

Runtime Revocation in Multi-Agent Systems: Protocol Verification and Evaluation of Privacy and Containment

Executable Models, Failure Tests and Evidence for Implementation

Dr. Carsten Stöcker
Spherity GmbH

Prepared: 10 October 2026



Except where otherwise noted, this work is licensed under the Creative Commons Attribution 4.0 International License (CC BY 4.0). © 2026 Dr. Carsten Stöcker.

Abstract

Credential withdrawal does not itself stop a queued or remotely executing operation. The effect depends on how authoritative state reaches a protected receiver, how freshness is bounded, and whether unresolved work survives failures. This paper develops the protocol and evaluation component of a two-paper study of cascading trust failures in artificial intelligence agent networks. It specifies generation-checked admission, parent-capped leases, persistent operation identity and authenticated terminal closure. Three conditional protocol propositions are examined using an exhaustive finite-state abstraction; the two-identifier baseline contains 6,651 reachable states, and deliberate omissions produce counterexamples. A matched discrete-event study executes 6,144 synthetic incidents across signed-mirror and local-ledger publication paths. Both support local decisions; their enforcement and availability differ under the configured publication and failure conditions. A causal fixed-cohort decoy construction is analysed using exact membership posteriors and 5,000 synthetic cohorts. Matched complete histories preserve the allocation prior, whereas hazard mismatch and genuine-only shocks reveal membership. Public active initialization makes a fixed mask information-equivalent to the unmasked history while increasing encoded bytes. Two further propositions extend conditional cascade bounds to continuing external roots and componentwise parameter uncertainty. The contribution is a reproducible composition of protocol obligations, privacy analysis and bounded-harm reasoning. The results do not establish production performance, standards conformance or safety outside the stated authority, receiver and stochastic premises.

Keywords: multi-agent systems; credential revocation; runtime assurance; authoritative freshness; decoy privacy; distributed ledgers; bounded containment.

Contents

1	Research question and contribution	5
1.1	Contributions and evidence classes	5
1.2	Scope of the two-paper study	6
2	Scope, authority contracts and related work	6
2.1	One effect boundary across three domains	6
2.2	Authority, evidence completeness and freshness	7
2.3	Adversary and observation boundaries	7
2.4	Position among established architectures and status methods	8
2.5	Bounded scope and contribution boundary	9
3	Protected execution and revocation protocol	10
3.1	Roles and the complete execution record	10
3.2	Durable state and guarded transitions	10
3.3	Local withdrawal exclusion	13
3.4	Source-age caps and finite leases	15
3.5	Persistent identity, cancellation and terminal closure	16
4	Exhaustive finite-state verification and omission counterexamples	17
4.1	Executable abstraction and explored bounds	17
4.2	Monitors and connection to the proofs	18
4.3	Removing individual controls	19
4.4	Packed-word update interleavings	19
4.5	Reproduction and remaining refinement work	20
5	Standards-bound implementation profiles	20
5.1	Normative baseline and study-specific requirements	20
5.2	Standard bitstrings, decoding and status interpretation	21
5.3	Pinned terse reference profile	22
5.4	Experimental decoys and initial masks	22
5.5	Packed ledger words and concurrency	23
5.6	Resident state, authority and recovery	23
5.7	Shared signals and receiver enforcement	24
5.8	Signature sizes, transport and algorithm migration	24
5.9	Profile evidence and conformance limits	25

5.10	Anonymous non-revocation proof adapter	25
6	Matched workload and hosting evaluation	26
6.1	Executable scope and comparison contract	26
6.2	Equivalent policy decomposition and reserved work	27
6.3	Publication, renewal and failure conditions	28
6.4	Synthetic incident outcomes	29
6.5	Byte boundaries, interpretation and replication	31
7	Causal decoy generation and full-transcript membership privacy	31
7.1	Objective, allocation and observation boundary	31
7.2	A causal candidate under a common environment	32
7.3	Exact inference with a fixed-count prior	33
7.4	Approximate matching over complete histories	35
7.5	Initial masks and the known-initialization observer	36
7.6	Reserve lifetime and representation costs	36
8	Synthetic privacy, reserve and publication results	37
8.1	Experiment and reproducibility boundary	37
8.2	Membership inference and shock leakage	38
8.3	Fixed-mask comparison and encoded bytes	39
8.4	Reserve use and remaining implementation limits	41
9	Continuing roots, uncertainty and conditional harm	41
9.1	From receiver enforcement to a propagation premise	41
9.2	Finite-horizon immigration theorem	42
9.3	Robust envelopes and limits of estimated parameters	43
9.4	Exact two-type illustration	44
9.5	Pending tails and operational interpretation	46
10	Integrated evaluation and statistical interpretation	46
10.1	What connects the three studies	46
10.2	Units of replication and uncertainty	47
10.3	Progression to a deployment experiment	48
11	Practical implementation guidance	48
11.1	Protected gates and persistent operations	48
11.2	Caches and dependency processing	49

11.3 Service admission and operational thresholds	49
11.4 Privacy lifecycle and algorithm changes	50
11.5 From the abstract protocol to implementation tests	50
12 Research extensions and external validity	52
13 Scope and conditions for article claims	53
References	54
A Reproduction contract	56
A.1 Computational artifacts and fixed inputs	56
A.2 Regeneration and interpretation	56
B Multi-agent implementation checklist	57
C Evidence required beyond the model study	58
D Runnable model code and selected outcomes	58
D.1 Environment and complete source-package bootstrap	59
D.2 Protocol invariants and omission traces: Sections 3–4	61
D.3 Paired synthetic workloads: Section 6	62
D.4 Fixed-cohort inference, masking and shared shocks: Sections 7–8	64
D.5 Continuing roots and parameter uncertainty: Section 9	66
D.6 Reproduction record and implementation boundary	69
E Implementation conformance and standards contribution fixtures	70
E.1 Separate the standard, the profile and the experiment	70
E.2 Minimum producer and reader fixtures	70
E.3 Packed-word updates require their own contract	71
E.4 Evidence needed before a standards contribution	72
F Evaluation plan for anonymous status adapters	72

1 Research question and contribution

An artificial intelligence (AI) agent can hold a valid credential when its authority to perform a particular effect has already changed. The change may concern a delegated mandate, an assurance premise, a source of evidence or the policy of the resource owner. A signed withdrawal helps only when the receiving system interprets it correctly and prevents the affected effect in time. Between the source decision and that boundary lie publication, transport, validation, scheduling, state installation and possibly a remote operation already accepted for execution. The research question is which explicit protocol and service conditions connect those stages to a defensible containment claim.

The companion article, *Cascading Trust Failures in AI Agent Networks*, hereafter Article A, establishes conditional relationships among complete-record eligibility, credential multiplicity, status representation, freshness and expected cascade harm [1]. It also calculates storage factors and measures reproducible compression fixtures. The present paper develops the operational part of that framework. It specifies a guarded receiver protocol, checks a bounded executable abstraction, constructs a causal decoy experiment and examines matched publication paths under a common workload. A finite-horizon result extends the cascade model to continuing external incidents.

The setting is a multi-agent system (MAS) coordinating engineering work across organisations. Agents propose releases of synthetic design files. A protected receiver decides whether the proposed release is permitted and records the actual effect. The study separates prohibited releases, legitimate work lost before its deadline and operations whose outcome remains unresolved. It does not infer harm from an agent's own explanation. This setting makes the effect boundary observable while retaining the central difficulty of cross-domain authority and delayed withdrawal.

1.1 Contributions and evidence classes

The first contribution is an explicit state-transition contract. Admission binds one complete execution record, reserves its persistent operation identity and records its debit. Execution checks the record at the last preventable-effect boundary. Withdrawal installation advances a protected generation, while receipt-based closure preserves accountability across retries, crashes and missing responses. The associated propositions state what follows from those transitions and which properties still depend on the receiver's implementation.

The second contribution is a set of executed model studies. The finite-state checker explores all reachable states of its declared small configuration and supplies omission traces. The workload model compares independently operated signed mirrors with an authenticated local ledger view. Both retain local state and obey the same accepted-age rule. The decoy study evaluates a specified absorbing-state generator under matched hazards, misspecified hazards and a common incident shock. These are reproducible computational results about the supplied models. They are not measurements of a production network, a deployed ledger or a population of autonomous language-model agents.

The third contribution connects those studies to conditional harm analysis. The protocol identifies which delay is an enforcement bound. The service model states the work and capacity premises needed to support such a bound. The privacy model identifies the observation law under which cover traffic hides unknown membership. The continuing-root theorem then shows how new external episodes enter the expected-harm calculation without implying that a subcritical system has bounded cumulative harm for unlimited time.

These contributions use established techniques. State-machine invariants, local linearisation, conditional expectation and matched cover processes are not claimed as new mathematical methods. Their role here is to expose the assumptions required by this combination of credential status, distributed enforcement and agent-network containment. Novelty lies in the specified composition and the tests of its failure boundaries. The manuscript therefore states the status of each result at its point of use rather than presenting a single undifferentiated claim of system safety.

1.2 Scope of the two-paper study

Article A provides the architecture and conditional scaling argument. This article owns the protocol, executable abstractions, causal-decoy candidate and evaluation method. The distinction avoids repeating the first paper's propositions as new results. It also permits a negative outcome: a hosting alternative may provide no advantage under a stated fault, a causal generator may reveal genuine positions, and a fitted workload may fail to supply the assumptions needed for a harm certificate.

Readers seeking an implementation view can begin with the file-release record and worked ordering example in Section 3. Section 4 shows which failures the small executable checker finds and what its successful checks leave unproved. Section 5 separates current standards from experimental profiles. Section 6 explains how the same synthetic incident is compared across hosting paths. Sections 7–8 examine what an observer can infer from decoy histories; the short examples explain the mathematical question before its detailed calculation. Section 9 connects supported timing assumptions to conditional harm bounds, including continuing external incidents. Sections 10–12 turn the results into evaluation and implementation obligations. Readers checking proofs should also examine each proposition's premises and the abstraction limits in Sections 2, 4 and 13. The runnable appendices make selected results inspectable; executing a notebook does not establish conformance or safety of a deployed service. Appendix D gives the runnable learning sequence. Appendix E identifies profile fixtures and proposed standards contributions.

2 Scope, authority contracts and related work

2.1 One effect boundary across three domains

The study concerns a finite engineering-agent workflow in which one organization may authorize computation while another controls release of the resulting design file. Domain A issues or withdraws the relevant authority and evidence. Domain B distributes authenticated state, or supplies an attestation under a separately recognized mandate. Domain C operates the resource and enforces the requested effect. These domains represent administrative responsibility, not three compulsory network services. An organization may operate several components without removing their logical trust boundaries.

The protected object is the complete execution record $g = (a, h, \pi, \nu)$ inherited from Article A [1]. It binds action, duration, continuation and context. Context includes the audience, resource, evidence and policy versions, operation identity and relevant harm definition. Agreement about the action name is insufficient when the permitted continuation differs. The companion therefore reuses Article A's eligibility and accounting contracts; it does not present their proofs or completed codec fixtures as new results. The new question is whether a concrete protocol and its representation profiles preserve these contracts under concurrent updates, delayed delivery and restricted service capacity.

The primary harm is an executed file release outside the actual permission. The measurement boundary is the last point at which the receiver can prevent that release. A decision log that says “deny” does not establish prevention if a previously queued effect executes afterward. Legitimate completion by deadline and unresolved remote work are reported separately. An architecture that rejects every request can avoid prohibited releases while failing its useful task. These outcomes are not collapsed into a weighted score that allows throughput to compensate for confidentiality loss.

The same distinction applies to propagation. Resource containment and communication containment are separate controls. If the cascade model counts an agent’s outgoing messages, the receiver or channel must enforce the corresponding bound on covered influence. A resource gate alone cannot certify termination of an uncontrolled messaging path. Later experiments may use language-model agents to produce proposals; the protocol state and deterministic effect records remain the basis for evaluating the security properties.

2.2 Authority, evidence completeness and freshness

Three kinds of reliance are separated. Cryptographic verification establishes that an accepted key authenticated specified bytes under a selected algorithm. Recognition establishes that the relying policy permits the corresponding entity to make a particular kind of assertion. Applicability establishes that those assertions jointly justify this record in the current context. None of these steps is interchangeable with the others. A valid signature from an unrecognized source supplies no mandate; recognition of a source does not establish that every authentic statement it makes is true.

The World Wide Web Consortium (W3C) Recognized Entities draft describes entities in relation to recognized actions. The version fixed here is the Working Draft of 6 September 2026, whose status text identifies it as experimental and unsuitable for production deployment. Its companion threat model is the Group Note Draft of 8 October 2026 [2,3]. These are useful design references, not evidence that the present system conforms to a mature trust-list standard. In this study, source recognition is explicitly scoped to claims, actions and evidence classes. Distribution by Domain B does not enlarge that scope.

Evidence discovery is complete only relative to a declared contract. The contract identifies mandatory sources, required adverse evidence, applicable alternatives and a termination rule for discovery. A content digest detects changes to a known object; it cannot prove the absence of an unknown adverse fact. Likewise, a collection of individually current assertions need not describe a jointly permissible continuation. Cross-source version compatibility is an explicit protocol input. If that relationship cannot be established, the gate restricts the continuation or refuses ordinary admission.

The authoritative watermark ω_v has a stronger meaning than retrieval time: it states that the accepted view includes all relevant authoritative decisions through the indicated time. A new download, signature or ledger block does not establish that premise. A publisher may distribute the same old contents successfully while the source has already withdrawn a dependency. The profile therefore binds the watermark’s source, scope and completeness assertion. Its truth is a declared trust assumption unless a separate mechanism supplies the premise.

2.3 Adversary and observation boundaries

The operational adversary may submit unauthorized proposals, replay authenticated stale objects, delay or reorder messages, exploit update races and induce crashes. The threat and

deployment-test scope includes bursts that compete with withdrawal work, ambiguous encodings, lost terminal receipts and a partition that prevents all new authoritative evidence from arriving. Each executed study states which of these cases its abstraction represents. The trusted computing base contains the gate, its protected persistent state, its applicable policy and the supported clock-error mechanism. A bypass of this boundary is a separate omission case, not a successful execution of the specified protocol.

Authority compromise needs a separate interpretation. A publisher can issue inconsistent views or sign a false but recent active state. Replication may expose inconsistency if different views are compared. It cannot turn a common false assertion into truth. For each such case, the analysis distinguishes prevention, detection under an added mechanism and violation of a model premise. This prevents authentic misinformation from being counted as correct status processing. The recognition threat model provides relevant examples of stale recognition, substituted resources and overextended recognition chains [3].

The privacy observer is specified independently of the operational adversary. Its transcript can include public snapshots, update transactions, event identifiers, publication writers, timing, known credential positions and external incident information. A holder or legitimate verifier may know its own index and initial mask. Consequently, the model does not assume that every mask bit is an operator-only secret. Public initialization can reveal further information. Active interventions require a declared strategy; passive transcript matching does not establish resistance to an adaptive attacker.

