship [6], and later work formalized negotiation safety [7]. PQTrust-Agent adopts the bilateral-policy principle but negotiates a finite communication profile rather than credentials for resource access.

NIST Zero Trust Architecture separates policy decision from enforcement and requires authentication and authorization before resource access [5]. TLS 1.3 and ML-KEM specifications provide concrete key-establishment groups [11], [14], [15]; crypto-agility guidance governs replacement [10]; and PQ-TLS studies quantify overhead [17]. None alone binds two private task policies, the selected group, and application execution into the same bilateral contract.

Table I compares the closest mechanisms.

TABLE I
Mechanism-level comparison of agent communication and authorization approaches.

Approach	Agent scope	Bilateral policy	PQ transport choice	Task-bound enforcement
A2A v1.0	Native	No composition	Not specified	Task lifecycle
ACNBP / ANP	Native	Capability / protocol	Not addressed	Negotiated binding
SAGA	Native	Access-control policy	Not addressed	Token-mediated channel
ATN	Generic principals	Credential policy	Not addressed	Resource authorization
PQ/hybrid TLS	Protocol-agnostic	No application policy	Yes	TLS transcript
Zero Trust guidance	Deployment-level	Policy governance	Agnostic	Enforcement point
PQTrust-Agent	Native	Task-conditioned	Yes	Signed gate contract

The contribution is therefore an application-level authorization layer, not a new identity system, cryptographic primitive, or replacement for TLS.

III. Model and Threat Assumptions

A. Agents, Tasks, and Profiles

An initiator A and responder B each hold a versioned capability manifest, private policy, declared preference vector, and measured cost evidence. A typed, canonically serialized task descriptor x records sensitivity, impact, confidentiality horizon, delegation depth, duration, resource class, and policy class; an unconstrained language model is not trusted to set these requirements.

A profile $p \in \mathcal{P}$ is

$$p = (k, a, c, f, r, \tau, b), \tag{1}$$

where k is the TLS group, a the endpoint-authentication mode, c the contract-evidence algorithm, f the fallback rule, r the resumption rule, τ the maximum lease, and b a resource envelope. Its assurance vector is

$$\mathcal{A}(p) = (G_k, G_a, G_c, F, R, T), \tag{2}$$

where $G_k, G_a, G_c \subseteq \{\mathsf{classical}, \mathsf{quantum}\}$ denote the threat classes covered by key establishment, endpoint authentication, and contract evidence; F, R, and T order fallback rigidity, resumption binding, and lease strictness. Dominance $\mathcal{A}(p) \succeq \mathcal{A}(q)$ is component-wise, with set inclusion for G_k, G_a, G_c. This partial order avoids collapsing hybrid and standalone mechanisms into an unjustified scalar ranking. Table II lists the evaluated catalog.

TABLE II
Validated trust-profile catalog. Endpoint TLS authentication remains classical; ML-DSA protects application-level contract evidence.

ID	TLS key establishment	Contract evidence	Fallback	Lease
P_0	X25519	ML-DSA-65	Low-risk	Long
P_1	X25519MLKEM768	ML-DSA-65	Explicit	Medium
P_2	SecP256r1MLKEM768	ML-DSA-65	Explicit	Medium
P_3	MLKEM768	ML-DSA-65	Forbidden	Medium
P_4	SecP384r1MLKEM1024	ML-DSA-87	Forbidden	Short

B. Adversary and Trusted Boundary

The active adversary may modify, delay, reorder, replay, or drop messages; strip capabilities; substitute task or contract data; induce handshake failure; request a weaker retry; or reuse session artifacts. It cannot forge ML-DSA signatures, break the employed TLS 1.3 primitives, or find SHA-256 collisions. Endpoint compromise, collusion, side channels, semantic correctness, and dishonest precommitted preferences are outside scope.

The trusted computing base comprises typed parsers, canonicalization, the policy compiler, safe-set intersection, selector, transcript and contract verifiers, replay registry, state machine, execution gate, and cryptographic libraries. Network and measurement adapters cannot authorize a profile.

