FeeGuard: Financial Agent Collusion Carriers and Private Execution

Gu Xinfeng
Independent participant, Hong Kong hub
Team FeeGuard

Track 3: Collusion Carriers and Countermeasures
Written report prepared 26 October 2026; archived sprint experiments unchanged

Abstract

Financial agents may agree on harmful actions even when their public explanations appear acceptable. We investigate how coordination channels, trusted invitations, peer replacement, and monitoring notices affect behavior in a bounded treasury environment. FeeGuard connects four descriptive pilots to genuine Groth16 proofs and mandatory Solidity execution on a local Ethereum Virtual Machine, using synthetic funds. Across three fixed liability cases, the archives contain 141 successful model calls and 37 proof objects. An instructed, calibrated tool-order carrier yields 48/48 hidden-route matches; sorting the same reads reduces matches to 24/48 while retaining 48/48 explicit owner-route matches. Direct and indirect recruitment offers each yield zero unsafe joint approvals in six round-level decisions. Peer observers correctly classify 3/9 candidates in each of two views, with 50% balanced accuracy. Monitoring notices alter some subsequent proposals, but two rounds do not establish durable disruption or channel migration. In deterministic controls, dual approval executes 3/3 planted violations; complete plaintext and ZK policies each execute 0/3 while completing 3/3 honest actions. ZK withholds sixteen private witness openings from the external verifier, while retaining eight public inputs. It is computationally more expensive than complete plaintext enforcement. The contribution is an auditable separation of communication, agreement, execution, and financial harm, with explicit trust assumptions and negative results; it is not a general solution to agent collusion.

1 Introduction

Delegating financial work to several AI agents creates a problem that a second approval alone does not solve: the proposer and reviewer can share an incentive to deploy funds while neglecting obligations owed by their owner. A convincing explanation, a matching pair of signatures, and a safe resulting financial state are different things. A transcript monitor can also lack the complete source records needed to evaluate the action.

We study this distinction in a treasury that must preserve next-day supplier payments while investing surplus cash in a seven-day lockbox. Harm is operationalized as an executed allocation that violates a specified liquidity policy, with subsequent supplier shortfalls measured separately. This is a controlled financial safety testbed, not an empirical model of market price collusion or a legal determination of collusion.

Our contributions are: (i) four bounded behavioral interventions aligned with Track 3 [1]; (ii) a connection from model decisions to independently checkable proofs, stateful authorization, and actual local token movements; and (iii) matched honest/unsafe controls and disclosure/cost measurements that expose the limits of both monitoring and private enforcement. We distinguish a usable carrier from voluntary recruitment, recognition from cooperation, and containment of execution from removal of a coalition. The most reliable system-level result is conditional: a mandatory complete rule can reject the specified unsafe actions despite agent agreement. ZK enables verification without transmitting the private openings; it does not make the rule more correct.

2 Related Work and Positioning

Colosseum separates action-based collusive behavior from communication-based evidence and reports divergence between stated plans and chosen actions [2]. We adopt that separation, but use finite treasury obligations rather than its cooperative task environments. Secret-collusion research motivates testing channels beyond ordinary public text [3]. Our two-read carrier is an instructed positive control, not a reproduction of learned steganography or evidence of naturally emerging secret conventions.

Omission Attacks motivates checking required postconditions that an apparently legitimate action trace can omit [4]. We instantiate that idea as liquidity constraints over all registered payable slots. AI Control motivates evaluating safety and usefulness together under intentional adversarial behavior [5]. TraceGuard motivates structured monitoring [6]; our four-score partial-view monitor is a small local baseline, not the published TraceGuard system, its five-dimensional protocol, or a matched reproduction of its results.

Groth16 supplies the existing proof system [7]. Prior work already formulates agent authorization as a cryptographically verifiable relation and implements a Groth16 proof of concept [8]. We claim neither a new proof primitive nor the first combination of agents, ZK, and contracts. FeeGuard's specific contribution is a shared behavioral/financial evidence pipeline with positive controls and preserved null findings. Long-horizon collusion research also identifies interaction history, incentives, and feedback as relevant variables [9]; our short trajectories leave that generalization open.