Four privacy objectives remain distinct: hiding which positions represent genuine credentials; limiting disclosure of issuer activity; avoiding decision-specific lookup metadata; and limiting linkage between credential presentations. A construction may address one objective while leaving another unchanged. The protocol and evaluation identify both the observer and the protected variable before comparing costs. A lower byte count is not a privacy measure, and an unsuccessful classifier is not a proof of indistinguishability.

2.4 Position among established architectures and status methods

The National Institute of Standards and Technology (NIST) Zero Trust Architecture separates the policy decision point (PDP) into policy-engine and policy-administrator roles and distinguishes the policy enforcement point (PEP) at the resource boundary [4]. This paper adopts that separation. Its additional obligations concern complete-record binding, authoritative freshness and unresolved cross-domain effects. They are a profile of the present problem, not claims that NIST requires a ledger or provides a general agent-safety theorem.

W3C Bitstring Status List v1.0 is a Recommendation dated 15 May 2025 [5]. It provides a status representation, not a complete resource-enforcement protocol. The dated v1.1 First Public Working Draft adds a terse reference profile [6]. Section 5 separates those normative formats from experimental masking and decoys. OpenID Shared Signals Framework (SSF) and Continuous Access Evaluation Profile (CAEP) supply event mechanisms [7,8]. They do not make event receipt equivalent to execution control. The receiver’s authority, state-transition and deadline rules remain necessary.

CRSet uses Bloom-filter cascades, padding and regular publication to address issuer-activity privacy [9]. It therefore prevents a novelty claim based merely on adding cover information or checking status locally. The relevant difference is the present connection between representation workload, protected receiver enforcement and conditional harm. A fair comparison would require equivalent observer, population and freshness assumptions; a byte comparison across different privacy objectives would not establish superiority.

ShadowPath moves credential-specific status work to the holder and uses a zero-knowledge proof against a selected authenticated root. Its August 2026 preprint separates verifier-visible lookup information from observable state synchronization, and excludes malicious issuer behavior and global consistency from its threat model [10]. It is a useful example of presentation privacy with explicit limits. It does not remove the root-freshness or effect-enforcement obligations studied here. The current article does not reproduce its benchmarks or claim that public bitstrings provide the same property.

Table 1: Position of the study among architectures and status methods.

Research line	Established contribution	Boundary addressed here
Zero Trust Architecture [4]	Separate decision, administration and enforcement roles.	Bind one complete record and persistent operation to the protected receiver.
Credential status [5, 6]	Interoperable status references and list representations.	Establish which accepted state justifies an effect and when reliance must stop.
Scoped recognition [2]	Express recognition for entities and their actions.	Declare source trust, applicability and discovery completeness.
Shared signals [7, 8]	Authenticated event exchange and defined event meanings.	Order accepted updates against admission and observe completed enforcement.
CRSet [9]	Issuer-activity privacy through a different representation and publication design.	Account for privacy workload in an effect-enforcement contract.
ShadowPath [10]	Holder-side private status verification against authenticated roots.	Preserve authority, freshness and closure obligations across hosting choices.

2.5 Bounded scope and contribution boundary

The reference comparison retains one resource workflow and two publication backends: issuer-signed snapshots from independently operated mirrors, and an authenticated local ledger view. Both may retain verified local state. Their comparison therefore concerns update publication, synchronization, recovery and enforcement; it does not award a unique lookup-privacy benefit to a local ledger read. A remote node interface, action-dependent refresh or observable telemetry changes the privacy boundary and must be measured explicitly.

Credential multiplicity is varied only under equivalent authorization predicates in the overhead study. If finer credentials reduce authority scope, affect incident reach or change assurance, that is a separate policy intervention. Population-level multiplicity, per-decision evidence demand and shared dependencies are not the same factor. The analysis records their separate influence before assigning a change in processing delay to any one of them.

The article’s contribution is thus a protocol and evidence framework for a specified composition problem. Established standards provide data and event formats; prior research supplies privacy constructions and general analytical tools. The present study identifies the contracts needed to connect these components to the actual effect boundary. Its results must remain partitioned into mathematical consequences of assumptions, bounded protocol checks, exact representation

calculations and observations from stated workloads. No one evidence class silently substitutes for another.

3 Protected execution and revocation protocol

3.1 Roles and the complete execution record

The protocol separates authoritative decisions, evidence delivery and the boundary that can prevent an effect. Domain A determines a scoped mandate or credential status. Domain B distributes origin-authenticated state; a distinct delegation is required if B evaluates evidence on another domain's behalf. Domain C operates the receiving resource and its protected PDP and PEP. These role names follow the resource-centred architecture in [4]. The protocol below adds explicit ordering, persistence and freshness obligations. Figure 1 summarizes these boundaries. A publication backend supplies evidence to this architecture; it does not replace the receiver's decision.

An untrusted agent proposes the complete record $g = (a, h, \pi, \nu)$ from [1]. The action a , duration h , continuation π and context ν bind the resource, audience, principals, payload, policy, evidence versions, named harms and accounting episode. Mandate and assurance must hold for that same record. Checking an action against one permitted continuation and a different assured continuation is insufficient. A canonical digest identifies a record only when its serialization and interpretation are unambiguous and the receiver validates the bound contents. An operation identifier (ID) names the logical operation bound to that record. The profile distinguishes immutable evidence and policy versions in the complete record from renewable status and freshness assertions in its supporting witness.

Engineering example. Two requests can both say “release the design file” while naming different recipients, evidence versions or later uses. They are different execution records. A permission for the named subcontractor cannot justify release to another company, and an assessment for one continuation does not cover a changed continuation. The identifier and digest preserve this binding through queues and retries. They do not make the underlying permission correct: the receiver must still evaluate the bound record under its own policy.

Let $\mathcal{L}_g$ contain every mutable premise required by the relying policy. It includes status, issuer recognition, key authority, policy and allocation state, as applicable. The evidence contract also names adverse assertions and discovery rules. The selected evidence must be jointly applicable under a declared cross-source version relation. Equality of timestamps does not establish this relation. Neither does a vector of locally observed versions. An atomic snapshot, a validated read set or another justified procedure must establish the coherence claimed by evaluation. An incomplete dependency set is a failure of a stated premise, rather than a condition repaired by the final version check.

3.2 Durable state and guarded transitions

At domain v , let $\Gamma_v(\ell)$ be the durable generation of premise ℓ . An accepted installation that changes any relevant premise also advances its generation in the protected local order. Generations cannot be reused. A source epoch and version identify authenticated origin state; the local generation additionally records changes in its relying interpretation. Installing a new recognition policy may therefore advance a generation even when the status bytes remain unchanged. Lower origin checkpoints are rejected. Epoch migration must invalidate or explicitly migrate live approvals; it must not reset their effective age.

Evaluation returns a witness $W_{v,g}$ containing the coherent evidence, the captured generation vector $\gamma_{v,g}$, complete-record identifier and expiry. Capturing $\gamma_{v,g}$ does not grant execution authority. For an operation identifier o , admission uses the following conceptual guard:

$$\begin{aligned} \text{Admit}_v(o, g, t) \Rightarrow & \text{Eligible}_v(g, W_{v,g}) \\ & \wedge \text{Fresh}_v(g, t) \wedge \text{SlotFree}_v(o) \wedge \bigwedge_{\ell \in \mathcal{L}_g} \Gamma_v(\ell) = \gamma_{v,g}(\ell). \end{aligned} \quad (1)$$

The protected transition atomically checks this guard, consumes the operation slot, records the permanent accounting debit and persists the unresolved operation. No effect may escape before that persistence succeeds. The immutable operation binding fixes o , g , payload, audience, receiver contract and covered continuation. Supporting authorization records carry their authenticated evidence, generations and absolute expiry. Reusing o for different immutable contents is rejected. A retry preserves the accepted authorization record and does not create another debit or silently extend expiry. Section 3.4 permits a separate renewal of supporting authorization within the same bound continuation and parent caps.

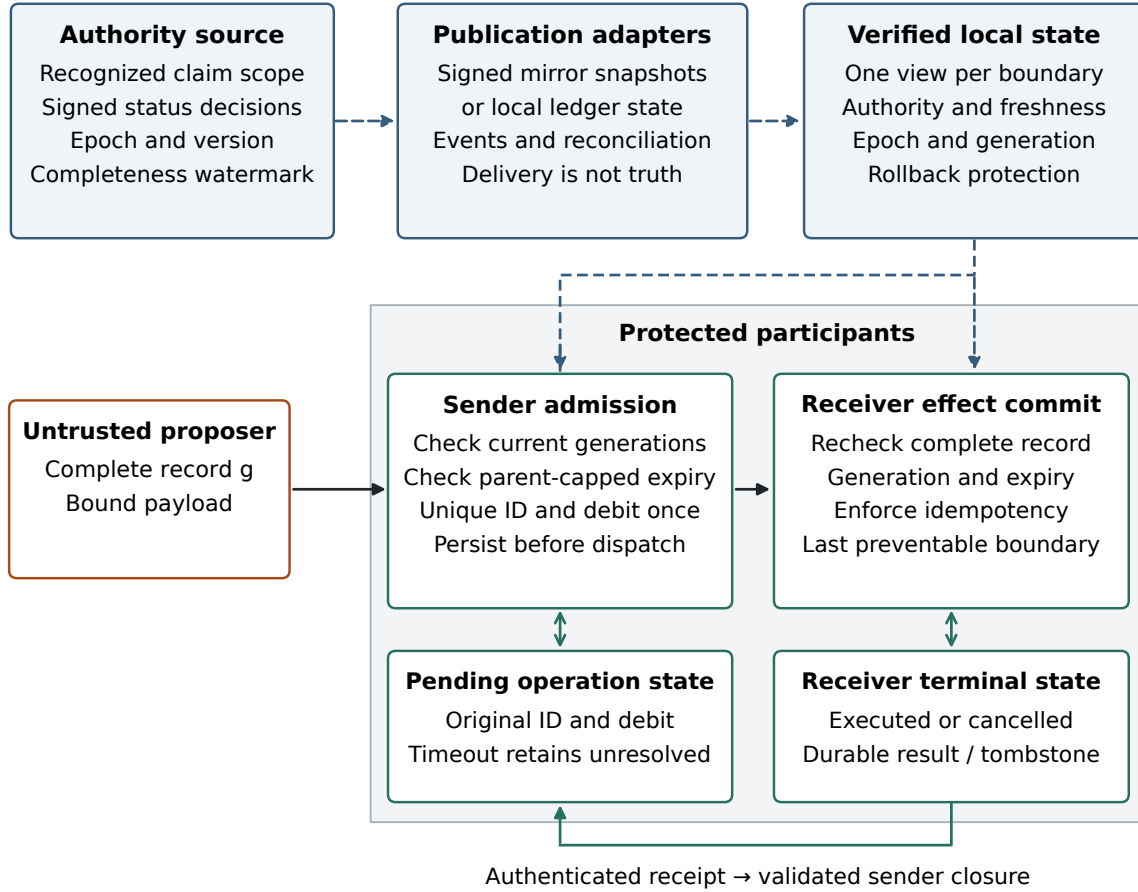
The sender's pending set $P(t)$ contains admitted operations without a contract-valid terminal outcome. Dispatch preserves the immutable operation binding and its identifier and carries the applicable authenticated authorization record. The receiver evaluates the required record under its own authority and evidence policy. A sender's generation number has no meaning as the receiver's generation number: each domain compares its own captured witness with its own current state. Where the receiver relies on a delegated evaluation, that reliance and its dependency-freshness obligations must be explicit. Merely forwarding an authenticated allow message supplies no such delegation.

Table 2: Protected transitions and their persistent consequences.

Transition	Protected condition and persistent consequence
Evaluate	Establish a coherent, policy-complete witness for one g ; capture local generations and expiry. No effect is authorised.
Admit	Compare the witness with current state, test expiry and the unique slot, then persist the debit and unresolved record atomically.
Install	Authenticate scope, epoch and version; update accepted state and all affected generations in the same local order used by the gate.
Renew authorization	Explicitly revalidate supporting authorization for the same bound operation; preserve parent age and continuation caps, identity and debit.
Dispatch or retry	Send the same immutable operation binding with its accepted authorization record. A retry cannot silently renew expiry. Preserve unresolved state after missing acknowledgements.
Receiver effect	Validate the receiver's applicable witness, current generations, expiry and logical identity at its last preventable-effect boundary.
Cancel	Atomically choose cancellation instead of an uncommitted effect; retain a terminal tombstone that rejects late dispatch.
Close	Validate authoritative terminal evidence, remove the operation from $P(t)$ and retain debit and audit records.
Restart	Restore checkpoints, generations, operation identities, debits and outcomes; reestablish clock assurance before ordinary admission.

Protocol architecture and protected effect boundary

Proposed protocol • Independent authority and local enforcement



Retries preserve the bound logical ID; the proposer has no bypass to effects.

Atomic effect/result persistence is an explicit receiver-adapter assumption.

Figure 1: Proposed protocol architecture. An authoritative source supplies scoped signed status decisions through either signed mirrors or local ledger state. Each protected participant validates its own authority, epoch, version and freshness premises; the diagram does not imply a global consistent cut. Sender admission atomically checks the current local generations and parent-capped expiry, allocates the logical operation, debits once and durably records pending state before dispatch. The bound envelope reaches a receiver that rechecks at the last preventable-effect boundary. Idempotency and atomic effect/terminal-state persistence are assumptions of the modeled receiver adapter, not properties inferred from sender persistence. Executed or cancelled outcomes produce authenticated receipts; sender timeout alone does not close an operation. Publication or event receipt does not establish semantic truth or completed resource enforcement.

The receiver's final transition is stronger than an ordinary database update followed by an external application programming interface (API) call. In the model, the last preventable effect and its durable terminal outcome share one protected linearization boundary. This uses the established reasoning discipline of linearizability [11]; it does not introduce a new concurrency correctness condition. Execution and cancellation compete in that order. For a release adapter, this is an assumption about where the irreversible release becomes committed. A practical adapter must justify the assumption, or replace it with a protocol that explicitly represents uncertain external execution. A local transaction cannot make an arbitrary remote API atomic. The finite checker in Section 4 models this protected boundary; it does not implement or verify an external adapter.

3.3 Local withdrawal exclusion

Let $R_{x,v}$ install a change that withdraws reliance on premise x , and let $E_{o,v}$ be the receiver's effect-authorisation transition. A local total order resolves races between these transitions. This order need not agree with another organisation's receipt order. Invalidation of a cache entry alone is insufficient: the final gate must compare the protected generation or perform a sound full reevaluation.

Proposition 1 (local exclusion of a stale witness). Assume that evaluation captures a coherent witness; every relevant change advances at least one captured generation without reuse; and the final effect transition atomically compares those generations with current protected state. If a relevant invalidation is installed after that evaluation and before the effect transition, the old witness cannot authorise the effect.