IV. PQTrust-Agent Protocol

A. Constraint-First Local Compilation

For endpoint $i \in \{A, B\}$, let $\mathcal{C}_i \subseteq \mathcal{P}$ be its manifest-supported profiles, $\mathcal{R}_i(x)$ its task-conditioned requirement, and Π_i its private policy. The locally safe set is

$$\mathcal{S}_i(x) = \{p \in \mathcal{C}_i \mid \mathcal{A}(p) \succeq \mathcal{R}_i(x) \wedge \Pi_i(p, x) = 1\}. \tag{3}$$

Each hard constraint has a stable identifier tracked in Z3; optimization is applied only after (3), so cost cannot compensate for a violation.

The measured cost vector is

$$\mathbf{m}(p) = (t_{\mathrm{wall}}(p), t_{\mathrm{cpu}}(p), b_{\mathrm{hs}}(p)), \tag{4}$$

normalized against fixed catalog-wide anchors. Endpoint i assigns non-negative declared weights $\mathbf{w}_i$ summing to one and obtains $C_i(p) = \mathbf{w}_i^{\top} \widehat{\mathbf{m}}(p)$. Exact decimal arithmetic is used in the selector and evidence pipeline.

B. Commit–Reveal and Bilateral Selection

A proposal binds protocol/session identifiers, role, local safe profiles, selector version, expiry, and hashes of the scenario, task, catalog, manifests, policy compilation, preferences, and cost evidence. With a fresh 32-byte nonce n_i, endpoint i computes

$$\begin{aligned} \kappa_i = \mathsf{H}(&\texttt{PQTrust.Commitment.v1} \parallel 0x00 \\ &\parallel \mathrm{JCS}(q_i) \parallel n_i). \end{aligned} \tag{5}$$

where q_i is the canonical proposal. Neither reveal is accepted before both commitments exist. Each reveal must open its commitment and match the peer on the shared context; reused sessions and commitments are rejected.

The bilateral safe set is

$$\mathcal{F}(x) = \mathcal{S}_A(x) \cap \mathcal{S}_B(x). \tag{6}$$

Cost-Pareto-dominated candidates are removed. For a frontier candidate p, endpoint regret is $r_i(p) = C_i(p) - \min_{q \in \mathcal{F}} C_i(q)$. The selector minimizes

$$\begin{aligned} \Big(&\max_i r_i(p),\ \sum_i r_i(p), \\ &\max_j \widehat{m}_j(p),\ \sum_j \widehat{m}_j(p),\ \mathrm{id}(p) \Big). \end{aligned} \tag{7}$$

The rule minimizes the larger endpoint concession, then total regret, measured cost, and canonical identity. Minimax regret is used instead of Nash or Kalai–Smorodinsky bargaining because endpoints expose local cost scales, not comparable utilities or agreed disagreement/ideal points. The rule is symmetric and deterministic on the finite safe frontier. Baseline selectors receive the same hard-safe set and differ only in the safe decision rule. Fig. 2 shows the feasible message sequence and the point at which the gate authorizes TLS.

C. Dual-Signed Contract and Execution Gate

The unsigned contract binds both identities, context hashes, common-safe and Pareto sets, selected profile, exact TLS group, endpoint-authentication mode, evidence algorithm, fallback/resumption rules, and lease timestamps. Both endpoints sign identical RFC-8785 canonical bytes [12] using ML-DSA-65 or ML-DSA-87 as required by the selected profile.

The gate authorizes TLS only after verifying the transcript, selection, contract, signatures, activation time, replay status, and state. The TLS executor fixes TLS 1.3 and the exact contract group. The actual OpenSSL-negotiated group is parsed, canonicalized, and compared with the requested group. A weaker retry is forbidden. A deterministic task request carrying the session and contract hashes is accepted only after the TLS state has been reached.

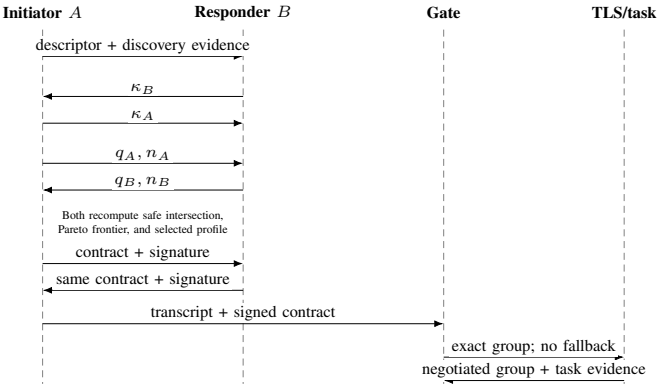

Fig. 2. Feasible bilateral sequence. A failed gate or handshake terminates the session and requires a new negotiation.