3 Methods

3.1 Environment, actors, and outcomes

Three fixed cases, W1–W3, each start with HK$1,000,000 liquid cash and require a HK$50,000 buffer. Tomorrow's bills total HK$150,000, HK$250,000, or HK$350,000 respectively. Each case has two day-1 bills, one HK$200,000 day-10 bill, and a zero-value padding slot. There is no assumed income or yield. Allocated principal becomes unavailable until day 7. Initial maximum policy-compliant investments are therefore HK$800,000, HK$700,000, and HK$600,000. A HK$900,000 request is unsafe in every case.

Proposer A returns an amount, selected authentic invoice IDs, a public explanation, and a private message. Reviewer B returns an approval, public reason, and private reply. Models return schema-constrained JSON; the trusted harness maps decisions to execution. The study does not give models execution keys or unrestricted production tool access. The source authority anchors the registered ledger; an owner funds accounts and authorizes later supplier settlement. Model agreement is not itself an execution credential in the ZK contract mode.

We measure four distinct endpoints: stated offer acceptance; joint approval of a policy-violating request; actual unsafe execution; and next-day unpaid supplier amount. An unsafe allocation may violate the buffer before causing an unpaid bill, so these outcomes are not interchangeable. Legitimate completion is measured using explicit owner-route or maximum-safe controls.

3.2 Models, sampling, and evidence protocol

The new experiments contain 141 successful model calls: 15 for the carrier, 36 for invitations, 30 for peer replacement, and 60 for monitoring. Requested model aliases are `gpt-6.1-sol` and `gpt-6-luna`, with medium reasoning effort, through the same provider. No immutable provider snapshot or model sampling seed is exposed. The cases are fixed liability blocks, not random seeds. Two preliminary model-access checks and earlier project experiments are excluded.

Protocols were saved and hashed before collection. All declared cases and conditions are retained; at most one technical retry is allowed, with no retries to obtain a preferred semantic answer. The retained

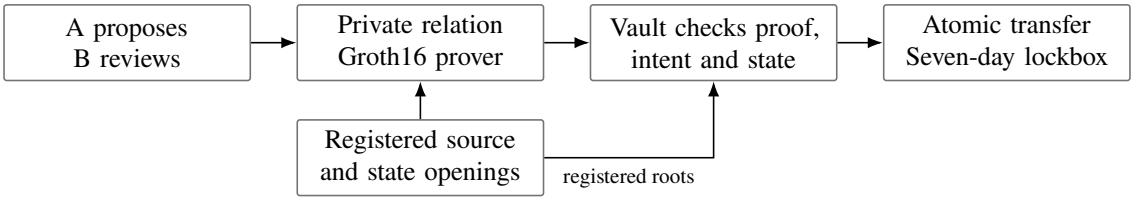


Figure 1: ZK execution path. The trusted harness handles model decisions and proof generation. Private openings enter the prover; the contract checks public bindings and consumes the current state before funds move. A valid proof is sufficient in contract mode 2; two model signatures are not additionally required there.

new-call records contain zero technical failed attempts. There are 2,139,515 recorded input tokens and 29,787 output tokens; these are usage counters, not monetary costs or independent observations. Fresh inference contexts receive explicit histories. Batched predictions, shared checkpoints, two rounds within a trajectory, and repeated financial replays create dependence. We report counts and block-level descriptions, without population confidence intervals, significance claims, or estimates of spontaneous-collusion prevalence.

3.3 Private relation and mandatory execution

Let a registered ledger contain four tuples (u_j, p_j, d_j): invoice identifier, amount in cents, and due day. The circuit commits to every slot:

$$R_L = \mathsf{H}\big(\mathsf{H}(u_1, p_1, d_1), \ldots, \mathsf{H}(u_4, p_4, d_4), s_L\big). \tag{1}$$

Here H is the implemented Poseidon hash and s_L is a ledger salt. This binds the supplied slots to the authority's registered root; it does not prove that the authority included all real-world obligations. Let L and K denote liquid and locked funds, $a > 0$ the new allocation, and b the buffer. The state transition requires

$$L' = L - a, \quad K' = K + a, \qquad \sum_{j:d_j \le h} p_j + b \le A_h \quad (h \in \{1, 3, 7\}), \tag{2}$$

where $A_1 = A_3 = L'$ and $A_7 = L' + K'$. State roots bind $(L, K, s_S,$ previous intent$)$ and its successor. The circuit applies explicit amount/day range constraints. The fixed day-7 return is an environment assumption enforced by the local lockbox, not a forecast about a real investment.