Proof. Let ℓ be a captured dependency advanced by the invalidation. After installation, $\Gamma_v(\ell) \neq \gamma_{v,g}(\ell)$. No later transition restores the captured generation. The conjunction in the final guard is therefore false. Atomic ordering prevents the check and effect from straddling an installation without one of the two orders being selected. A new coherent witness may justify another decision, but the old witness cannot. $\square$

The premise concerns relevant changes in the same domain, not the instant when an issuer decides to withdraw. The implication is

$$R_{x,v} \prec_v E_{o,v} \Rightarrow \text{the invalidated witness does not authorise } E_{o,v}. \quad (2)$$

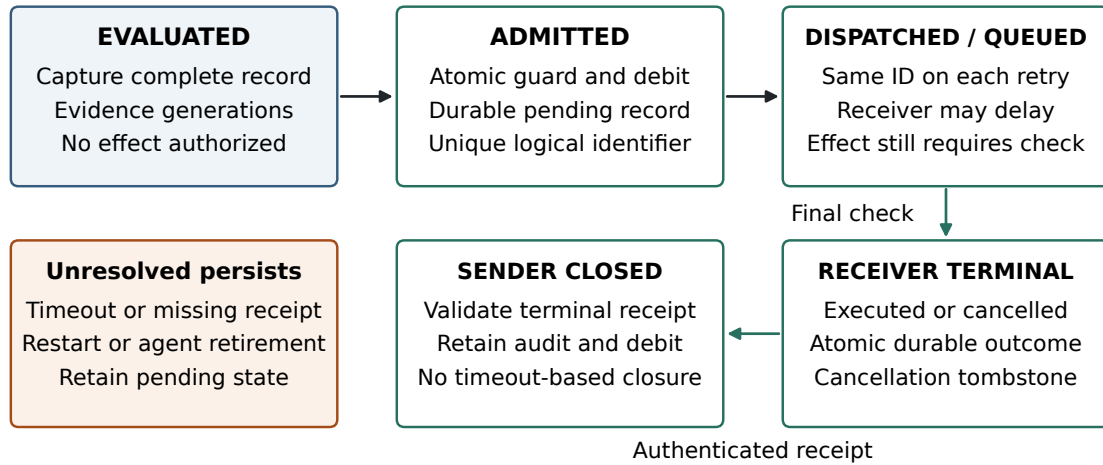
If the withdrawal remains in transit, the receiver may still admit a stale but age-acceptable view. If the effect commits first, later invalidation cannot undo it. Such operations remain represented in the harm and closure model. Network delay, publication ordering and source truth are therefore separate from the local exclusion proposition. The protocol does not establish global instantaneous revocation.

Engineering interpretation. The important race occurs where the receiver can still stop the file release. If withdrawal has already changed that receiver's protected state, its old approval must fail the final check. If the release commits before the change is installed, the local exclusion rule cannot undo it. This is why source decision time, receiver installation time and effect time are separate measurements. A prompt event acknowledgement does not tell the sender which of these two orders occurred.

Worked example: withdrawal reaches the receiving gate before effect commitment.

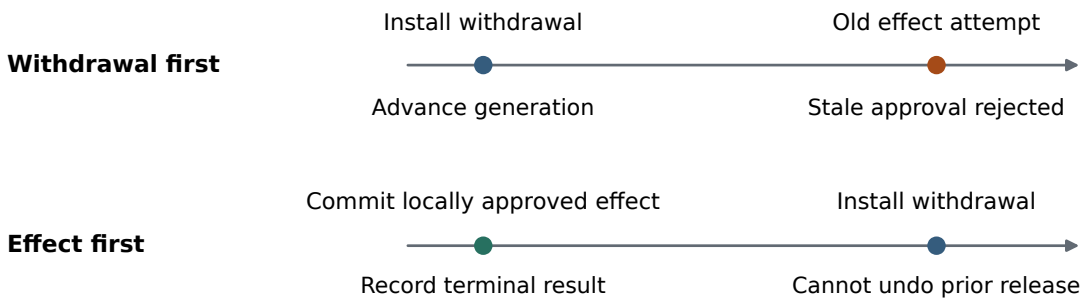
Persistent state and withdrawal-effect ordering

A Logical operation lifecycle



B Two legal orders at one receiver

Relevant withdrawal and final effect commit share a protected local order.



Cancellation competes with execution; its tombstone rejects later dispatch.

Figure 2: Logical operation lifecycle and protected receiver ordering. Evaluation authorizes no effect. Admission atomically checks generations and expiry, reserves the original logical identifier, debits once and persists unresolved state. Dispatch and retries retain that binding. The receiver performs the final applicable check and records an executed or cancelled terminal outcome under the stated atomic-adaptor contract. A validated authenticated terminal receipt permits sender closure; timeouts and restarts retain unresolved state. The lower panel shows the two legal orders between a relevant local withdrawal installation and effect commit. Installation first blocks the old approval; an already committed release cannot be reversed by a later withdrawal. Cancellation is ordered against effect commit and persists a tombstone. This local ordering neither supplies cross-domain atomic revocation nor proves correctness of an arbitrary external effect application programming interface.

Table 3: Worked local ordering with withdrawal installed before effect commitment.

Local step at the receiving domain	Protected state and result
1. Evaluate the proposed release	The witness captures dependency generation 0 for the complete record. No release has occurred.
2. Admit and queue operation o	The bound operation and accounting record are durable. The file remains behind the protected effect boundary.
3. Install withdrawal, $R_{x,v}$	The relevant generation advances to 1. The witness still records 0.
4. Attempt the effect transition, $E_{o,v}$	The final comparison fails because 0 differs from 1. The old witness cannot authorise release.

After $R_{x,v}$ is installed, an attempted $E_{o,v}$ using the old witness is blocked. If the effect instead commits before withdrawal installation, its outcome is accounted for as executed; later withdrawal cannot reverse that release. In either case, sender closure still requires contract-valid terminal evidence. The table illustrates the protocol's ordering rule and is not one of the reported shortest checker traces.

3.4 Source-age caps and finite leases

A lease limits the duration of a local approval; it does not make its evidence newer. Let s_ℓ be the true authoritative completeness cut for dependency ℓ , represented by an authenticated watermark $\omega_{v,\ell}$. With protected clock estimate $\hat{t}_v$ and a valid aggregate error margin $\kappa_{v,\ell}$, require the conservative age calculation

$$A_{v,\ell}(t) = \hat{t}_v(t) - \omega_{v,\ell} + \kappa_{v,\ell}(t) \geq t - s_\ell. \quad (3)$$

The guard requires $A_{v,\ell}(t) < \tau_\ell$ for every mandatory dependency, where τ_ℓ is its accepted-age limit. It also enforces mandate, policy and continuation validity and the lease's absolute expiry. Let b be the lease issue time and v_g the earliest applicable evidence, mandate, policy or continuation validity deadline. In a common supported timestamp convention, the expiry is capped as in [1]:

$$e_g \leq \min \left\{ b + \ell_g^{lease}, v_g, \min_{\ell \in \mathcal{L}_g} (s_\ell + \tau_\ell) \right\}. \quad (4)$$

Implementations translate these caps with their stated clock margins. Renewal may extend a shorter child lease within an already covered continuation, but cannot extend the source-age or validity cap without new authoritative evidence. Fetch time, wrapper-signature time and process uptime are not replacement watermarks. If clock assurance is lost, the corresponding age claim is unavailable and ordinary effects stop. Restart cannot turn an expired absolute deadline into a new relative lease. Renewal changes supporting authorization, not the immutable effect identity or its covered continuation. Evidence and policy versions bound into the complete record remain immutable. Renewal may replace only supporting status or freshness attestations assigned to the witness. Changing a bound version requires a new complete record and a separately admitted operation. Each renewal authenticates the original operation identifier and record digest, the new witness, its absolute expiry and an ordered authorization version. The receiver rejects stale renewal replays. The finite checker represents renewal as a guarded state transition; it does not model signed renewal messages or their independent delivery.

Proposition 2 (partition-safe stopping under honest source freshness). Suppose premise ℓ is withdrawn at true time r . Every stale active assertion accepted afterward has completeness cut $s_\ell \leq r$. Assume the conservative-age inequality and the strict accepted-age guard are enforced at every covered receiver effect. Then no such assertion can authorise a covered effect at $t \geq r + \tau_\ell$, even if no further message is delivered.

Proof. For $t \geq r + \tau_\ell$ and $s_\ell \leq r$, the true age satisfies $t - s_\ell \geq \tau_\ell$. Conservative age is at least true age, so the strict guard fails. Earlier lease or validity expiry can only stop effects sooner. Renewal retaining the same completeness cut cannot change this argument. $\square$

Clock uncertainty is included inside this guard. A convention that checks uncorrected timestamps must instead include its supported clock allowance in the external stopping bound. The proposition needs no delivery fairness, but it provides no availability guarantee. It also fails against an authority that falsely asserts fresh active state after withdrawal. An advancing ledger height does not establish an advancing completeness cut. Replication and signatures cannot supply this semantic premise.

3.5 Persistent identity, cancellation and terminal closure

The receiver maintains a durable outcome map keyed by the bound logical identifier. Its terminal states include executed and cancelled. A cancelled state is a tombstone, not deletion of the identifier: a delayed execution message must still be rejected. Replicated receivers need one serializable operation namespace and a failover fencing rule. Separate replicas that each deduplicate locally do not establish the required invariant.

Proposition 3 (conditional at-most-once effects and persistent accountability). Assume immutable operation bindings; one serializable admission ledger per bound operation identifier and accounting namespace; atomic first admission with its debit and pending record; a receiver transition that atomically chooses one terminal outcome and commits any modelled effect; persistent outcome and cancellation records across restart; and sender closure only on contract-valid terminal evidence. Then each logical identifier incurs at most one admission debit in that accounting namespace and at most one modelled effect at the bound receiver. An admitted operation remains in $P(t)$ until terminal evidence is validated, irrespective of timeout, duplicate delivery or restart.

Proof. Induct over the protected transitions. Before first admission the identifier is unused. First admission consumes its unique slot and writes one debit and pending record; later admission attempts cannot repeat that transition. An effect is enabled only while the receiver outcome is unresolved. Its atomic transition records executed, after which neither another effect nor cancellation is enabled. Cancellation instead records a terminal tombstone and similarly disables execution. Restart restores these facts. Finally, the only transition removing an admitted identifier from the pending set is validated terminal closure. Timeout and retry preserve the set. These cases preserve all three assertions. $\square$

The proposition concerns the modelled effect per identifier. Equivalent requests submitted under different identifiers require separate policy and budget controls. It does not provide exactly-once completion, progress, or cancellation of an effect already committed. A terminal signature is meaningful only under the receiver's trusted transition contract. An acceptance acknowledgement, a cancellation request and absence of a response are not terminal evidence. If downstream queued work can still execute, cancellation is not yet terminal under the stated contract. Separately authorised containment and recovery remain possible when ordinary admission stops, but their scope and assurance must be established independently.

4 Exhaustive finite-state verification and omission counterexamples

4.1 Executable abstraction and explored bounds

We implemented a finite transition system that makes selected obligations from Section 3 executable. It represents one authoritative credential, one status list, a protected gate and receiver, and one or two immutable operation identifiers. The sender and receiver records are separate logical milestones within one enumerated global state. They share one installed-generation variable and one up/down flag; they are not independently implemented processes. Domain-specific crashes and separately delayed accepted views are not represented. Signature validity, authority recognition, terminal-receipt authenticity and coherent evidence evaluation are assumptions of the abstraction. The result is a check of a specified abstract machine, not verification of a distributed service, ledger client, cryptographic implementation or production release adapter.

The clock takes values $\{0, 1, 2, 3\}$. Its error is zero. An initial active snapshot has authoritative watermark zero and maximum accepted age two. Child leases have duration one and cannot exceed that parent-age cap in the baseline. There is one fixed credential epoch and at most one authoritative withdrawal. The accepted local generation is initially zero and becomes one when withdrawal is delivered. The source decision and its delivery are different transitions. A replay of the old active snapshot after installation leaves the checkpoint unchanged. Epoch rotation, new issuance and multi-source coherence are not explored. Complete-record binding and continuation-scope validity are assumed. The checker has no separate continuation-deadline field; its explicit time constraints are the source-age cap and child expiry. The deployed protocol must enforce every additional validity limit in Section 3.4.

Each operation records its phase, captured generation, expiry, debit and effect count. The phases distinguish absence, evaluation, admission, dispatch, executed or cancelled outcomes awaiting a receipt, and sender closure. Evaluation captures an eligible local view. Admission consumes the unique slot and persists the debit and unresolved record. An effect can occur only from the dispatched phase and with the configured final guard. A terminal receipt changes sender knowledge without executing another effect. Cancellation creates a terminal receiver state before closure. Thus receiving a receipt and obtaining its represented outcome are distinct transitions.

The explorer nondeterministically interleaves ticks, source withdrawal, status delivery, operation transitions, and one crash followed by restart. A crash disables gate transitions but preserves all modelled durable state. Ticks and the source decision can occur while the gate is down. Restart restores the same absolute clock and checkpoints; uncertainty in a real clock is outside this finite case. Duplicate dispatch, timeout and rejected replay are represented as identity-preserving self-loops when they leave state unchanged. Execution or receipt delivery may be postponed indefinitely. No fairness condition or eventual-completion claim is imposed.

The message abstraction retains one outstanding logical execution request and one terminal outcome per identifier. There is no explicit network queue. It does not enumerate arbitrary byte strings, an unbounded transport queue, message multiplicities with distinct histories, or independently crashing sender and receiver replicas. Collapsing identical duplicates is valid for the checked idempotent transition definition; it is not evidence that an implementation safely bounds memory under a duplicate flood. These limits keep the checked graph finite and state exactly which behaviours the enumeration covers.

4.2 Monitors and connection to the proofs

The checker compares captured and installed generations at admission and final effect. It also enforces the absolute lease expiry. A monitor records a violation if an effect occurs with a stale installed generation or at a tick outside the original source-age cap. Other monitors reject duplicate effects or debits, an admitted phase without a durable debit, and an effect lacking the corresponding terminal state. Sender closure without terminal evidence is prohibited. Pending membership is the operation's admitted, dispatched or terminal-awaiting-receipt phase; closure retains the audit and debit fields.

These monitors correspond to the conclusions of Propositions 1–3, specialised to this state space. The generation monitor checks the local exclusion boundary, not the truth of a status assertion. The age monitor checks preservation of the initial cap through renewal and restart. Because no new active heartbeat is generated, it tests a stronger fixed-cut restriction than a deployment with legitimately refreshed status. The persistence monitors evaluate the model's transition discipline. Atomic effect and outcome persistence are built into that discipline; the checker does not establish atomicity for a real resource.

We enumerate the reachable-state graph using breadth-first search until its work queue is empty. Exact immutable state tuples determine visited-state equality. There is no depth cutoff or probabilistic sampling. The finite variable domains provide the bounds. An enabled transition instance is a labelled successor examined from one reachable state; distinct labels and self-loops are counted even when they reach the same state. Maximum shortest-path depth is a graph diagnostic, not an execution-time bound. Cyclic duplicate delivery permits arbitrarily long traces without adding states.

Table 4: Exhaustive baseline reachable-state results.