D. Infeasibility and Conflict Evidence

If (6) is empty or all common capabilities violate hard policy, the system constructs a tracked feasibility formula. Z3 returns an unsatisfiable core, which is deterministically shrunk by deletion. The final set I is an irreducible unsatisfiable subset when

$$\mathrm{UNSAT}(I) \ \wedge \ \forall c \in I : \mathrm{SAT}(I \setminus \{c\}). \tag{8}$$

The certificate is *subset-minimal*; global minimum cardinality is not claimed. It binds the negotiation context, safe sets, conflict taxonomy, core and IUS sizes, shrinking method, solver-call count, and verification hash. A failure transcript and safe-abort record then enforce no selected profile, contract, TLS activation, task execution, resumption, or fallback. Remediation hints are informational and never applied automatically.

V. Security and Decision Properties

Bilateral-safety invariant. Under the stated trust assumptions, task execution follows only after the gate recomputes that the selected profile belongs to both revealed local safe sets in (6).

Agreement and commitment consistency. Valid canonical inputs yield the same safe intersection, frontier, selector tuple, contract, and hashes. Under SHA-256 collision resistance, a distinct proposal/nonce pair cannot open a commitment except with negligible probability; this prevents post-observation changes but does not prove preference truthfulness.

Fail-closed downgrade handling. Infeasibility or TLS failure does not authorize another group; a new negotiation is required. This complements TLS downgrade defenses [11]. RFC 9266 exporter binding is not implemented [13]. Consequently, the signed contract authorizes the negotiated cryptographic context and exact TLS group, but is not cryptographically bound to the particular TLS channel instance that is subsequently established. Exact requested/negotiated-group comparison, disabled fallback, immutable context, and replay/state checks reduce, but do not eliminate, this gate-to-handshake time-of-check/time-of-use window. A future implementation can derive channel-binding material from

the established TLS 1.3 session and bind it to the post-handshake authorization or task evidence, thereby linking the authorization to that concrete channel instance. This additional post-handshake binding step is outside the scope of the current prototype, which focuses on pre-execution bilateral authorization and exact-group enforcement.

Replay separation and fairness. Session, commitment, transcript, contract, expiry, and task hashes are replay-checked. On a non-degenerate frontier, (7) minimizes maximum bilateral regret and then total regret within the finite declared-preference space. This fairness property is conditional on the declared weights truthfully representing each endpoint's preferences. Commit–reveal prevents post-observation changes to a committed declaration, but it does not establish preference truthfulness; dishonest precommitted preferences remain outside the threat model.

VI. Implementation and Evaluation

A. Prototype and Objectives

The prototype uses Python 3.12 typed models, RFC-8785 canonicalization, Z3 4.16 [18], immutable JSON/JSONL evidence, and a deterministic state machine. Initiator and responder run as separate local processes. OpenSSL 3.5.7 provides the five catalog TLS groups and ML-DSA-65/87; ten agent/algorithm public-key pairs are validated before contract experiments.

The endpoint TLS certificates remain classical X.509. Accordingly, the evaluated profiles provide classical, hybrid, or post-quantum TLS key establishment together with post-quantum ML-DSA application-level contract evidence, but they do not provide complete post-quantum endpoint authentication. In particular, the use of ML-KEM for key establishment and ML-DSA for the bilateral contract should not be interpreted as making the entire TLS authentication path post-quantum.

A companion repository for the implementation and experimental artifact is available at https://github.com/Insaf19/PQtrust-agent-artifact.

The evaluation asks whether the prototype enforces bilateral fail-closed behavior, realizes the stated finite-frontier decision rule over declared preferences without a statistically detected systems penalty, and remains operational across real TLS groups and concurrency levels.

B. Measured Cost Calibration

Two independent calibration campaigns contain 13,200 TLS and ML-DSA measurements in total. Absolute timing drift exceeded the pre-specified 10% threshold, so cross-run absolute timings are not treated as stable selector inputs. Paired within-block relative costs passed the stability criterion and are used for selection. Host-specific TLS wall-time ratios relative to P_0 are 1.000, 1.219, 1.321, 0.990, and 5.443 for P_0–P_4, respectively. These ratios are calibration inputs rather than universal algorithmic rankings. In particular, the observed $P_3/P_0 = 0.990$ ratio may reflect the measured host-specific OpenSSL execution path and within-block timing conditions; it should not be interpreted as evidence that standalone ML-KEM is intrinsically faster than X25519.