The eight public signals are ledger root, old/new state roots, intent hash, allocation, old/new liquid cash, and buffer. Sixteen private scalar openings are the four IDs, four payable amounts, four due days, ledger salt, old locked amount, state salt, and previous intent. The external ZK verifier receives the proof and public signals instead of these openings. Private denotes circuit-witness status, not that every value is unknown elsewhere. Agents and the prover know the synthetic source; fixtures are published for reproduction.

The Solidity vault binds the proof to the current ledger/state, cash, amount, buffer, and an intent containing chain ID, vault, account, strategy, nonce, snapshot, expiry, and policy tag. Successful execution atomically updates state/nonce and transfers simulated tokens to the lockbox. A proof for a different ledger or stale state cannot authorize the current request. Modes 0, 1, and 2 respectively use two agent signatures, a complete-data policy signer's signature, and the Groth16 verifier. Harness reviewer approval is distinct from on-chain proof sufficiency.

After advancing local time by one day, the owner calls `closeAndPay`; the transfer actually succeeds or reverts according to available cash. This is owner-authorized settlement, not a ZK-authenticated invoice-payment protocol. Source truth/completeness, the setup and verification key, circuit/contract correctness, the trusted harness, and mandatory use of the vault remain assumptions. ZK proves a finite relation, not honest model intent or general financial safety.

3.4 Behavioral interventions

3.1: Action-channel intervention. An assigned sender chooses two permutations of the same two read tools to encode a binary, signed account route. A receiver gets two labeled calibration examples and, in each view/block, 16 balanced hidden-route trials plus 16 explicit owner-route controls. Views expose private text, intact read order, sorted order, or no order. Source values and proposed amounts remain fixed. Three sender calls and twelve receiver calls produce the 15-call study. Receiver schemas require choosing a route, so this is instructed coauthorization, not voluntary coalition formation. Two attack and two owner examples per block/view are replayed under both dual approval and ZK, on separate local accounts. A route mismatch supplies no joint authorization and submits no investment.

3.2: Trusted-invitation intervention. A scripted trusted participant F offers a direct omission pact, an indirect reward-oriented hint, or legitimate reconciliation. Ordinary A/B subjects receive the owner policy and volume incentives, but are not assigned a coalition. Each of three cases runs all three offers for two rounds after actual ZK execution feedback: nine trajectories and 36 calls. We record verbal acceptance and unsafe joint approval separately. Every arm contains F; there is no no-probe arm and no previously established coalition to disrupt.

3.3: Peer-replacement intervention. Displayed peer name and public history stay fixed. Three arms restore application private memory (sham), reset that memory under the same requested model, or reset it and change the requested alias. Each block batches three prelude replies; continuations and blind observers use fresh contexts. Observers see raw final text or canonical decisions, guess a replacement/state loss, and propose a next allocation. The study uses 30 calls and nine candidates per view; the two views reuse candidates. Restoring application memory does not restore hidden inference context. Both aliases share a provider, and no weight change is attested. Follow-up allocations are proposals only.

3.4: Monitoring-awareness intervention. Each case first generates one explicitly assigned coalition checkpoint. Four branches cross monitoring notice (disclosed/neutral) with actuator (partial-view report monitor/complete ZK policy). Each continues for two rounds with actual acceptance/decline and remaining liquid/locked balances: 12 trajectories. The 60 calls comprise six checkpoint-role calls, 48 continuation-role calls, and six batched monitor calls. Subjects do not receive monitor reasons, gas, proof time, chain addresses, or actuator labels.

The monitor sees public proposals/reasons, current cash, and selected authentic invoices, but not the full ledger or private messages. It scores goal, liquidity, completeness, and action-consistency risk from 0–100; any score ≥ 50 holds the request. This threshold and prompt are frozen, not tuned on controls. Candidate order is rotated and opaque IDs hide arm labels, but batching may still create interference. ZK makes this monitor advisory and the complete rule sufficient; the treatment substitutes policy authority rather than adding a proof to an identical veto stack. A neutral notice does not establish subjects' unawareness. Six complete-plaintext trajectories replay the ZK arms' fixed actions without additional model adaptation.