Baseline	Reachable states	Enabled transition instances	Maximum shortest-path depth	Monitor violations
One operation identifier	477	1,487	12	0
Two operation identifiers	6,651	26,905	17	0

Both baseline enumerations reached a fixed point. The second includes interleavings of two operations sharing one credential and gate state. It therefore checks that withdrawal, restart and per-operation persistence interact under this small shared-dependency workload. It does not justify a cutoff theorem extending two identifiers to arbitrary populations. The general invariant arguments are those in Section 3 under their explicit premises; the finite enumeration provides an independently inspectable executable instance and counterexample search.

Reading the result. The checker visits every state reachable under its small declared rules, rather than drawing a sample of executions. That makes zero violations informative for this model. It does not mean that every possible deployment failure was tested. Independent receiver crashes, cryptographic parsing and a real release adapter are outside the enumerated machine. Engineers can use the omission traces as regression tests, but must also show how their own storage, concurrency and effect boundary preserve the assumptions used by the model.

4.3 Removing individual controls

We constructed three protocol variants by removing one obligation at a time. The underlying clock, source state and operation domains were retained. For these variants, search stopped at the first monitored violation. Their reported trace lengths are shortest within the breadth-first exploration and the selected transition definitions. Their partial discovered-state counts are not exhaustive coverage results. Each complete state trace is supplied with the source package.

Table 5: Shortest counterexamples after removing one control.

Omitted obligation	Shortest violating trace length	Result
Final dependency-generation comparison	6 transitions	A previously admitted witness authorises an effect after local withdrawal installation.
Parent-age cap during renewal	7 transitions	A new child expiry authorises an effect after the original source-age limit.
Terminal evidence before sender closure	3 transitions	A timeout removes unresolved work without evidence of any terminal outcome.

Without the final generation comparison, the source first withdraws, but the old active snapshot remains locally available. The operation is evaluated and admitted against generation zero. Withdrawal is then delivered and generation one installed. Dispatch follows, and the altered final guard permits the effect before its lease expires. The violation is the receiver's use of the old witness after local installation. A generation comparison only at admission would not catch this interleaving. Conversely, a complete current reevaluation at the final boundary could replace the incremental comparison; the experiment removes the only such safeguard in this machine.

The stale-renewal trace evaluates and admits at tick one with expiry two. After the clock reaches two, the request is dispatched. The altered renewal rule sets a new child expiry to three without new authoritative evidence. Execution at tick two then violates the parent's strict age limit. The source need not actually withdraw for this monitor failure: the assertion is too old to justify ordinary execution under the declared contract. This counterexample isolates freshness renewal from incident detection and publication delay.

The timeout variant evaluates and admits one operation, then incorrectly closes its sender record. No execution, cancellation or receipt occurs. This is already an accounting violation, even if a later effect never happens. A demonstration that the timeout sometimes precedes actual cancellation would not repair the rule. Valid closure requires the contractual evidence in every case where unresolved state is removed.

A further supplied trace passes every baseline monitor while a withdrawal remains in transit. The operation is evaluated, admitted and dispatched; the source then withdraws, and the receiver commits before the update is installed or the age cap expires. This allowed trace is an explicit limit of the result. Zero local monitor violations do not mean zero releases after authoritative withdrawal. They establish the checked local and expiry properties, whose residual exposure is analysed separately.

4.4 Packed-word update interleavings

A separate exhaustive model contains one two-bit semantic word and two authorised writers. Each writer reads the word once and commits its own irreversible revocation bit once. The

invariant requires every already committed bit to remain set. Writer stages and saved read values are explicit, so the model exposes stale read-modify-write races rather than treating each client workflow as one indivisible operation.

Bitwise OR sets selected bits without clearing existing ones. With atomic semantic OR at commit, the graph contains 13 reachable states and 14 enabled transition instances, with no invariant violation. With whole-word replacement computed from the earlier client read, it contains 14 states and 14 transition instances. The four-transition counterexample has both writers read zero, one commit its bit, and the other overwrite that word with a value containing only its own bit. The first committed withdrawal is lost. This is an exact interleaving result, not a measurement of transaction gas or chain finality.

The model establishes why transaction ordering alone does not protect stale whole-word updates. It does not verify an Ethereum contract: access controls, setters, upgrades, epoch semantics, replay checks and bit ordering remain implementation obligations. Atomic OR applies to the irreversible semantic word. A masked wire word requires its stated decoding relation; blind OR on that representation is not justified by this model.

4.5 Reproduction and remaining refinement work

The supplied Python source uses only the standard library. Running `python protocol_check.py --output` results regenerates the complete baseline counts, omission witnesses, packed-word graphs' counts and the allowed in-transit-withdrawal trace. The result manifest records Python 3.12.14, all finite bounds and the source digest from the Secure Hash Algorithm with a 256-bit output (SHA-256). The checker stores counterexamples as structured states and labelled transitions, permitting independent reconstruction of each failure. Runtime is not treated as a protocol-performance result.

A deployed implementation needs a refinement argument relating its actual storage, scheduling and effect traces to this abstract machine. Refinement mappings are an established method for relating implementation and specification behaviours [12]. No such mapping from production code is proved here. The required argument needs validated evidence reads, non-repeating generations, persistent absolute expiry, failover-safe identity and a justified last-effect boundary. It also requires operational measurements of publication, service and receiver delay. The current experiment supplies none of those measurements. It identifies enforceable contracts and shows, within a declared finite model, how specific omissions break them. Larger fault domains and richer message encodings can strengthen testing, but do not remove the need to justify the abstraction itself. Section 11.5 turns these remaining obligations into an implementation test sequence.

5 Standards-bound implementation profiles

5.1 Normative baseline and study-specific requirements

A reproducible protocol must identify more than an algorithm name. Its profile binds the accepted object format, authority interpretation, exact signed bytes, decoding limits, status semantics, publication state and receiver behavior. The standards cited here supply selected components. Requirements introduced by this article are identified as profile requirements; they are not attributed to those standards. A successful test of the profile likewise does not establish general conformance to every document on which it depends.

The reference baseline uses an unmasked status representation and origin-authenticated assertions. Experimental decoys and masks require distinct profile identifiers in any deployment. The dated W3C v1.0 Recommendation and v1.1 draft remain separate configurations [5, 6]. A deployment experiment must record the retrieved specification version and implementation revision. A later editor’s draft cannot silently change the interpretation of an earlier trial. This is particularly important for terse references, because a change in indexing or list length can alter both correctness and workload.

Table 6: Implementation profile layers and obligations.

Profile layer	Fixed input	Requirement introduced or selected here
Signed object	Media type, suite, key identifier and signed payload.	Unambiguous bytes, bounded parsing, domain separation and protected verification result.
Status representation	Dated standard, purpose, width, allocation and index.	Exact semantic interpretation and unsupported-profile rejection.
Authority and freshness	Recognized source, epoch, version and watermark assertion.	Scope-bound completeness and no freshness reset through redistribution.
Publication backend	Mirror object or selected ledger state.	Authenticated local installation and explicit rollback/recovery behavior.
Experimental privacy	Allocation, mask, lifecycle and observer.	Separate conformance status and complete transcript accounting.
Operation envelope	Complete record, audience, identity and expiry.	Receiver-side effect control and contract-valid terminal closure.

5.2 Standard bitstrings, decoding and status interpretation

Bitstring Status List v1.0 represents status in a credential carrying a bitstring compressed with GZIP (GNU zip), then encoded as an unpadded base64url multibase value. Index zero is the left-most bit. Its generation algorithm starts from zero bits and assigns the status of issued credentials. Its optional time-to-live (ttl) property concerns refresh timing and does not replace the list’s validity period [5]. GZIP is a lossless compression format; base64url uses the Uniform Resource Locator (URL)-safe Base64 alphabet [13, 14]. These encoding operations are separate from signature verification and status interpretation.

The receiving profile limits encoded length before allocation, decoded length during decompression, graph expansion and cryptographic work. It rejects truncated input, unsupported widths, out-of-range indices and an unexpected status purpose. The relevant question is not merely whether a parser produced bytes, but whether those bytes have one accepted interpretation under the bound profile. Duplicate or conflicting status references require a declared resolution rule. A holder cannot choose a less restrictive purpose when policy requires revocation and suspension to be considered.

For comparison with Article A, a fixture may use 131,072 one-bit positions, or 16,384 raw bytes. That population is a declared experimental size. The raw size, compressed size, encoded size and signed-envelope size are recorded separately. A compression ratio therefore does not become a total transport ratio unless every wrapper and publication path is included. Decoding cost also cannot be inferred solely from output size under adversarial inputs. A deployment service contract must account for bounded rejection paths as well as successful verification.

The status credential and the original credential may have different authoritative sources [5]. The profile consequently carries distinct recognition dependencies where needed. Recognizing the original issuer does not automatically authorize an unrelated status publisher. Distribution caches retain the authenticated source and do not replace it with their own transport identity. Re-signing cached data, when permitted, records the additional attestation role rather than erasing the original authority boundary.

5.3 Pinned terse reference profile

The 24 September 2026 v1.1 draft defines an unsigned 32-bit terse index and fixes 2^{26} entries per list with 2^6 lists. Conversion supplies the status purpose as a verifier input [6]. For the selected index q , the profile therefore uses

$$j = \left\lfloor \frac{q}{2^{26}} \right\rfloor, \quad i = q \bmod 2^{26}. \quad (5)$$

A one-bit list at that size contains 8 mebibytes (MiB), where one MiB is 2^{20} bytes. This is a derived raw representation size, not a statement that every action requires a new transfer. A warmed cache may reuse an accepted version. Cold start, recovery and publication still face the actual object size.

The deployment profile requires tests at the boundaries 0, $2^{26} - 1$, 2^{26} and $2^{32} - 1$, plus rejected negative and oversized inputs. It binds the reconstructed locator, required purpose and resulting list identity to the same credential verification. Equivalent references must resolve to equivalent semantics. A successful integer conversion does not establish that the selected list is authentic, allocated or current. These are separate checks.

The terse variant is a sizing and interoperability extension, not a mandatory baseline for the core experiment. Benchmarking a small ordinary list against a large terse list without separating population and residency effects would confound the hosting comparison. A deployment experiment must therefore report list population, occupied positions and accepted version reuse alongside byte counts.

5.4 Experimental decoys and initial masks

The standard’s privacy discussion acknowledges decoys but leaves their algorithms outside scope and warns about distinguishable cover values [5]. This does not establish conformance for nonzero unissued positions or exclusive-OR (XOR)-decoded genuine states. The experimental profile must state its relation to the normative generation and processing algorithms. Naming an extension cannot override them.

In the masked experiment, semantic state is $x_{i,t}$ and wire state is $Y_{i,t} = r_i \oplus x_{i,t}$. Credential binding must authenticate the reader’s required index, epoch and r_i in a deployment. The operator’s complete allocation J remains private only under the declared threat model. If the initial public semantic state is known to be all zero, the initial snapshot reveals the full mask. Snapshot comparison also cancels a fixed mask. These are algebraic limits of this construction, not implementation defects that a stronger signature repairs.

A profile instance includes cover generation, reserve exhaustion, rotation and reuse rules. Irreversible genuine revocation is compared with one-transition decoys. New epochs cannot silently reactivate old credentials. All public writers, transaction arguments, event subjects and batch identifiers belong to the observer transcript when accessible. A private bitstring experiment paired with real-only public events is not the same privacy construction.

5.5 Packed ledger words and concurrency

Ethereum Request for Comments (ERC) 7506, titled Trusted Hint Registry, remained a Draft at inspection. It defines hint lists within namespaces, writable values and delegation facilities. Its `setHint` method accepts a `bytes32` value [15]. The use of that value as a 256-bit credential-status word is the present profile’s choice. ERC-7506 is not, by itself, a standardized packed bitstring with the monotonic update semantics required here.

Let C be the semantic word and F an authorized mask selecting entries to revoke. The irreversible transition is

$$C' = C \vee F. \quad (6)$$

The symbol $\vee$ denotes bitwise OR. The operation is performed against current contract state in the transaction path. Two clients must not read the same old word, modify different bits independently and then overwrite each other’s result. A replay nonce authenticates a distinct request but does not repair that lost-update race. Atomic semantic OR, or a correctly checked versioned update, addresses the state conflict.

For fixed masked publication, the corresponding wire transition is

$$Y' = r \oplus (C \vee F). \quad (7)$$

Applying OR directly to Y is not equivalent. Blind XOR toggling is also unsafe under retries because a repeated update can undo the earlier transition. If the contract does not know r , a bare masked word is insufficient to enforce semantic monotonicity; additional protected state or a construction-specific proof is required. Publishing that state may change the privacy objective. A deployment experiment must record the extra information rather than assuming cryptographic authentication makes it harmless.

Every mutation route is part of the invariant: ordinary setters, namespace owners, delegates, upgrades, initialization and epoch changes. Revocation of writer authority has its own ordering and freshness semantics. An unset word is not automatically evidence that all credentials are active. Accepted allocation and list metadata must first establish that the zero value represents initialized state.

For a standard-order 32-byte slice interpreted as a big-endian integer, bit i uses word $\text{floor}(i/256)$ and shift $255 - (i \bmod 256)$. This is the profile’s adapter derived from the standard ordering, not an additional W3C storage requirement. Required boundary fixtures test both directions across bytes and words. A least-significant-bit convention needs an explicit conversion; equality of numerical indices alone does not establish interoperability.

5.6 Resident state, authority and recovery

The ledger adapter binds accepted data to chain identity, contract identity, initialized epoch and selected state. Its local verifier distinguishes submitted, included and policy-accepted updates. A completed local bit read is constant-time only once the required representation is resident and decoded. It excludes synchronization, validation, decoding, recognition and installation work. The mirror adapter receives the same benefit from an already verified resident snapshot.

The authoritative completeness watermark remains a source assertion under the protocol contract. A block timestamp, selected finality rule or successful synchronization does not manufacture it. During an unbounded partition, both backends eventually exhaust their accepted-age allowance

unless new authoritative evidence becomes available through another supported path. Recovery preserves monotonic checkpoints and unresolved operations. It cannot substitute a convenient older active view merely because that view is available.

Both adapters expose the same installation interface to the protected controller. This isolates hosting effects from policy differences. An independently authorized restrictive event may suspend reliance before durable ledger settlement, but the profile must not use the same shortcut to introduce an unfinalized grant. The authority and recovery obligations of this fast path are reported separately. Any claimed deadline includes detection boundaries, publication, delivery, validation and completed receiver enforcement as applicable; observed stage percentiles are not deterministic upper bounds.

5.7 Shared signals and receiver enforcement

The SSF and CAEP specifications fixed here were published on 29 August 2025. SSF Security Event Tokens (SETs) use JSON Web Tokens (JWTs), based on JavaScript Object Notation (JSON). These SETs require explicit event typing and prohibit the `exp` and `sub` claims. CAEP Credential Change includes the credential type `verifiable-credential` and change type `revoke` [7, 8]. The profile uses these semantics without treating an event as an execution lease. Expiry belongs to the separately bound evidence or operation contract.

The receiving profile authenticates issuer, audience, authority scope, subject mapping and event type before applying an event. It associates updates with the relevant list, allocation and epoch. Arrival order is not semantic order: an old revocation, reinstatement or prior-epoch message can arrive after newer state. Idempotent processing and reconciliation resolve the accepted version. If an event is only a notification, authoritative state is retrieved before it supplies the required premise. An event with independently sufficient authority can trigger the restrictive action explicitly permitted by policy.