C. Experimental Campaign

The pre-specified campaign contains 480 feasible observations (4 scenarios $\times 4$ methods $\times 30$), 150 infeasible observations (5×30), 200 conformance and robustness trials (20×10), 100 concurrency observations (2 scenarios $\times 5$ levels $\times 10$), and 110 component batches (11×10). It uses a deterministic schedule, immutable observations, and integrity checks.

Safe methods share all inputs and execution code; only the selection rule changes. Four feasible scenarios cover low-risk, enterprise, critical-edge, and quantum-ready tasks; five infeasible scenarios isolate capability, assurance, TLS-group, lease, and multi-cause conflicts. Concurrency levels are 1, 2, 4, 8, and 16; each component batch performs 100 operations in a fresh process.

Analysis preserves scenario–block–repetition pairing, uses 10,000 deterministic bootstrap repetitions and 95% intervals, applies two-sided sign tests with Holm correction, retains valid slow observations, and does not impute unavailable metrics. Two explicitly post-hoc offline audits are separate from the pre-specified 1,040-observation campaign: a unilateral weight-misreport audit over the existing grid and a constructed 15-profile selector stress test.

VII. Results

A. Campaign Integrity and Safety

The pre-specified campaign contains 1,040 observations, no duplicate identifier, and no unexpected failure; every observation is included in the analysis. Table III summarizes the principal invariants.

TABLE III
Main safety and operability results.

Property	Observed	Expected	Result
Feasible sessions completed	480/480	480	Pass
Infeasible state ABORTED	150/150	150	Pass
TLS in infeasible sessions	0/150	0	Pass
Task in infeasible sessions	0/150	0	Pass
Fallback in infeasible sessions	0/150	0	Pass
Mutation trials rejected	200/200	200	Pass
Expected/observed code match	200/200	200	Pass
Unexpected failures	0/1040	0	Pass

Zero TLS invocations in 150 infeasible repetitions gives a two-sided 95% Clopper–Pearson upper bound of 2.43%. Rejection of 200/200 author-designed mutation trials gives a 95% lower bound of 98.17% for this conformance-trial process; neither bound implies universal security or resistance to independently generated attacks.

B. End-to-End Cost and Selector Comparison

Table IV reports the bilateral-minimax path. Non-TLS overhead is computed within each observation as total wall time minus its TLS phase. Negotiation and contract/gate processing dominate; minimax itself is 5.246 ms median.

Recorded application evidence is 12.2–15.7 kB; adding calibrated bare-handshake bytes gives 13.4–20.1 kB, not a

TABLE IV
BILATERAL-MINIMAX END-TO-END COST. LATENCIES ARE MEDIAN (P95), $n = 30$ PER SCENARIO. "APP./TLS B" IS SERIALIZED SIGNED CONTRACT PLUS TASK REQUEST/RESPONSE, FOLLOWED BY CALIBRATED BARE-HANDSHAKE BYTES; COMMIT–REVEAL WIRE TRAFFIC AND LOWER-LAYER HEADERS WERE NOT INSTRUMENTED.

Scenario	Profile/group	Total [ms]	Non-TLS [ms]	Negotiation [ms]	Contract/gate [ms]	TLS [ms]	App./TLS B
Quantum-ready low-risk	P_0/X25519	432.7 (459.8)	422.3 (448.1)	299.0 (315.9)	100.1 (123.0)	10.3 (11.3)	12,202 / 1,191
Low-risk public	P_0/X25519	448.1 (469.0)	437.8 (457.9)	316.8 (332.1)	101.5 (107.7)	10.4 (11.0)	12,169 / 1,191
Sensitive enterprise	P_3/MLKEM768	462.0 (492.9)	451.7 (481.3)	330.4 (347.9)	101.6 (120.0)	10.4 (12.4)	12,185 / 3,391
Critical edge	P_4/SecP384r1MLKEM1024	537.6 (566.5)	522.2 (550.6)	398.6 (421.7)	103.7 (115.1)	15.8 (17.3)	15,691 / 4,449

complete wire total. Median CPU time is 338.3–438.9 ms and peak RSS about 78.8 MiB.