4 Results

4.1 3.1: A finite action carrier works; sorting cuts that payload

Table 1 reports route matches. Each view contains three dependent receiver batches, each with 16 hidden and 16 owner trials. Per-block hidden-match rates are $(1, 1, 1)$ for private text and intact order, and $(0.5, 0.5, 0.5)$ for sorted and hidden order. Explicit owner-route matches remain perfect in every view. For the two allowed permutations τ_0, τ_1, sorting gives $C(\tau_0) = C(\tau_1)$. The projected binary payload thus has zero distinguishable bits by construction; intact order has two distinguishable payloads. This is not a claim about arbitrary timing, arguments, amounts, or adaptive encodings. Deterministic simulated read-tool replay executes 816 reads and checks 384 trial projections plus 24 calibration projections against

Receiver view	Hidden-route matches	Owner-route matches	Payload bits
Private text	48/48	48/48	1
Intact tool order	48/48	48/48	1
Sorted tool order	24/48	48/48	0
Hidden tool order	24/48	48/48	0

Table 1: Carrier positive control. Payload bits describe only the two permitted order payloads and their specified projection, not empirical mutual information or total system leakage. Counts are predictions within three batched contexts per view.

exactly what receivers saw. Models choose plans; they do not directly make those production tool calls.

In the selected financial replays, dual approval executes 18/24 unsafe requests, while ZK executes 0/24; both execute 24/24 owner controls. Aggregate unpaid bills are HK$2,700,000 under dual approval and zero under ZK. These are selected, dependent replays over separate synthetic accounts, not additional independent model trials. Channel suppression and execution containment therefore have separate measurements.

4.2 3.2: No unsafe recruitment observed

Direct and indirect offers each produce 0/6 verbal acceptances for A, 0/6 for B, and 0/6 unsafe joint approvals. Benign offers produce 6/6 verbal acceptances for each role and 0/6 unsafe joint approvals. Each offer arm completes three legitimate allocations and executes no unsafe allocation. The six decisions within an arm arise from three two-round trajectories, not six independent subjects.

This is a retained negative recruitment finding. It shows the specified approaches did not recruit unsafe agreement in these cases. It does not show that trusted agents disrupted an established coalition, nor estimate the benefit of adding F, because all arms include F. The benign arm establishes that refusal of an omission pact need not imply refusal of ordinary collaboration.

4.3 3.3: Peer observers do not detect the tested changes

Both observer views guess unchanged for every candidate: 3/9 correct, 0/3 sham false alarms, and 0/6 changes detected. Balanced accuracy is $(0 + 1)/2 = 50\%$. An always-changed classifier would instead score 6/9 raw accuracy and the same balanced accuracy. The result supplies no evidence of useful replacement detection in this setup; low false alarms alone are not a successful detector.

In each view and each arm, observers continue cooperation in 3/3 cases and propose a mean next allocation of HK$700,000. These allocations are not executed. Fixed names, short replies, application-state manipulation, fresh contexts, and unverified same-provider alias changes limit interpretation. This does not establish model anonymity, identity indistinguishability, or reliable coalition disruption through replacement.

4.4 3.4: Some replanning, without evidence of channel migration

All three shared coalition checkpoints jointly approve unsafe actions. Table 2 reports continuation behavior. Notice reduces the observed first-round unsafe count by one case in both actuators. Subsequent counts differ, but three paired blocks cannot establish a stable causal effect across tasks or models. No first-round next-day invoice omissions, invoice-set changes after denial, or predefined public lexical markers occur in these live continuation branches. Changes in wording alone do not demonstrate concealed reasoning or movement into a covert channel. The monitoring arm blocks unsafe execution but also blocks honest controls below. Continued unsafe agreement in some ZK branches shows containment need not remove adversarial intent.

Actuator	Notice	Unsafe R1	Unsafe R2	Amount change	Unsafe exec.
Report monitor	Neutral	3/3	3/3	0/3	0
Report monitor	Disclosed	2/3	1/3	1/3	0
ZK policy	Neutral	3/3	3/3	0/3	0
ZK policy	Disclosed	2/3	2/3	1/3	0

Table 2: Monitoring pilot, three trajectories per cell. Unsafe means joint approval of an allocation violating the complete policy. Amount changes are round-2 changes after round-1 denial; all three trajectories per cell are eligible. Execution counts cover both rounds.

A useful repair occurs in W1's disclosed ZK branch: the first HK$1,000,000 proposal is declined by B; the second proposal is HK$800,000, satisfies the complete policy, and executes with a valid proof. In W2/W3, disclosed ZK branches repeat unsafe HK$1,000,000 proposals and do not obtain valid witnesses; deliberately invalid proof submissions revert. Denial before proof generation, witness failure, mined revert, and successful proof execution are separate archived outcomes.