Push delivery under Request for Comments (RFC) 8935 uses an accepted-response acknowledgement for successful transmission [16]. The protocol separately records receipt, accepted-state installation, protected local invalidation and remote effect enforcement. These are distinct milestones. A subscriber or broker acknowledgement is not a terminal receipt from the resource. Lost events are repaired by reconciliation, while expiry limits continued ordinary reliance. Neither mechanism proves that a dishonest source reported all decisions.

5.8 Signature sizes, transport and algorithm migration

Federal Information Processing Standard (FIPS) 204 defines the Module-Lattice-Based Digital Signature Algorithm (ML-DSA). It was published on 13 August 2024; the NIST publication page records a 31 July 2026 planning note about potential corrections [17, 18]. A profile pins its implementation and applicable corrections. The standard's signature sizes are shown below. Encoded sizes are calculated for unpadding base64url and exclude all wrappers [14, 17].

Table 7: Signature sizes and calculated unpadded encoding lengths.

Parameter set	Raw signature bytes	Encoded signature characters
ML-DSA-44	2,420	3,227
ML-DSA-65	3,309	4,412
ML-DSA-87	4,627	6,170

For an n -byte value, the unpadded encoded length is $\text{ceil}(4n/3)$. Signature, public-key, proof-wrapper, context and transport costs are counted on the links that carry them. A shared cache can reuse verified key material where the profile permits it; it cannot omit transmitted bytes from network accounting. Article A’s 1,024-byte illustrative envelope was a sensitivity parameter, not an ML-DSA envelope.

The signature primitive does not define the complete credential format. The deployment selects the message representation, verification context, algorithm identifier, key reference and negotiation rule. Verification and signing must agree about any prehash mode. Deployment tests must include changed payloads, wrong contexts, unsupported suites, withdrawn keys and downgrade attempts. Hybrid signatures require an explicit acceptance rule; the word “hybrid” does not specify whether both proofs are mandatory.

Object authenticity also does not encrypt a channel or establish claim truth. Transport key establishment, transport authentication, ledger authorization and consensus may use different algorithms. Accordingly, replacing one credential signature does not establish system-wide post-quantum security. A deployment comparison must fix an interoperable signature baseline; broader algorithm migration is an explicit extension whose added bytes and service work must be measured separately.

5.9 Profile evidence and conformance limits

A deployment conformance package must identify every profile choice, accepted and rejected byte fixture, authority binding and decoding limit. Its refinement mapping must connect accepted objects to protected transitions. The supplied model package does not implement these wire formats or establish their conformance. A conformance failure must be separated from an unavailable dependency, an exceeded age cap and a violated trust premise. This distinction matters because a successful parser and a denied effect can coexist for different reasons.

The reference comparison therefore requires common authority, freshness and effect semantics before comparing backends. It does not require identical wire encodings when the designs inherently differ. Such differences are visible measurements. This preserves the intended causal question: whether a supported implementation can absorb the work needed for its trust and privacy requirements while maintaining the receiver’s enforcement contract.

5.10 Anonymous non-revocation proof adapter

AnonCreds implements accumulator-backed non-revocation proofs that hide the presented credential’s revocation index. Its roots in Hyperledger Indy do not make a ledger necessary to the current specification [19, 20]. Formal analysis establishes cryptographic properties under its stated model [21]. The adapter proposed here connects that mechanism to Section 3’s receiver contract. It is not implemented by the supplied checker or workload simulator. A cryptographic

membership witness, denoted w^{acc} , is distinct from the controller’s evidence witness $W_{v,g}$: the former supports a proof; the latter records the coherent premises, generations and expiry needed for an effect.

The accepted profile must establish that the hidden status handle belongs to the same authenticated credential whose claims satisfy the requested policy. Proving that some credential is unrevoked is insufficient. The proof statement or its authenticated protocol binding must also fix the accepted registry, root and epoch, verifier challenge, audience and complete-record context. The construction must preserve the discovery and completeness contract when some premises remain hidden. These are proposed integration obligations, not claims that every AnonCreds presentation format supplies each binding directly. Treat the selected registry as a mandatory dependency ℓ . With Section 3.4’s protected clock, justified watermark and aggregate clock margin, the conservative pre-expiry guard requires

$$\hat{t}_v(t) - \omega_{v,\ell} + \kappa_{v,\ell}(t) < \tau_\ell.$$

A proof-generation timestamp does not advance $\omega_{v,\ell}$. A valid proof about an older root cannot override a newer installed withdrawal. Root authentication, rollback protection and freshness remain separate checks; a missing or stale witness justifies withholding an effect but does not, by itself, establish revocation.

Hiding the status index changes dependency processing. One profile may expose a scope-limited dependency handle only to the protected controller, with an authenticated, complete withdrawal-to-handle mapping bound to the same credential proof. This profile must document that party’s correlation capability. A stronger index-hiding profile instead invalidates all dependent decisions when the relevant registry epoch changes. It then requires fresh proofs or bounded lease termination. This broadens revalidation fan-out and reserved service demand. Neither profile removes durable operation tracking or the receiver’s last-effect check.

The implementation must distinguish issuer intent, pending publication, accepted registry state, local installation and completed enforcement. ACA-Py documentation explicitly separates pending revocations from publication; its batching option can defer the ledger write [22]. A holder notification is not evidence that a remote resource has stopped. Stopping must follow installed restrictive state or supported expiry, without waiting for holder cooperation. Witness refresh, proof construction and verification instead enter the recovery and re-admission budget. Finally, an ML-DSA signature on the root envelope does not upgrade the accumulator or proof system to post-quantum security. Appendix F specifies the additional evaluation work.

6 Matched workload and hosting evaluation

6.1 Executable scope and comparison contract

The workload study is an executable discrete-event model, not a benchmark of a deployed ledger, cryptographic library or industrial agent system. Its purpose is to expose the connection between publication, local work, protected enforcement and legitimate service. The model instantiates the control obligations of Article A [1] with stated timing constants. It does not infer those constants from production measurements. The code generates 6,144 finite incident runs: 128 seeds in each of 48 configurations. The same incident seed supplies arrival times, action labels, execution durations, reconciliation phase and initial queue work for both hosting paths and all credential decompositions.

Reading the comparison. Imagine replaying the same file-release workload twice, changing only the publication path and its declared timing rules. Both versions then face the same withdrawal and requests, so differences are easier to attribute to the modelled change. This is the purpose of pairing. It does not make the chosen delays representative of an operational network. The many requests within one incident also share its conditions; counting each request as an independent experiment would exaggerate the evidence.

Both paths use verified resident state for ordinary decisions. One receives origin-authenticated snapshots through signed mirrors. The other receives ledger state after the selected inclusion and finality delays. A resident lookup creates no remote per-decision request in either model. Background distribution remains visible. The comparison consequently does not assign a unique lookup-privacy benefit to a local ledger. It isolates the consequences of the stipulated publication and failure paths under the same authority, freshness and receiver policy.

There is one authoritative withdrawal at time zero. Runs end at 80 seconds. Proposals follow a seeded Poisson process of rate 30 per second; 15% request the withdrawn scope and 85% remain legitimate. Their execution delays are sampled uniformly between 0.05 and 0.30 seconds. Each legitimate proposal has a 0.60-second deadline. These distributions are workload assumptions. They are neither measured agent behaviour nor fitted incident laws. A protected receiver repeats the eligibility check at the effect boundary. A proposal that passed admission can therefore be rejected before execution. Terminal effects within the horizon are recorded immediately; missing remote receipts are not simulated in this experiment.

6.2 Equivalent policy decomposition and reserved work

The conceptual semantic policy is a conjunction of twenty fixed predicates, grouped into one, ten or twenty credentials. The workload program represents each request by a Boolean semantic label and does not construct or evaluate those credentials. Policy equivalence across decompositions is imposed by the fixture: each comparison uses the same permitted-action labels and withdrawal incident. Thus $k_{pop} = k_{dec} \in \{1, 10, 20\}$ only in these fixtures. Their equality is a chosen comparison; population multiplicity and per-decision evidence demand remain distinct quantities in the general model. The model charges every affected-record job for complete reevaluation through a stipulated work function. It does not execute a credential evaluator. The configured cost is

$$w(k_{dec}) = 2 + 0.2k_{dec} \quad \text{abstract work units.} \quad (8)$$

This gives 2.2, 4 and 6 work units per job. It is not a measured signature cost, a codec byte multiplier or a claim that all implementations scale linearly. The simulation increases per-job work once. It does not also multiply the job arrival rate by the same credential factor. An incremental verifier with a different sound dependency strategy needs a separately specified work function.

A first-in, first-out (FIFO) control service supplies a stipulated $R_{srv} = 40$ work units per second. Initial work ahead of the batch is uniformly distributed between 10 and 30 units. A batch contains either eight or 32 affected-record jobs. The record observed by the release workload occupies a uniformly sampled rank in that batch. All draws are paired across configurations. Within a supported detailed-service envelope, its completion time after validated receipt is $(Q + r_k w(k_{dec})) / R_{srv}$, where r_k is its rank.

The 128-unit value is a detailed-service envelope threshold, not a claim that every recovery batch fits a queue of that size. When $Q + nw(k_{dec}) > 128$, validated receipt causes an atomic broad hold on ordinary effects. This is an explicit protected transition of the abstract model.

Detailed recovery then drains the finite batch before unaffected work reopens. The largest recovery workload is 222 units. A real implementation must reserve and bound the broad-hold path itself; constant-time model assignment does not establish a hardware execution deadline. The model never deletes a control job to preserve availability.

This distinction prevents a misleading interpretation of overload. A slower detailed path can trigger an earlier conservative stop. Fewer harmful releases can then coexist with greater legitimate loss. Such a result does not show that higher credential multiplicity improved selective enforcement.

6.3 Publication, renewal and failure conditions

The mirror path makes the first authenticated withdrawal available after 0.75 seconds. The ledger path uses 0.5 seconds for submission, 1.5 for inclusion, 3 for selected finality, and 0.25 for propagation and local validation: 5.25 seconds in total. These are stipulated model constants. They are not Ethereum performance estimates or worst-case public-network guarantees. A status heartbeat is generated every five seconds. Both paths reconcile on the same ten-second period, with a uniformly sampled phase. Delivered push permits retrieval when the corresponding authoritative state is available; it does not bypass the ledger's selected finality rule.

At time zero, the resident authoritative watermark is $\omega = -5$ seconds. The maximum accepted age is 20 seconds and the clock margin is 0.5 seconds. Child execution leases last at most six seconds, while their expiry remains capped by the source's authoritative age. In the absence of updates,

$$\begin{aligned} e_g &\leq \min\{t_{admit} + 6, \omega + 20 - 0.5\}, \\ H_{fresh} &= 20 - 5 - 0.5 = 14.5 \text{ s}. \end{aligned} \tag{9}$$

Renewal cannot move the second cap. The receiver enforces strict pre-expiry acceptance. This is a conservative abstract clock check, not an additional source of freshness. The four failure conditions have precise meanings:

Table 8: Failure conditions in the matched workload model.

Condition	Distribution and fallback assumptions
Delivered push	Authenticated state becomes available on the specified publication path; the protected service applies it.
Lost push	All update signals are absent; periodic reconciliation retrieves available authoritative state.
Origin service or Domain Name System (DNS) outage	A preconfigured independent mirror remains reachable, adding 0.5 seconds to mirror delivery. Ledger publication and peer connectivity remain intact.
Complete inter-domain partition	Both update paths and signals are unavailable throughout the run; the resident copy remains readable, and parent-age expiry stops ordinary effects.

The origin-outage result depends on the independent routes just specified. If the ledger publisher also loses its only route, or all mirrors share the failed dependency, a different fault has been injected. Neither architecture is exempt from correlated infrastructure failures. A received heartbeat cannot install a newer completeness watermark ahead of unfinished withdrawal

processing; the implementation model delays or coalesces it until the initial batch has been applied.

6.4 Synthetic incident outcomes

The following table reports the twenty-credential, eight-job comparison. Delay is the model's time after authoritative withdrawal beyond which the named revoked scope cannot execute. Harm counts executed prohibited releases before that boundary. Legitimate loss is the fraction of legitimate proposals not completed before their own deadline, among proposals whose deadlines fall within the observation horizon. Later deadlines are reported as censored. The 10-second target is an operational comparison threshold, separate from the 14.5-second expiry guarantee.

Table 9: Synthetic workload outcomes for twenty credentials and eight affected-record jobs.

Failure condition	Hosting path	Mean enforcement delay (s)	Mean prohibited releases	Legitimate loss (%)	Runs exceeding 10 s (%)
Delivered push	Signed mirror	1.955	7.945	0.000	0.000
Delivered push	Ledger model	6.455	27.578	0.000	0.000
Lost push	Signed mirror	6.779	29.656	0.000	22.656
Lost push	Ledger model	11.535	50.570	1.214	72.656
Origin/DNS outage	Signed mirror	2.455	10.359	0.000	0.000
Origin/DNS outage	Ledger model	6.455	27.578	0.000	0.000
Complete partition	Either path	14.500	63.789	81.967	100.000

All 6,144 runs satisfy the executable consistency checks: no named harmful effect occurs at or after its modelled enforcement boundary; every boundary is at most 14.5 seconds; complete partitions stop at that cap; and neither path makes a remote request for an ordinary resident lookup. These checks establish properties of this fixture. They do not validate the assumptions of a production gate or establish a general protocol proof.

In the eight-job delivered-push mirror cells, raising credential multiplicity from one to ten to twenty raises mean enforcement delay from 1.506 to 1.719 to 1.955 seconds. The change follows the configured work function while policy and incidents are fixed. With 32 jobs, the ten- and twenty-credential profiles instead exceed the detailed-service threshold and invoke the broad hold. For twenty credentials, the delivered-push mirror path stops at 0.75 seconds but loses 6.871% of otherwise legitimate completions. The corresponding ledger path stops at 5.25 seconds and loses 6.900%. These percentages describe the finite model, including the placement of legitimate arrivals relative to the hold. They are not estimates of a universal architecture trade-off.

Engineering interpretation. When detailed reevaluation becomes too expensive, the model stops a wider set of ordinary releases while it catches up. That broad hold can reduce prohibited releases even though selective processing has become slower. It can also block files whose permissions remain valid. A lower harm count alone would therefore give an incomplete account

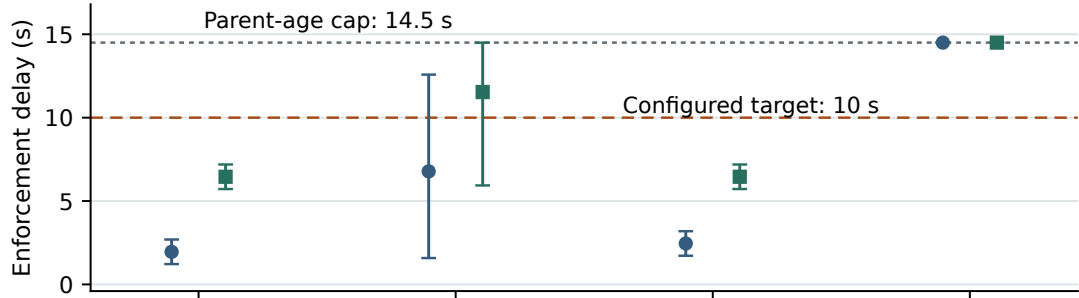
of the change. The implementation must record both kinds of outcome and establish that the broad-hold control itself remains timely under the overload being tested.