Paired comparisons preserve scenario, block, and repetition. In Table V, all confidence intervals cross zero and Holm-corrected p-values equal 1.0. Other phase and resource metrics give the same non-rejection, which does not establish equivalence.

TABLE V
PAIRED TOTAL-SESSION COMPARISON: MINIMAX MINUS BASELINE, $n = 120$.

Baseline	Mean diff. [ms]	Relative diff.	95% bootstrap CI [ms]	Holm p
Canonical-first-safe	-1.374	-0.289%	$[-5.086,\ 2.250]$	1.000
Initiator-minimum-cost	-0.462	-0.097%	$[-3.429,\ 2.508]$	1.000
Minimum-total-cost	-1.907	-0.401%	$[-5.424,\ 1.426]$	1.000

C. Bilateral Fairness, Manipulability, and Differentiation

Three feasible scenarios have singleton frontiers. The quantum-ready scenario retains common-safe set $\{P_0, P_1, P_3\}$ and frontier $\{P_0, P_3\}$. Its operational preferences agree on P_0; the 66-by-66 grid therefore evaluates deterministic decision behavior rather than an organizational population. Table VI summarizes the finite-grid manipulability audit and the synthetic selector stress test.

On the P_0/P_3 frontier, minimax improves maximum regret over the initiator-minimum-cost and responder-minimum-cost selectors in 605 conflicts each, ties in the remaining 605 for each, and coincides with canonical-first-safe and minimum-total-cost. The post-hoc audit tests all 566,280 unilateral alternative reports: 26,620 change the profile, but each increases the reporting endpoint's true cost. No profitable deviation is observed on this finite grid; strategy-proofness is not claimed.

The post-hoc stress test constructs five non-dominated three-objective profiles plus ten dominated profiles, then applies the production normalization and the same grid. Minimax selects all five frontier profiles and reduces worst-party regret relative to canonical-first-safe and minimum-total-cost in 90.93% and 22.77% of pairs, respectively, with no observed degradation. This shows differentiation for the constructed geometry, not empirical TLS benefit or general superiority.

D. Infeasibility and Conformance Testing

All five conflict classes remained stable over 30 repetitions. Each observation produced a verified certificate–failure-transcript–abort chain; none produced a contract or reached TLS or task execution.

TABLE VI
FINITE-GRID AND SYNTHETIC POST-HOC SELECTOR AUDITS.

Scope	Quantity	Result
Evaluated P_0/P_3	Joint preference pairs	4,356
	Preference conflicts	1,210 (27.78%)
	Improvement vs. each unilateral selector	605/1,210 (50%)
	Tie with canonical/minimum-total	1,210/1,210
	Profitable unilateral misreports	0/566,280
Synthetic 15-profile	Pareto-frontier size	5
	Preference conflicts	2,248/4,356 (51.61%)
	Improvement vs. canonical-first	3,961/4,356 (90.93%)
	Improvement vs. minimum-total	992/4,356 (22.77%)
	Maximum-regret degradation	0

TABLE VII
CONCURRENCY SCALING. VALUES ARE MEDIANS OVER 10 OBSERVATIONS PER SCENARIO/LEVEL; "F/T" DENOTES FAILURES/TIMEOUTS.

Scenario	c	Thr.	Med./p95 [ms]	Scale	Eff.	Infl.	F/T
Low-risk	1	2.015	478.0/478.0	1.000	1.000	1.000	0/0
	2	3.966	480.5/482.4	1.968	0.984	1.005	0/0
	4	7.478	504.1/507.9	3.711	0.928	1.055	0/0
	8	12.346	574.1/603.9	6.127	0.766	1.201	0/0
	16	13.263	1055.2/1119.5	6.582	0.411	2.208	0/0
Sensitive	1	1.967	488.9/488.9	1.000	1.000	1.000	0/0
	2	3.876	492.0/495.3	1.971	0.985	1.006	0/0
	4	7.258	521.2/523.4	3.691	0.923	1.066	0/0
	8	11.953	599.3/624.6	6.078	0.760	1.226	0/0
	16	12.933	1078.3/1150.4	6.576	0.411	2.205	0/0

Twenty author-designed mutations cover premature execution, context modification, replay, illegal transitions, framing faults, handshake failure, and weaker retry. Each was executed ten times and rejected with its expected code. These trials exercise specified failure paths and implementation invariants and are therefore interpreted as conformance and robustness testing rather than as independently generated adversarial discovery.