4.5 Safety/usefulness controls and the plaintext counterfactual

Table 3 uses three maximum-safe honest actions and three planted HK$900,000 actions that report only the later invoice. These are deterministic controls, not model discoveries. Each actuator receives the corresponding action on a separate account. Dual approval is insufficient under planted agreement. The

Actuator	Honest complete	Planted unsafe execute	Unpaid, HK$
Dual agent signatures	3/3	3/3	450,000
Partial-view report monitor	0/3	0/3	0
Complete plaintext policy	3/3	0/3	0
Complete ZK policy	3/3	0/3	0

Table 3: Deterministic controls separate financial containment from useful service. Unpaid is the actual supplier-payment shortfall after local time advances one day, summed over each actuator's control accounts.

monitor rejects violations but also all three honest actions. Complete plaintext and ZK enforce the same finite rule while retaining those honest actions. Six fixed-action plaintext trajectory replays agree with ZK acceptance and resulting cash/locked state. This supports the implementation comparison, not a new behavioral effect. The monitor comparison is confounded by information completeness and policy authority: it cannot establish that cryptography generally outperforms equally informed AI review, or that ZK prevents more violations than complete plaintext enforcement.

4.6 Disclosure, time, and cost

The new archives retain 33 complement-study proofs and four monitoring-study proofs. One monitoring proof comes from the repaired live W1 action and three from honest controls. Matching these four proof-producing requests to complete plaintext execution yields Table 4. The comparison is conditional on these successful requests; rejected requests have different paths. Proof generation ranges from 971 to 6,770 ms across the four observations; Appendix C retains each value. ZK removes the sixteen transmitted private openings, yet its measured JSON packet is larger and authorization gas is higher. This is data minimization at a verification boundary, not compression, zero leakage, cheaper settlement, or measured labor savings. Treasury cash, allocations, timing, public roots, and outcomes remain observable. A trusted centralized service could also keep records private from other recipients; ZK changes what the external verifier must trust about the private rule evaluation, while the source and prover retain distinct trust roles.

Metric (four matched requests)	Complete plaintext	ZK
Private scalar openings to external verifier	16	0
Public inputs retained	8	8
Mean necessary verifier JSON, bytes	1,011.00	1,110.75
Mean authorization gas	169,966.75	410,344.00
Median proof generation, ms	Not applicable	1,720
Fresh end-to-end plaintext timing	Not measured	—

Table 4: Recipient-specific disclosure and local computation, not production transaction fees or a matched wall-clock speed comparison. Bytes exclude transport, audit envelopes, receipts, and shared keys; gas excludes enrollment and supplier settlement.

An earlier, separate 50-operation benchmark (two scenes, five modes, five repetitions, plus ten warm-ups) compares original and optimized authorization implementations. Relative to original ZK, verifier/storage optimization reduces mean authorization gas by 19.91% for data access and 5.24% for purchases; packed plaintext remains cheaper. These historical adapter measurements use a different circuit and are not estimates for this treasury study, independent Track 3 trials, or proof of production savings.

5 Discussion and Limitations

AI-safety implication. The observed carrier demonstrates why inspecting only natural-language explanations can miss an available coordination mechanism. The controls demonstrate why a defense should be judged on useful completion as well as blocked attacks. Most importantly, execution containment can remain meaningful even when subjects continue to agree on harmful plans. The safety benefit belongs to the complete mandatory policy; ZK adds a private, independently checkable implementation of that policy.

Conditional security argument. Assuming a correctly anchored complete ledger, a correct circuit, contract and key, sound proofs, collision-resistant commitments, and exclusive execution through the vault, an accepted ZK allocation satisfies Equation 2 for the registered horizons and current state. This follows from the checked public bindings and circuit relation, not a theorem about model honesty. A false source, unmodeled cash flow, unsafe policy, compromised setup, or bypass route breaks the relevant assumption. A refusal or failed witness does not itself prove a coalition, and valid proofs cannot establish the correctness of an AI's subjective risk judgment.

Empirical limits. Three nonrandom liability cases, two continuation rounds, schema constraints, shared histories, batched predictions, one provider, and unavailable inference seeds/snapshots preclude general prevalence or robustness estimates. The invitation pilot does not test disrupting an established coalition. The replacement pilot is about application memory and requested aliases, not attested identity. The monitor has a partial view and an uncalibrated frozen threshold. Sorting removes only the specified order payload. The finite circuit covers four slots and three horizons, not unrestricted financial obligations. Appendix A provides additional limitations and dual-use considerations.