Lost push can defer application until after the old parent watermark expires. The ensuing gap explains legitimate loss without relaxing the age rule. Complete partition has the same result for both backends because neither obtains new evidence. A readable local ledger remains useful for authenticated historical state, but it cannot justify ordinary acceptance after this policy's freshness budget has ended.

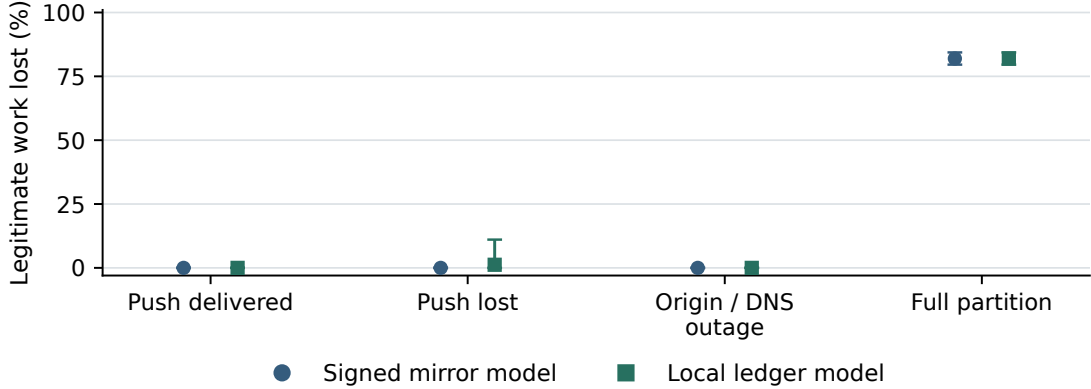
Hosting and freshness in paired synthetic incidents

$k_{pop} = 20$; eight affected-record jobs per batch; 128 paired incidents per fault

A Delay from authoritative withdrawal to enforcement



B Loss of legitimate work before its deadline



Points: incident means. Whiskers: incident minimum-maximum, not confidence intervals.

Both models use resident reads; differences follow the configured publication and fault rules.

Figure 3: Hosting and freshness outcomes in the completed synthetic incident model. Each fault cell contains 128 paired seeded incidents with population multiplicity $k_{pop}=20$ and eight affected-record jobs per batch. Points are incident means and whiskers are observed incident minima and maxima, not confidence intervals. (a) Time from the authoritative withdrawal decision to the tagged scope's modeled last-possible acceptance boundary, with the configured 10-second target and 14.5-second residual parent-age cap. (b) The fraction of legitimate proposals not completed before their deadline, excluding proposals whose deadlines lie beyond the observation horizon. Both hosting models perform resident local checks without external per-decision reads. The signed-mirror and local-ledger differences arise from their configured publication, reconciliation and fault rules; the chart does not estimate a deployment effect or demonstrate universal superiority of either hosting option. Full partition achieves the same expiry-based enforcement cap while losing approximately 82 percent of legitimate work in this workload.

6.5 Byte boundaries, interpretation and replication

Byte counts are retained separately from work. The fixture uses explicit synthetic interface constants: 4,096 bytes per mirror snapshot, a 1,536-byte ledger update batch, 512-byte later checkpoints, and 512-byte delivered signals. These are accounting placeholders, not standard encodings. They exclude signatures, transport headers, initial synchronization, consensus gossip, request traffic and retransmission. Consequently their difference cannot support a claim that the ledger consumes fewer total network bytes. Article A's codec results [1] remain inherited representation measurements; they are not substituted into $w(k_{dec})$.

The incident is the paired unit. Requests within one incident share the same withdrawal, queue and reconciliation phase and are not independent replications. The released files retain all 128 seeds per cell, including deadline exceedances and horizon censoring. The main table reports arithmetic means and named loss fractions. Figure 3 uses incident ranges to show model variability; these are not confidence limits on a deployment. The fixed seed set is an exploratory reproducibility fixture, not a completed confirmatory experiment with a population sampling claim.

The source, configuration, incident records, grouped summaries and paired contrasts are supplied as separate files. A deployment study must replace stipulated publication and work constants with supported measurements, qualify any hard service claims, include actual signed-object and synchronization costs, and repeat the receiver tests under the implementation's fault envelope. Until those steps are completed, the valid finding is narrower: the executable model makes the hypothesised work-to-enforcement path explicit, and it shows why availability, prohibited effects and publication costs must be reported separately.

7 Causal decoy generation and full-transcript membership privacy

7.1 Objective, allocation and observation boundary

The privacy question is whether an observer can identify which undisclosed positions represent genuine credentials. It differs from hiding a known credential's status, concealing an issuer's total activity, or preventing verifier-query tracking. A holder who discloses its credential index identifies that position. Local status reads address a different observation channel. CRSet studies issuer-activity privacy using a distinct revocation construction [9]. The analysis below concerns a fixed-cohort membership experiment and does not transfer that construction's guarantees.

Let a cohort contain L one-bit positions, exactly M genuine entries and $D = L - M$ decoys. Thus $d = D/L$ and $\alpha = M/L = 1 - d$. The allocation $J \in \{0, 1\}^L$ is chosen uniformly among vectors with $\sum_i J_i = M$. The operator knows J ; a credential recipient learns its own authenticated index. The initial experiment discloses L, M but no individual label. This is a dependent fixed-count prior, not a collection of independent Bernoulli labels. Side information Z must be declared before stating a guarantee. If some labels are disclosed and the remaining allocation stays uniform, the residual counts determine the remaining prior. A disclosed genuine position has posterior one and is outside the unknown-membership objective.

Write $x_{i,t} = 1\{T_i \leq t\}$ for an absorbing semantic transition at publication tick T_i . A genuine transition denotes authoritative withdrawal; a decoy transition is cover. Tick zero has $x_{i,0} = 0$. The wire value is $Y_{i,t} = r_i \oplus x_{i,t}$, with either $r_i = 0$ or an independent fixed random mask. Irreversible revocation permits one semantic transition. Repeated decoy reversals would distinguish them from this genuine lifecycle. No entry is reused within the cohort.

The observer receives every published whole-vector snapshot, its ordering and time, the declared common writer, profile and epoch, and all included side information. A ledger implementation must additionally account for senders, calldata, event fields, word updates, fees and batching. In the numerical experiment, a single fixed writer and deterministic publication schedule supply no label-dependent metadata. Complete snapshots reveal each transition tick or censoring after the horizon. Consequently the vector of observed transition categories is sufficient for that experiment's transcript. This reduction is not valid when omitted metadata identifies genuine events.

7.2 A causal candidate under a common environment

Let E be an exogenous environmental process independent of J and the per-position random innovations. Before interval t , the operator obtains a predictable probability $q_t \in [0, 1]$, measurable from the environmental information already available. A deployment interpretation could be a declared hazard class informed by a common incident indicator. That interpretation requires evidence; the numerical schedule below is a constructed model. Conditional on the relevant environmental history, each still-active genuine credential is assumed to undergo an independent authoritative withdrawal with probability q_t . The operator does not choose those genuine decisions. It samples an independent Bernoulli variable with the same probability for each unflipped decoy.

The implementable cover procedure is:

1. Create and authenticate a fixed cohort and its epoch. Allocate genuine indices randomly and retain the full allocation privately.
2. Before interval t , obtain q_t from the available environment. At its publication boundary, ingest genuine withdrawals through the authoritative path; do not fit the current cover probability to those newly observed withdrawals.
3. Independently sample cover transitions for unflipped decoys using q_t . Persist genuine and cover semantic changes with the same absorbing-state rule.
4. Publish the resulting vector through one declared writer path and schedule. Keep reasons, identities and label-specific event channels outside the protected public transcript, or include them in the model.
5. Preserve spent entries until authenticated cohort retirement. Retire only under the credential-lifecycle and receiver-freshness contract; exhaustion never permits a silent reset.

The operator and observer need not have the same environmental information. A common shock may be revealed to the operator at the start of its interval while remaining latent to the public observer. The operator can then use the current shock state for cover without predicting it in advance. Analysis conditioned on that environment and analysis integrating it out are different experiments, both permitted under the independence assumptions. An incident signal revealed publicly later is an additional observation; it cannot be ignored in a claimed full-history guarantee.

This construction is causal because cover at tick t uses only available environmental information and new randomness. It does not make arbitrary incident-driven revocations obey the assumed law. Observing a genuine burst and subsequently generating similar-looking cover is insufficient to establish equality of joint histories. Publication cadence also has a control cost: its maximum batching delay belongs in the supported withdrawal bound. Privacy does not authorize delaying an urgent genuine withdrawal beyond that bound.

Proposition 4 (conditional allocation privacy). Fix $Z = z$ for which the residual allocation prior is uniform with genuine fraction α_z . Suppose the complete observable transcript law, conditional on the entire relevant environmental path and $Z = z$, is the same for every admissible allocation. Require the conditional law of that entire environmental path given $Z = z$ to be independent of J , whether the path is public, hidden, or only partly disclosed. Then, for every protected index,

$$\Pr(J_i = 1 \mid O, Z = z) = \alpha_z. \quad (10)$$

Proof. For every admissible allocation j , integrate the common conditional transcript law against the environmental law, which is independent of J conditional on $Z = z$. The resulting transcript likelihood is the same for every allocation, so Bayes' rule leaves the allocation distribution unchanged. Public environmental information may be included in the transcript; the same argument applies. Marginal independence before conditioning is insufficient: selecting or disclosing side information can couple the environment and allocation. Under the candidate's conditional independent-clock construction, identical transition probabilities and absorbing-state rules give an identical joint law for every allocation. The common publication mechanism preserves that independence. Conditioning on observations then gives the stated marginal. $\square$

Exchangeability of positions alone is weaker than this premise. A process can remain permutation-symmetric while genuine credentials share an incident-specific dependence that decoys lack. The transcript then depends on which positions carry the genuine labels. Matching average flip counts or one-position lifetime distributions does not repair that dependence. A shared environmental shock is compatible with Proposition 4 when it affects both classes through the same conditional law and does not disclose labels. A genuine-only shock is not.

7.3 Exact inference with a fixed-count prior

For a declared environment, let $f_R(u)$ and $f_D(u)$ be the probability masses of a genuine and a decoy transition category $u \in \{1, \dots, H, \infty\}$. Under conditional independence given the allocation, an observed vector $\mathbf{t}$ has likelihood proportional to $\prod_{i: J_i=1} w_i$, where $w_i = f_R(t_i) / f_D(t_i)$ and the omitted factor is $\prod_i f_D(t_i)$. Let $e_m(\mathbf{w})$ be the elementary symmetric polynomial of degree m , with $e_0 = 1$. Assume positive masses on the compared support.

Reading the model. Suppose four positions contain exactly two genuine credentials. If one position becomes known to be genuine, the other three contain only one genuine position between them. Their possibilities are linked by the fixed total. The posterior calculation preserves this constraint while weighing the observed histories. Treating each position as an unrelated coin toss would discard information available to the observer. The larger calculation below applies the same reasoning when histories provide partial evidence rather than revealing a label outright.

Proposition 5 (fixed-count posterior and range bound). For $0 < M < L$,

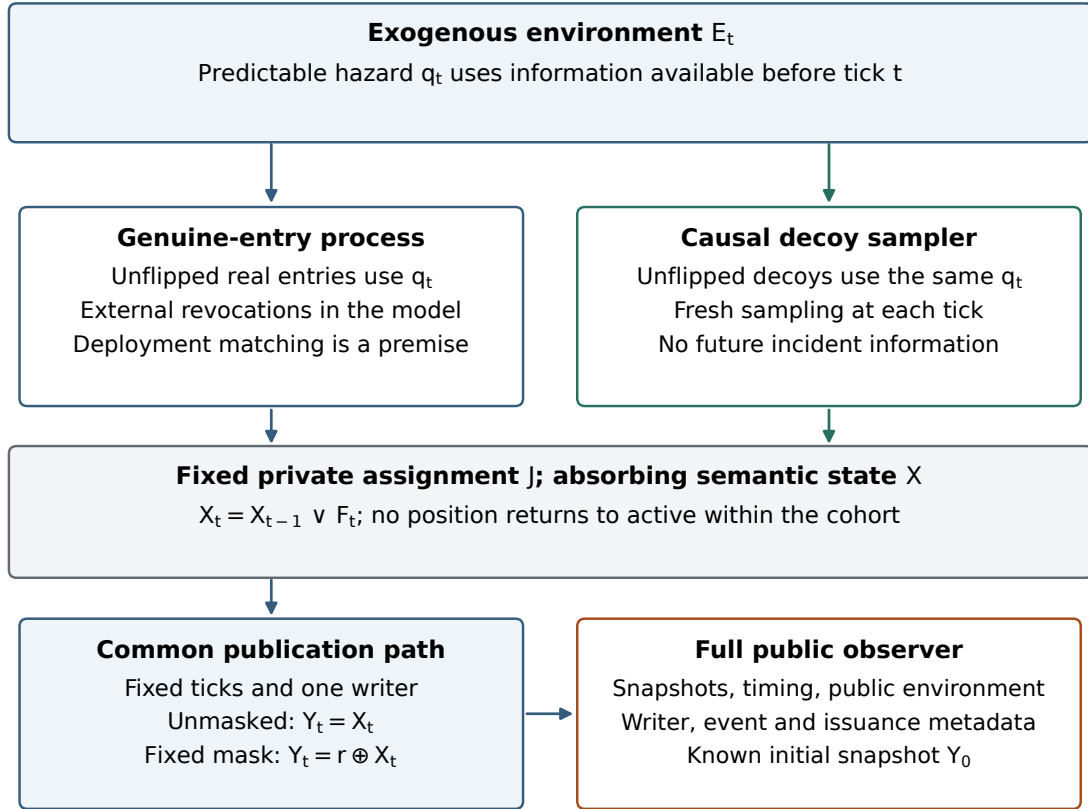
$$\begin{aligned} p_i = \Pr(J_i = 1 \mid \mathbf{T} = \mathbf{t}) &= \frac{w_i e_{M-1}(\mathbf{w}_{-i})}{e_M(\mathbf{w})}, \\ \frac{p_i}{1 - p_i} &= \frac{w_i e_{M-1}(\mathbf{w}_{-i})}{e_M(\mathbf{w}_{-i})}. \end{aligned} \quad (11)$$

If every possible coordinate ratio lies in $[a, b]$, with $0 < a \leq b < \infty$, then

$$p_i \leq \frac{ab}{ab + (1 - \alpha)a}. \quad (12)$$

Causal decoys and the public observation mechanism

Synthetic fixed cohort: 64 positions; 32 genuine and 32 decoy entries



Known $X_0 = 0$ gives $Y_0 = r$ and $Y_t \oplus Y_0 = X_t$: the fixed mask adds no secrecy.