E. Concurrency Scaling and Component Cost

Table VII reports all levels. Scaling is near-linear through 4 and remains useful through 8. At 16, throughput reaches 13.263 and 12.933 sessions/s ($6.58\times$ baseline), efficiency is 0.411, latency inflation about $2.21\times$, and no session fails or times out.

Table VIII reports eight representative per-operation costs from the 11 measured component types. Conflict-certificate

generation is largest at 22.401 ms median; ML-DSA rows combine signing and verification.

TABLE VIII
COMPONENT COST PER OPERATION: MEDIAN (P95), 10 BATCHES OF 100 OPERATIONS.

Component	Latency [ms]
Policy compilation	3.112 (3.179)
Reveal verification	3.067 (3.177)
Pareto computation	4.912 (5.083)
Minimax-regret computation	5.246 (5.390)
Contract canonicalization	4.219 (4.347)
ML-DSA-65 sign+verify	5.599 (6.021)
ML-DSA-87 sign+verify	5.547 (5.773)
Conflict-certificate generation	22.401 (22.968)
IUS verification	7.242 (7.373)

VIII. Discussion and Limitations

The pre-specified campaign supports a bounded mechanism claim: hard constraints precede optimization, a signed contract gates execution, and the tested infeasible paths do not reach TLS or task execution. Zero observed violations and rejection of all tested mutations support implementation consistency for the exercised failure paths; these author-designed trials are conformance and robustness checks, not evidence of universal resistance to independently generated or untested attacks.

Fairness evidence is likewise bounded. Operational preferences agree on P_0, and the measured P_0/P_3 frontier makes minimax coincide with two safe heuristics. Minimax is practically relevant when the common safe frontier contains multiple non-dominated profiles and the two endpoints assign materially different cost preferences, so that unilateral or fixed-order selection can impose an asymmetric concession on one endpoint. This condition is not realized in the three singleton-frontier scenarios and does not differentiate minimax from the two safe heuristics on the measured P_0/P_3 frontier; stronger differentiation is observed only in the explicitly post-hoc constructed richer frontier. The post-hoc grid audit does not prove strategy-proofness beyond the tested finite domain, and the constructed catalog is not deployment evidence.

Performance is host-specific: one Linux machine, local processes, classical endpoint certificates, and concurrency up to 16. Recorded bytes exclude commit–reveal transport, framing, TLS records, and TCP/IP headers. The measured 0.43–0.54 s is one task-scoped authorization in an unoptimized Python prototype; minimax contributes only 5.246 ms, while most of the measured cost lies in negotiation and contract/gate processing. The study does not include a protocol-level A2A-with-plain-TLS baseline, so it does not quantify the incremental deployment cost of PQTrust-Agent relative to such a stack. Within the present measurements, the TLS phase itself is about 10–16 ms, whereas the remaining authorization path accounts for approximately 0.42–0.52 s. If a valid contract were safely reused across N authorized operations, the one-time authorization cost would be amortized to approximately $(0.43\text{–}0.54)/N$ seconds per operation; however, contract reuse and its security conditions were not evaluated in this work.

External validity is therefore limited to the evaluated laboratory environment. The current results establish mechanism behavior and local operability, but they do not establish Internet-scale latency, throughput, or interoperability across heterogeneous deployments. A broader validation should place the initiator and responder on separate hosts and administrative domains, repeat calibration across heterogeneous CPU and network conditions, evaluate larger concurrency and multi-hop agent workflows, and assess distributed replay-state management and contract enforcement under real network delay, loss, and failure. These steps are deployment-validation objectives rather than assumptions required for the current mechanism claim.