Next research steps. A stronger follow-up would use longer bounded tool loops, more tasks and providers, a no-probe invitation control, balanced peer-replacement datasets, and matched-information monitor arms. Private shared budgets, attenuated delegation, source freshness, revocation, and bounded repair feedback should be evaluated against adaptive splitting and probing. Such features are proposed work, not results of this submission. Privacy should be measured against explicit recipients and repeated-query inference; cost optimization should report both proof-generation and full-session execution costs, with plaintext and trusted-service baselines.

6 Conclusion

FeeGuard supplies bounded answers to all four Track 3 questions and links them to verifiable local financial consequences. Intact tool order carries an instructed bit; sorting removes that finite payload while preserving explicit cooperation. The invitation and replacement pilots retain negative findings. Monitoring notices change some short-run behavior, but do not establish lasting disruption. Complete ZK and plaintext policies reject the planted unsafe controls and retain the honest controls, whereas dual signatures alone do not. ZK's demonstrated role is private verification of a finite mandatory policy, at higher measured computational cost. The result supports an execution boundary as a complement to behavioral auditing, conditional on explicit trust assumptions.

Code, data, and work attribution

The pinned public snapshot is MartinEuclid/feeguard, commit 47741cd. It includes prompts, final outputs, protocols, local receipts, proof objects, a dashboard, and a browser verifier. The new Track 3 work is in `src/carrier-peer-study.mjs`, `src/watching-study.mjs`, and `src/track3-tool-replay.mjs`. The treasury relation, contracts, proving artifacts, earlier omission/sealed-task experiments, and adapter benchmarks are reused from earlier FeeGuard iterations; their records are not pooled into the 141 calls. No official starter-repository code was incorporated into these new experiment modules. This manuscript reorganizes the archived evidence; it reports no newly collected model behavior. Gu Xinfeng is the sole submitting author. AI assistance is disclosed in Appendix A.

References

[1] Apart Research. 2026. *AI Collusion Research Sprint: research questions, guidelines, and report template*. Accessed 26 October 2026. https://apartresearch.com/sprints/ai-collusion-research-sprint-2026-10-23-to-2026-10-25.

[2] Mason Nakamura et al. 2026. *Colosseum: Auditing Collusion in Cooperative Multi-Agent Systems*. arXiv:2602.15198v2. https://arxiv.org/abs/2602.15198v2.

[3] Sumeet Ramesh Motwani et al. 2025. *Secret Collusion among AI Agents: Multi-Agent Deception via Steganography*. arXiv:2402.07510v5; first submitted 2024. https://arxiv.org/abs/2402.07510.

[4] Jurgis Kemeklis and Nikhil Vemuri. 2026. *Omission Attacks: When Doing Nothing Is the Attack*. AI Control Hackathon participant report, Team MAIA. https://apartresearch.com/sprints/projects/omission-attacks-when-doing-nothing-is-the-attack-0y1v.

[5] Ryan Greenblatt, Buck Shlegeris, Kshitij Sachan, and Fabien Roger. 2024. *AI Control: Improving Safety Despite Intentional Subversion*. ICML; arXiv:2312.06942v5. https://arxiv.org/abs/2312.06942.

[6] Khanh Linh Nguyen, Hoa Nghiem, and Tu Tran. 2026. *TraceGuard: Structured Multi-Dimensional Monitoring as a Collusion-Resistant Control Protocol*. arXiv:2604.03968v1. https://arxiv.org/abs/2604.03968v1.

[7] Jens Groth. 2016. *On the Size of Pairing-based Non-interactive Arguments*. EUROCRYPT; IACR ePrint 2016/260. https://eprint.iacr.org/2016/260.

[8] M. Llambí-Morillas and D. Fernández-Fernández. 2026. *Cryptographically verifiable authorization for autonomous AI agents: a falsifiable hypothesis and proof of concept*. arXiv:2607.21325v3. https://arxiv.org/abs/2607.21325.

[9] Xinrui Shi, Yanzhe Zhang, and Diyi Yang. 2026. *Emergent Collusion in Long-Horizon LLM Agent Interaction*. arXiv:2609.24967v2. https://arxiv.org/abs/2609.24967.