Full-law independence of J suffices for exact membership privacy.

Negative controls: unequal hazards, genuine-only common shocks, revealing metadata.

Figure 4: Causal decoy construction and observation boundary. A fixed cohort contains 32 genuine positions and 32 decoys. Available environmental information determines the tick hazard; a latent common shock becomes known to the operator when its interval begins. Under the matched model, genuine and decoy entries share the same conditional absorbing-state law and publication path. Independence of the complete observable law from allocation is sufficient for exact membership privacy; average-rate matching is insufficient. The operator does not control genuine withdrawal decisions. Known active initialization reveals the fixed mask through $Y_0 = r$, allowing exact recovery of semantic history. Unequal hazards, genuine-only shocks and label-revealing metadata are negative controls. This specifies the synthetic candidate, not a deployed standards-conforming service.

Proof. Summing likelihoods over all size- M genuine subsets gives $e_M(\mathbf{w})$. Subsets containing i contribute $w_i e_{M-1}(\mathbf{w}_{-i})$. Splitting the sum into subsets that contain or omit i gives the odds formula. For the remaining $L - 1$ coordinates,

$$\begin{aligned} Me_M(\mathbf{w}_{-i}) &= \sum_{|S|=M-1} \left(\prod_{j \in S} w_j \right) \sum_{k \notin S, k \neq i} w_k \\ &\geq (L - M) a e_{M-1}(\mathbf{w}_{-i}). \end{aligned} \quad (13)$$

Combining this inequality with $w_i \leq b$ bounds the posterior odds by $[M/(L - M)](b/a)$. Converting odds to probability proves the result. $\square$

The other positions matter because their labels must sum to the known remaining count. The independent-label expression $\alpha w_i / (1 - \alpha + \alpha w_i)$ is generally incorrect here. Likewise, a marginal path likelihood bound cannot be used unchanged as a bound for the complete cohort transcript. The range b/a supplies one conservative route under the proposition's factorized conditional model.

The implementation computes the symmetric polynomials in the logarithmic domain. Forward and backward recurrences evaluate all coordinate posteriors with $O(LM)$ arithmetic and storage. This exact calculation provides an oracle for the finite declared model. It is not a fitted attack whose failure could be mistaken for proof.

For a latent common shock S , compute the posterior within each branch and average using the branch posterior. Its unnormalized branch weight is

$$\Pr(S = s) \left[\prod_i f_{D,s}(t_i) \right] e_M(\mathbf{w}_s). \quad (14)$$

The common allocation factor $\binom{L}{M}^{-1}$ cancels. This mixture retains correlation induced by the shared shock; replacing it with independent draws from marginal one-position mixtures would define a different, less informative observer.

7.4 Approximate matching over complete histories

A deployment rarely supplies an exact lifetime law. Article A's likelihood-ratio condition can instead be applied to complete observation increments [1]. Fix the protected index, initial information $Z = z$ and prior $0 < \alpha_z < 1$. Let $k_{R,t}$ and $k_{D,t}$ be the conditional laws of the *entire* next public increment under hypotheses $J_i = 1$ and $J_i = 0$. They condition on all prior public observations. Each increment includes newly disclosed side information, not only a selected bit.

Proposition 6 (finite-history composition). Suppose, with a common dominating measure, every history reachable under the genuine-target hypothesis satisfies

$$k_{R,t}(o_t \mid o_{<t}, z) \leq e^{\varepsilon_t} k_{D,t}(o_t \mid o_{<t}, z), \quad \varepsilon_t \geq 0. \quad (15)$$

Then the complete-history likelihood ratio through H is at most $e^{\varepsilon_{1:H}}$, where $\varepsilon_{1:H} = \sum_{t=1}^H \varepsilon_t$, and

$$\Pr(J_i = 1 \mid O_{1:H}, Z = z) \leq \frac{\alpha_z e^{\varepsilon_{1:H}}}{1 - \alpha_z + \alpha_z e^{\varepsilon_{1:H}}}. \quad (16)$$

Proof. The probability chain rule expresses the joint density as the product of its conditional increment densities. Multiply the supported inequalities, then apply Bayes' odds formula. A real-only event with positive probability and zero decoy probability violates every finite envelope. $\square$

This is a one-sided membership bound with a declared prior, not differential privacy. For fixed-count cohorts the increment law includes what the other entries reveal about the target's remaining allocation probability. Bounds fitted to unconditional per-bit frequencies do not meet the premise. Nor does a finite test establish an all-history envelope, especially for rare incidents or adversarial interventions. A growing publication horizon requires control of accumulated likelihood, an independently proved finite-history restriction, or retirement under a declared objective. If an adversary can trigger withdrawals at selected known credentials, the comparison must include that intervention policy. A law calibrated only to passive periods cannot certify this active setting. Model fitting should therefore estimate candidate cover laws separately from the all-history premises; calibration, held-out prediction and privacy proof are different obligations.

7.5 Initial masks and the known-initialization observer

Suppose the public initial state is known active and r is independent of allocation and semantic histories. Then $Y_{i,0} = r_i$ and

$$x_{i,t} = Y_{i,t} \oplus Y_{i,0}. \quad (17)$$

The masked transcript therefore determines the semantic transcript exactly. Conversely, the semantic transcript and an independent random mask generate the masked one. The extra randomness conveys no information about J , so both transcripts produce identical membership posteriors. Snapshot differences also identify every transition location. Masking does not add privacy under this observer, even when an individual later snapshot appears random.

Engineering example. A genuine position starts semantically active, with value zero. If its fixed mask is one, the initial published value is one. After revocation, its published value becomes zero. An observer who retained the known-active initial snapshot can recover that transition by comparing the two values. Random-looking bytes therefore do not establish hidden history. The matched decoys must supply the membership uncertainty, while masking's extra bytes require a separate benefit under a clearly different observer.

This argument does not exclude another benefit under a different observer, such as one lacking initialization and all earlier snapshots. Such a claim must define what is hidden, the prior on semantic states and any correlated side information. Mask values delivered to holders are not operator-exclusive secrets. The experimental scheme also changes status decoding; nonzero unissued cover entries and masking remain outside the ordinary standards profile described in Article A [1, 5].

7.6 Reserve lifetime and representation costs

Conditional on a fixed environment, let $s_H = \prod_{t=1}^H (1 - q_t)$ be survival without a transition. Independent decoy clocks give

$$\begin{aligned} D_{unflipped}(H) &\sim \text{Binomial}(D, s_H), \\ \mathbb{E}[D_{unflipped}(H)] &= D s_H, \\ \Pr(D_{unflipped}(H) = 0) &= (1 - s_H)^D. \end{aligned} \quad (18)$$

A minimum reserve $D_{\min}$ has failure probability $\Pr[\text{Binomial}(D, s_H) < D_{\min}]$. Expected new cover transitions at tick t equal $Ds_{t-1}q_t$. Under exact matching, real survivors have the same survival law and the ratio of their expected survivor counts remains D/M ; reserve depletion alone does not make an observed survivor more likely genuine. These binomial statements condition on the environment. A random common shock gives a mixture of binomials, not a binomial evaluated at the mean survival probability.

Spent positions remain allocated. They cannot be recycled while old credential references are live. Continuing issuance requires new cohort and retirement rules whose timing and occupancy join the transcript. We do not claim that one fixed-cohort proof supplies an indefinite issuance system. Deterministic publication cannot hide issuer activity if external issuance mappings independently reveal it.

The useful one-bit payload is $M = (1 - d)L$. At equal fully occupied bundle population, the asymptotic raw capacity factor against one status per bundle is $k_{\text{pop}}/(1 - d)$, with the finite-list rounding from Article A. Matching adds expected transition work in proportion to decoy population, but not necessarily the same factor in transactions: batching can combine updates. Scheduled full snapshots incur fixed publication opportunities even when no bit changes. Bytes, updates, signing work, state retention and receiver processing must therefore be measured separately. None of these counts is an unconditional privacy or containment guarantee.

8 Synthetic privacy, reserve and publication results

8.1 Experiment and reproducibility boundary

The completed experiment evaluates Section 7's discrete-time model. It uses generated allocations and withdrawals, not deployment traces or measured issuer hazards. The observer knows the generative model, population count and public baseline schedule. It sees every scheduled snapshot, including the known-active initialization. It does not receive allocation labels or the latent common-shock indicator. Those values are retained only as evaluation truth. No trained classifier, fitted rate or incomplete attack is used to establish a privacy guarantee.

Each inference cohort has $L = 64$ and $M = D = 32$, hence prior $\alpha = 0.5$. The horizon is 180 publication ticks. The genuine baseline transition probability is 0.002 per tick, except ticks 60–69, where it is 0.02, and ticks 120–129, where it is 0.01. These probabilities are illustrative and have no claimed time calibration. Five configurations use 1,000 independent cohorts each. The master seed is 20261009, with separate declared streams for each configuration. Code, truth labels, event categories, posteriors and cohort-level outcomes accompany the paper.

The cases are: identical genuine and decoy hazards; decoy hazard multiplied by one half; decoy hazard multiplied by two; a common shock affecting both classes; and a common shock affecting only genuine entries. For the shock cases, one latent Bernoulli variable with probability 0.25 is sampled per cohort. If present, it changes the tick-90 probability to 0.8 for the affected classes. The operator can observe that environmental trigger when tick 90 begins. Public inference integrates over its unknown value. Its dependence across positions is retained by the exact mixture calculation.

For efficient simulation, transition categories are sampled from the probability mass function induced by the tick hazards. This produces the same distribution as running the causal absorbing process tick by tick. It does not implement a live status service or give its operator access to future incidents. The model separates an executable stochastic cover rule from the unresolved task of establishing a real deployment's genuine-transition law.

The posterior implementation uses logarithmic forward and backward recurrences. A separate brute-force check enumerates every size-three allocation for six positions across all five models. The maximum discrepancy is 6.11×10^{-16} . The matched cases numerically recover the constant posterior to within 4.50×10^{-15} . These checks validate the implementation on stated finite cases; Proposition 4 supplies the mathematical matching result.

For reporting, posterior scores are rounded to twelve decimal places to remove floating-point differences between theoretical ties. A position is predicted genuine when its score is at least 0.5. This is marginal Bayes classification under per-position error loss, at the declared numerical precision. It need not output exactly 32 genuine labels and is not a test of whole-allocation recovery. In the matched case all scores tie; the deterministic tie rule predicts every entry genuine, giving exactly 50% accuracy because the true count is fixed. Uncertainty is computed across independent cohorts rather than treating their dependent positions as independent replicates.

Reading the privacy result. With thirty-two genuine positions and thirty-two decoys, a rule that labels every position genuine is already correct half the time. Accuracy of 50% alone would therefore be weak evidence about privacy. Here the stronger result is that the exact model-based posterior remains one half for each protected position under the matching assumptions. The mismatch cases then show how the same observer gains information when those assumptions change. Neither result covers undisclosed metadata or an unknown deployment incident law.

8.2 Membership inference and shock leakage

Table 10: Membership inference under the declared cohort models.

Configuration	Observed marginal accuracy	Cohort standard error	Mean posterior Bayes accuracy	Mean maximum posterior per cohort
Matched hazards	50.00%	0.00 percentage points	50.00%	0.5000
Half-rate decoy hazards	60.09%	0.19 percentage points	59.82%	0.6668
Double-rate decoy hazards	62.27%	0.18 percentage points	62.53%	0.6540
Matched common shock	50.00%	0.00 percentage points	50.00%	0.5000
Genuine-only common shock	58.04%	0.42 percentage points	58.02%	0.6363

The posterior Bayes column averages $\max(p_i, 1 - p_i)$ over observed cohorts. It is the model's conditional expected correctness, whereas observed accuracy also uses the sampled truth labels. The listed standard errors describe Monte Carlo precision for observed accuracy under this fixed model. They do not quantify uncertainty about the model's suitability for an issuer. The highest posterior measures an observer's ability to select an especially revealing position; average accuracy alone can hide that risk.

Hazard mismatch produces information even when genuine and decoy processes share their overall shape. With half-rate cover, observed transitions favour genuine membership. With double-rate cover, a long-lived unflipped position becomes comparatively more plausible as genuine. More cover is therefore not a monotone privacy improvement. Proposition 5's global

posterior caps are 0.7321 and 0.7895 respectively for these positive-support models. They bound all event-category vectors under the declared hazards; they are distinct from the observed mean maxima in the table.

The matched common shock preserves posterior 0.5 because its full law remains independent of allocation. In contrast, the genuine-only common shock exposes a group relationship. Among the 274 sampled cohorts in which that shock occurred, marginal accuracy is 79.14% and the mean maximum posterior is 0.9976. The pooled accuracy of 58.04% therefore understates the high-confidence inference possible during affected incidents. The evaluator uses the latent truth only to report this stratum; the observer's posterior already integrates over the unknown shock. This distinction prevents an unobserved label from entering the attack as an input.

These results reject average-rate matching as an adequate validation criterion for the candidate. They do not prove resistance to a wider observer, intervention policy or publication mechanism. In particular, a real-only event message or distinct transaction writer can reveal membership outside this experiment's transcript model. Such an extra channel is not repaired by matching the vector's probability distribution.

8.3 Fixed-mask comparison and encoded bytes

Every one of the 5,000 inference cohorts is evaluated with the same semantic trajectory under unmasked and fixed-mask publication. For each masked history, exclusive OR with its public initial snapshot reconstructs every semantic snapshot exactly. There are zero reconstruction discrepancies. The transition locations also agree exactly. This implements the information-equivalence argument of Section 7.5. The finding is stronger than equal observed classifier accuracy: the complete input available to any semantic-history attack can be recovered from the masked publication.

A separate codec experiment uses 30 independent matched cohorts with $L = 131072$ bits. Each publishes all 181 snapshots, including initialization, using GZIP level 6, a zero timestamp field, unpadding base64url encoding and one Multibase prefix per snapshot. Both variants use identical semantic histories. The code fixes its pseudorandom stream and records library versions. The values below exclude signatures, transport headers, ledger transactions, retransmission and receiver processing.

At initialization, the unmasked encoded snapshot contains 69 bytes; the masked mean is 21877.2 bytes. At tick 60, the respective means are 14098.6 and 21877.0 bytes. By tick 180, the expected semantic one-fraction is 0.463587 and the mean sizes are 21877.1 and 21877.0 bytes. The sparse coding advantage has disappeared as these synthetic absorbing histories accumulate transitions.

Across the complete 181-snapshot history, mean encoded bytes per cohort are 2978690.1 unmasked and 3959757.2 masked, a ratio of 1.32936. This is a measured codec ratio for the specified generated histories. It differs from Article A's 40.03-fold and 1475.44-fold population comparisons because those fixtures use a sparse density and a different population boundary [1]. No direct comparison of these factors establishes a network, gas, privacy or harm multiplier. This experiment shows a positive masking cost with no added membership protection under the known-initialization observer; it does not establish an optimum codec or publication policy.