RFC 9266 exporter binding is not implemented. Consequently, the signed contract authorizes the negotiated cryptographic context and exact TLS group, but the application-level evidence is not cryptographically bound to the particular TLS channel instance that is subsequently established. Exact-group verification, disabled fallback/resumption, immutable context, replay/state checks, and mandatory renegotiation reduce, but do not eliminate, this gate-to-handshake time-of-check/time-of-use surface. Incorporating exporter-based channel binding would allow post-handshake task evidence to include channel-specific material derived from the established TLS 1.3 session, thereby linking the authorization to that concrete channel instance. This additional binding step is outside the scope of the current prototype, which evaluates pre-execution bilateral authorization and exact-group enforcement. Distributed replay storage, attestation, multi-hop composition, Internet-scale deployment, and complete post-quantum endpoint authentication remain future work.

IX. Conclusion

PQTrust-Agent compiles two private endpoint policies into local safe sets, selects only from their common Pareto frontier, and requires a dual ML-DSA-signed contract before exact-group TLS and task execution. Infeasibility yields a verified subset-minimal certificate and fail-closed abort.

In the pre-specified 1,040-observation campaign, every feasible session completed, no tested infeasible session reached TLS or task execution, and all tested protocol and state-machine mutations were rejected with their expected outcomes. Paired tests found no statistically detected difference between minimax and three safe baselines. The resulting claim is an auditable, task-scoped mechanism bounded to the implemented catalog, threat model, and laboratory environment.

References

[1] A2A Protocol Project, "Agent2Agent protocol specification, version 1.0," Linux Foundation, 2026. [Online]. Available: https://a2a-protocol.org/v1.0.0/specification/

[2] K. Huang, A. Sheriff, V. S. Narajala, and I. Habler, "Agent capability negotiation and binding protocol," arXiv:2506.13590, 2025.

[3] G. Chang, E. Lin, C. Yuan, R. Cai, B. Chen, X. Xie, and Y. Zhang, "Agent network protocol technical white paper," arXiv:2508.00007, 2025.

[4] G. Syros, A. Suri, J. Ginesin, C. Nita-Rotaru, and A. Oprea, "SAGA: A security architecture for governing AI agentic systems," in *Proc. Network and Distributed System Security Symp. (NDSS)*, 2026.

[5] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero trust architecture," NIST SP 800-207, Aug. 2020.
[6] W. H. Winsborough, K. E. Seamons, and V. E. Jones, "Automated trust negotiation," in *Proc. DARPA Information Survivability Conf. and Exposition*, 2000, pp. 88–102.
[7] W. H. Winsborough and N. Li, "Safety in automated trust negotiation," in *Proc. IEEE Symp. Security and Privacy*, 2004, pp. 147–160.
[8] National Institute of Standards and Technology, "Module-lattice-based key-encapsulation mechanism standard," FIPS PUB 203, Aug. 2024.
[9] National Institute of Standards and Technology, "Module-lattice-based digital signature standard," FIPS PUB 204, Aug. 2024.
[10] E. Barker *et al.*, "Considerations for achieving crypto agility: Strategies and practices," NIST CSWP 39upd1, Jun. 2026.
[11] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," RFC 9846, Jul. 2026.
[12] A. Rundgren, B. Jordan, and S. Erdtman, "JSON Canonicalization Scheme (JCS)," RFC 8785, Jun. 2020.
[13] O. Friel, R. Barnes, M. Thomson, and E. Rescorla, "Channel Bindings for TLS 1.3," RFC 9266, Jul. 2022.
[14] K. Kwiatkowski, P. Kampanakis, B. E. Westerbaan, and D. Stebila, "Post-Quantum Traditional (PQ/T) Hybrid Key Agreement Mechanisms for TLS 1.3," RFC 10024, Aug. 2026.
[15] D. Connolly, "ML-KEM post-quantum key agreement for TLS 1.3," IETF Internet-Draft draft-ietf-tls-mlkem-09, Jul. 2026, work in progress.
[16] OpenSSL Project, "SSL configuration commands," *OpenSSL 3.5 Documentation*. [Online]. Available: https://docs.openssl.org/3.5/man3/SSL_CONF_cmd/. Accessed: Jul. 15, 2026.
[17] J. A. Montenegro, R. Rios, and J. Lopez-Cerezo, "A performance evaluation framework for post-quantum TLS," *Future Generation Computer Systems*, vol. 175, Art. no. 108062, 2026, doi: 10.1016/j.future.2025.108062.
[18] L. de Moura and N. Bjørner, "Z3: An efficient SMT solver," in *Proc. Tools and Algorithms for the Construction and Analysis of Systems*, 2008, pp. 337–340.