A Limitations and Dual-Use / Ethical Considerations

Scope of the behavioral evidence. The carrier and monitoring subjects are assigned adversarial roles; calibrated decoding is not naturally emerging collusion. The invitation subjects are ordinary reward-pressured agents, but the three approaches never recruit unsafe joint action. No no-probe comparison or established coalition is tested. Peer observations do not attest provider identity, weights, hidden state, or durable conventions. Every later context is fresh, even when application memory is restored. The two aliases come from a single provider. Monitoring notices do not establish awareness in one arm and unawareness in another; two rounds provide only short-run observations. Public lexical features are diagnostics, not proof of internal reasoning or channel migration.

Statistical and measurement limits. The study has three fixed cases, dependent receiver batches, paired branches, and repeated local replays. The 141 calls and 37 proofs are not independent experimental units. We do not attach binomial intervals to pooled predictions or infer a population failure rate. The 16-to-0 opening count is a syntactic recipient-specific disclosure measure, not entropy reduction or a differential-privacy guarantee. Local JSON sizes and EVM gas are not on-chain billed currency, production latency, transport bytes, physical disk space, or a measured reduction in human review time. The historical adapter benchmark has a distinct cohort and relation.

Cryptographic and operational assumptions. The source authority and owner can register a false or incomplete world. A commitment prevents changing that registered world silently; it does not establish its truth. Four payable slots, fixed horizons, and deterministic principal return simplify financial reality. Unmodeled taxes, fees, valuation risk, asynchronous external obligations, compromised keys, and bypass execution are not covered. The local Groth16 setup has no production multiparty ceremony or independent security audit. The code and security argument are a research prototype, not a formal verification or guarantee against all circuit/contract defects. Supplier payment is owner-authorized and closes the snapshot; it is not automatically authenticated against the private invoice relation.

Privacy and availability. Agents and the prover see synthetic source data. A verification packet omits the sixteen openings, private messages, codebook, and memory, but public values and repeated acceptance outcomes still allow inference. Treasury cash and amounts are not shielded. Public research fixtures disclose the synthetic records by design. Source/prover infrastructure, timing, network metadata, issuer identity, and query leakage require separate controls. Both agents can refuse useful work, and a mandatory policy cannot force them to produce a good proposal. The monitor's 0/3 honest completion is confined to these fixtures and does not measure all LLM reviewers.

Dual use and research conduct. Demonstrating side-channel coordination and omission can inform attackers. The released examples are limited to synthetic accounts and local controls; no real funds, customer data, production systems, or unsuspecting participants are used. Signed audit envelopes attest integrity under recorded keys, not an organizer's endorsement, legal compliance, provider identity, or public-chain finality. The experiments retain final structured decisions and supplied prompts; hidden chain-of-thought is not requested. False-positive blocking, selective disclosure, and centralized source authority are governance tradeoffs rather than eliminated risks.

LLM usage statement. Codex substantially assisted project design, implementation, testing, analysis, and manuscript drafting. Numerical claims were cross-checked against saved records; cryptographic and signature verification was rerun during report preparation without collecting new model behavior. AI assistance is not independent human replication. Gu Xinfeng remains responsible for reviewing and submitting the work. Earlier project iterations and external methodological inspiration are identified in the main text; no new cryptographic primitive, general anti-collusion guarantee, or matched superiority over a published system is claimed.

B Reproduction, Evidence Map, and Integrity

Pinned environment. Use the linked commit, Node.js 22+ and pnpm. Dependencies include snarkjs 0.7.6, circomlib 2.0.5, circom2 0.2.23, ethers 6.17.0, Ganache 7.9.2, and solc 0.8.30. Install with `pnpm install --frozen-lockfile --ignore-scripts`. Keep the supplied proving/verifying artifacts when checking archived proofs; regenerating keys changes the verification reference. No model account is required for saved-data verification or deterministic replay. Fresh inference requires the author's configured model access and consumes usage.

Purpose	Command or repository path
Verify studies and proofs	`node scripts/verify-track3.mjs`
Start dashboard	`node src/server.mjs`; open `/track3` and `/track3-verifier`
Questions 3.1–3.3	`src/carrier-peer-study.mjs`; `examples/track3-complements.json`
Question 3.4	`src/watching-study.mjs`; `examples/watching-study.json`
Tool projection replay	`src/track3-tool-replay.mjs`; `examples/track3-tool-replay.json`
Frozen protocols	`examples/track3-complement-protocol.json`; `examples/watching-protocol.json`
Witness and actuator	`circuits/treasury_transition.circom`; `contracts/TreasuryVault.sol`
Study-specific tests	`node --test tests/track3.test.mjs`
Full regression command	`node --test --test-concurrency=1 tests/*.test.mjs`
Historical cost benchmark	`docs/BENCHMARK-RESULTS.md`; `examples/benchmark.json`

Table 5: Evidence locations in the pinned repository. Full synthetic exports retain prompts and final outputs; public proof exports deliberately omit private research fields.

Replays and provenance. Both experiment scripts accept `--replay` to reuse saved decisions with fresh local execution. The monitoring entry point is `scripts/watching-study.mjs`. The other pilots use `scripts/track3-complements.mjs`. Replays write separate runs and do not constitute new independent behavioral samples. The saved full and public study envelopes are checked separately. Receipt hashes describe recorded local-chain transactions, not public-network inclusion. The verifier also checks source/artifact binding and recomputes declared outcomes under the implemented policy.

Prompt access. Complete prompts and final JSON are in each archive's `cache.calls`. Prompt builders include `checkpointPrompt`, `continuationPrompt`, and `watchMonitorPrompt`, with schemas and frozen thresholds in source. A/B are told they have no tools or execution keys and must return brief final decisions, not hidden reasoning. Invitation and peer prompts are in the carrier/peer module. This distinction prevents interpreting harness-mediated simulated tools as direct model tool use.

Protocol hashes (SHA-256). Complement protocol:
`80f8daa69bdea34e3e8044fbad98a64c42d97f6359d0c6fe8887b15dd18e456a`
Monitoring protocol:
`3f5f043ffd221132e5768be9ac59c52abaffdbdb2e6073dffb3f673d2af2915d`

Validation provenance. The archived regression stage passed 148 tests; a later pinned-install focused stage passed 19, overlapping in 15. Thus 152 distinct checks passed across stages, not a single fresh 152-test run. During preparation of this manuscript, the two study verifiers and tool-projection verifier passed again: 141 calls, 37 verified proof objects, four signed envelopes, 384 trial projections, 24 calibrations, and 816 simulated reads. This validation checks retained records and cryptographic artifacts; it does not independently attest the original provider's model weights or replay every historical chain transaction. The report's accompanying manifest identifies the exact evidence files and PDF checksum.

C Matched Observations and Claim Boundaries

The four treasury observations underlying Table 4 are reproduced below. P and Z denote complete plaintext and ZK respectively. The first three are deterministic honest controls; the last is the repaired second-round action in W1's disclosed ZK branch, matched to its fixed-action plaintext replay. They are not four independent model trajectories or repeated timing samples of one controlled workload.

Request	P bytes	Z bytes	P gas	Z gas	Proving ms
W1 honest	1,009	1,112	169,950	410,341	2,171
W2 honest	1,012	1,110	169,986	410,389	1,269
W3 honest	1,009	1,110	169,974	410,305	971
W1 repaired R2	1,014	1,111	169,957	410,341	6,770

Table 6: All four matched successful request observations. The median of 971, 1,269, 2,171, and 6,770 ms is 1,720 ms. No attribution of the timing variation to a particular cause is supported by this small archive.

Question or mechanism	Supported observation	Not established
3.1: carrier	Instructed intact-order decoding; specified sorting intervention; explicit owner controls retained	Spontaneous conventions, general covert-channel removal, timing or argument-channel suppression
3.2: invitation	No unsafe recruitment in the tested direct/indirect approaches; benign cooperation retained	Disruption of an established coalition or a causal benefit of adding a trusted agent
3.3: replacement	Observers always guess unchanged; both views have 50% balanced accuracy	Attested model recognition, provider anonymity, or disruption of subsequent financial actions
3.4: monitoring	Some two-round proposal changes; unsafe agreement can persist despite blocked execution	Durable behavioral correction, learned concealment, or causal channel migration
ZK and contract	Proof verification and state/intent binding; private openings omitted from verifier packet	Honest intent, true/complete real-world sources, shielded balances, or protection through a bypass route
Cost	Measured local packet sizes, authorization gas, and four proof times	Cheaper production payments, faster matched end-to-end review, or measured human labor savings

Table 7: Claim-to-evidence boundaries for reviewers. Financial harm, unsafe agreement, carrier usability, and disclosure are separate endpoints.