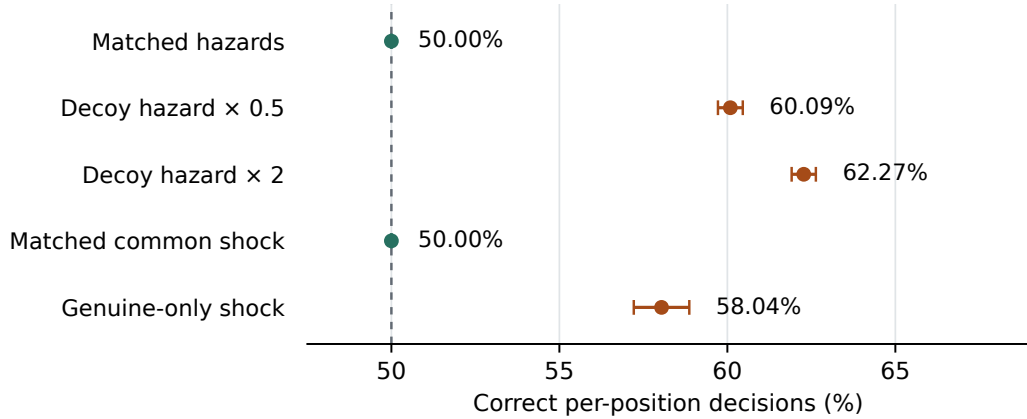
Figure 5 brings the inference, reserve and codec comparisons together. Its size axis uses kibibytes (KiB), each equal to 1,024 bytes.

Privacy, decoy reserve and encoded snapshot cost

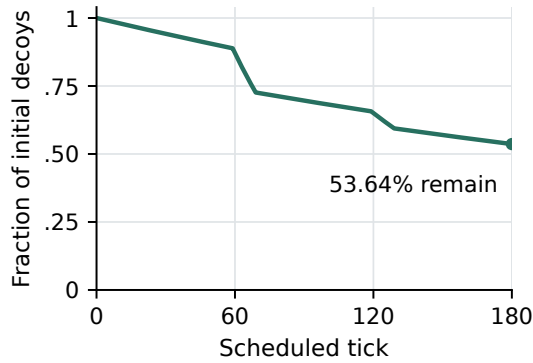
Synthetic model results; exact-model inference and measured generated-snapshot bytes

A Marginal Bayes classification accuracy

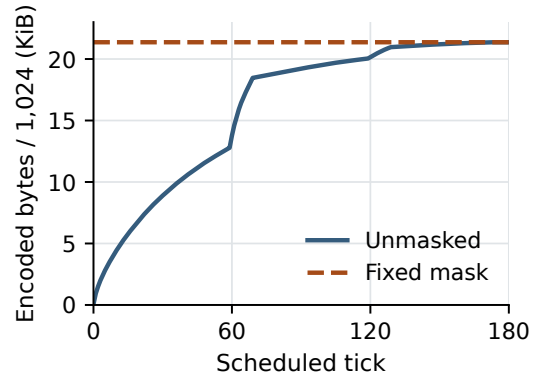
1,000 independent cohorts per case; 64 positions, with exactly 32 genuine entries



B Expected unspent decoy reserve



C Mean encoded snapshot size



A: bars are ± 1.96 cohort standard errors; the observer integrates over latent shock state.

B: baseline matched hazard. C: 30 paired 131,072-bit histories; GZIP + Base64url + prefix.

Known active initialization reconstructs both encodings to the same semantic trace.

Matched-case privacy depends on the stated full observation model; these are not field data.

Figure 5: Privacy, reserve and payload in the declared synthetic model. (a) Observed accuracy of the marginal Bayes classifier in 1,000 cohorts per case, with 32 genuine and 32 decoy positions. Predictions use posterior at least 0.5, with numerical ties rounded to twelve decimal places; they need not recover exactly 32 genuine labels. Bars are mean plus or minus 1.96 cohort standard errors. (b) Expected unspent fraction under the baseline hazard. (c) Mean encoded snapshot size in 30 paired histories of 131,072 bits, using GZIP level 6, timestamp zero, unpadded base64url and one Multibase prefix. Both profiles share the semantic trajectory. Fixed masking adds no privacy under known active initialization. Byte results exclude signatures, envelopes, transport, retries and ledger costs.

8.4 Reserve use and remaining implementation limits

For the baseline schedule, survival through tick 180 is $s_{180} = 0.536413$. An initial reserve of 32 decoys therefore has 17.1652 expected unflipped entries. The probability that every decoy has transitioned is 2.07×10^{-11} ; the probability of fewer than four remaining is 1.74×10^{-7} . A reserve of eight has only 4.2913 expected remaining and probability 0.2869 of falling below four. These exact binomial calculations show why a reserve threshold needs a stated workload and horizon. They are not observed service-exhaustion rates.

At this horizon, matched genuine and decoy entries each have expected transition fraction 0.463587. Their expected surviving proportions remain aligned; spent entries still occupy storage. The supplied reserve curves distinguish the design factor $k_{pop}/(1-d)$ from the bit-budget factor $k_{pop}L/(L-D)$ when $D = \text{floor}(dL)$. Whole-bundle capacity is obtained by dividing remaining genuine entries by statuses per bundle and rounding down; physical list-count comparisons additionally retain Article A's ceiling rules. Hidden common shocks require averaging conditional reserve distributions, not substituting a mean hazard into a binomial formula.

The implemented candidate is thus bounded in three respects: a fixed cohort, a declared genuine-event law, and a complete specified observer. It supplies a reproducible positive case and concrete mismatch failures. A production system additionally needs causal hazard estimation, incident-conditioned model checks, continuing-issuance and retirement rules, metadata-compatible publication, and a service budget that includes cover processing. Those obligations remain separate from the completed synthetic and codec evidence.

9 Continuing roots, uncertainty and conditional harm

9.1 From receiver enforcement to a propagation premise

Section 6 provides modelled resource-enforcement times under specified workloads. Those times do not automatically bound inter-agent influence. Article A [1] distinguishes resource effects from covered outgoing messages, including queued and in-flight communication. The same receiver checks must govern a stated communication channel before its supported delay can be used as Δ_{is}^M . A file-release gate alone supplies no such argument. This section therefore adds a mathematical extension with explicit premises; it does not fit a cascade model to the synthetic release counts.

Maintain Article A's orientation: K_{ij} bounds the expected number of type- i children produced by a type- j parent. Population vectors are columns. Types may describe distinct agent roles or taint episodes, but their meaning must be fixed before choosing parameters. A root is an episode caused externally to the counted genealogy. A child has one declared causal parent. An adapted assignment rule resolves multiple plausible parents at birth; changing the rule after observing the eventual harm would invalidate the construction.

For a type- j episode, let $\overline{D}_j$ bound its conditional expected time to authoritative withdrawal, uniformly over admissible birth histories. Let d_j^M bound the remaining covered-influence lifetime at the enforcing receivers. A predictable conditional offspring intensity bounded by β_{ij} while influence remains enabled gives the familiar sufficient matrix

$$K_{ij} = \beta_{ij} \left(\overline{D}_j + d_j^M \right). \quad (19)$$

The argument uses conditional compensator integration, as in Article A. Average contact frequencies, mean detection times across a mixed dataset, and empirical service percentiles do

not by themselves satisfy the uniform conditional premises. Correlated incidents and unusually adverse histories must remain within the proposed bounds. An unprotected communication bypass lies outside the claimed covered channel.

9.2 Finite-horizon immigration theorem

Let $(\mathcal{F}_t)_{t \geq 0}$ be a common history filtration. Initial active episodes are represented as residual roots at time zero, with finite expected vector bounded by b_0 . Their remaining offspring and cost must satisfy the stated bounds conditional on $\mathcal{F}_0$; an unconditional bound at their earlier physical birth does not suffice after selection for survival. Additional roots arrive at stopping times with types measurable at birth. Define $N_i^{root}(T)$ as the number of new type- i roots born by deterministic horizon T . Suppose

$$\begin{aligned} \mathbb{E} [N_i^{root}(T)] &\leq \int_0^T \bar{\eta}_i(u) du < \infty, \\ b(T) &= b_0 + \int_0^T \bar{\eta}(u) du. \end{aligned} \tag{20}$$

One sufficient condition is a predictable root-intensity process dominated by the deterministic integrable function $\bar{\eta}_i$. Shared shocks may create correlated roots; independence is not required. When a model contains fixed-time bursts or other atoms, replace the integral by the mass of an integrable deterministic measure, including those atoms. An absolutely continuous intensity formula must not silently omit them.

Consider the complete descendant forests of the roots born by T , including descendants born after T . Selection uses the ancestor root's birth time, not the tree's realised size or harm. At each episode's admissible birth history, its existence, type, generation and selected-root label must be measurable. Assume a finite, nonnegative matrix K uniformly bounds its conditional expected lifetime offspring vector. Assume also that a type- j episode's attributable nonnegative remaining harm has conditional mean at most c_j . These are the same kind of birth-history obligations as Article A, now required throughout continuing root arrival.

Every harmful operation occurring by T must be assigned exactly once to a selected episode or to pre-existing pending work outside these forests. Let H_{pre} bound the latter contribution pathwise in the decomposition, and suppose $\mathbb{E}[H_{pre}] \leq h_{pre} < \infty$. Previously active episodes represented in b_0 and previously pending operations represented in H_{pre} must use disjoint attribution. Episode cost includes its queued effects after the episode itself stops; retiring an agent does not erase that cost.

Proposition 7 (continuing-root expected harm). Under these premises and $\rho(K) < 1$, the named nonnegative harm through horizon T obeys

$$\mathbb{E}[H_T] \leq c^\top (I - K)^{-1} b(T) + h_{pre} =: U_T. \tag{21}$$

Proof. Let $Z_T^{(n)}$ count generation- n episodes in the selected complete forests. Selection of a root born by T is known at that root's birth, and its label remains known at each descendant birth. For each potential parent, multiply the conditional offspring bound by the birth-measurable indicator that the episode exists in the selected forest with the specified type and generation. Take expectations and sum. Nonnegativity permits Tonelli's theorem, yielding

$$\begin{aligned}\mathbb{E} \left[Z_T^{(0)} \right] &\leq b(T), \\ \mathbb{E} \left[Z_T^{(n+1)} \right] &\leq K \mathbb{E} \left[Z_T^{(n)} \right].\end{aligned}\tag{22}$$

Induction gives $\mathbb{E} \left[Z_T^{(n)} \right] \leq K^n b(T)$. Apply the same conditional-expectation argument to attributable episode costs. The non-overlapping pathwise decomposition gives an upper bound by H_{pre} plus all costs in the selected forests. Monotone convergence permits the sum over generations. Because K is finite-dimensional, nonnegative and subcritical, its Neumann series converges to $(I-K)^{-1}$. Substitution proves the claim. Neither independent roots nor independent descendant trees are used. $\square$

Counting complete descendant forests intentionally overcounts effects after T . It supplies a convenient upper envelope, not a time-resolved prediction of harm. Roots born after T cannot cause an earlier counted effect under the causal construction and therefore need not be included. Conditioning retrospectively on an unusually large tree or selecting only roots that eventually become harmful would change the birth-history selection and is not covered by the proof.

Subcritical propagation does not bound harm uniformly over an endless external input. For constant $\bar{\eta}$, U_T grows linearly with T . A stationary local prevalence, if separately established, would not imply a bounded cumulative count of releases. The theorem concerns finite-horizon accumulated harm, not a thermodynamic phase transition or a claim of complete system safety.

Engineering interpretation. A file-release system can contain secondary spread while still receiving new compromised sources or external incidents. Repeated new incidents continue to add potential harm, even when each incident's propagation is controlled. The theorem therefore asks about harm over a chosen observation period, not whether lifetime loss stays fixed forever. Its calculation also counts the complete descendant chains of roots arriving within that period, including later descendants. This deliberate overcount makes the bound conservative rather than a prediction of the exact timeline.

9.3 Robust envelopes and limits of estimated parameters

For a family of admissible models indexed by θ , suppose deterministic componentwise bounds satisfy $K_\theta \leq \bar{K}$, $c_\theta \leq \bar{c}$, $b_\theta(T) \leq \bar{b}(T)$ and $h_{pre,\theta} \leq \bar{h}_{pre}$. Each model must itself satisfy the conditional genealogy and cost premises. The matrix bound is an envelope of those premises, not merely an envelope of fitted sample means.

Proposition 8 (componentwise uncertainty domination). If $\rho(\bar{K}) < 1$, every admissible model satisfies

$$\mathbb{E}_\theta [H_T] \leq \bar{c}^\top (I - \bar{K})^{-1} \bar{b}(T) + \bar{h}_{pre}.\tag{23}$$

Proof. Nonnegative matrix multiplication preserves componentwise order. Hence $K_\theta^n \leq \bar{K}^n$ for every $n \geq 0$. Apply Proposition 7, compare each nonnegative term in the Neumann series and sum. The spectral condition guarantees convergence of the dominating series. $\square$

A rectangular uncertainty set may combine extremes that cannot occur together and therefore be conservative. A coupled parameter set can give a tighter bound through a separately justified supremum. If $\rho(\bar{K}) \geq 1$, this certificate is unavailable. That does not prove that any actual finite network has unbounded loss or even that an admissible individual model is supercritical.

Statistical uncertainty requires a different statement. A confidence set derived from data has a coverage probability; it is not automatically a deterministic family containing the true conditional mechanism. A confidence interval for a mean service time also does not establish a maximum enforcement delay. No unconditional expected-harm guarantee is inferred here by inserting point estimates or adding a confidence level to the matrix formula. Such a claim would require a data-conditioned future-risk argument, including the consequences of failed coverage and model misspecification.

9.4 Exact two-type illustration

The numerical illustration deliberately uses declared parameters rather than the workload's observed harm. Let

$$\begin{aligned} \beta &= \begin{pmatrix} 0.02 & 0.03 \\ 0.04 & 0.01 \end{pmatrix} s^{-1}, & \bar{D} &= \begin{pmatrix} 2 \\ 4 \end{pmatrix} s, \\ c &= \begin{pmatrix} 1 \\ 3 \end{pmatrix}, & b(T) &= \begin{pmatrix} 1 + 0.01T \\ 0.02T \end{pmatrix}, & h_{pre} &= 0.5. \end{aligned} \tag{24}$$

Use a common supported delay d^M for both types. At 60 seconds the expected root envelope has components 1.6 and 1.2. Type 2 carries three harm units per episode; the units are chosen severity weights, not measured damages. The release-count endpoint of Section 6 is therefore not numerically interchangeable with this illustration.

The delays 2.7 and 7.2 seconds are worst-case bounds inside the eight-job, twenty-credential delivered-push fixture: the stipulated publication delay plus $(30 + 8 \times 6)/40$. Mirror reconciliation adds at most ten seconds, giving 12.7 seconds. In the selected initial partition state, protected expiry supplies 14.5 seconds. To use any of these numbers for communication, the corresponding receiver contract must cover that channel. Moreover, 14.5 seconds is the residual budget of this initial watermark; arbitrary episode phases need the appropriate age bound, potentially the full configured cap.

Table 11: Exact harm envelopes under supported delay and parameter assumptions.

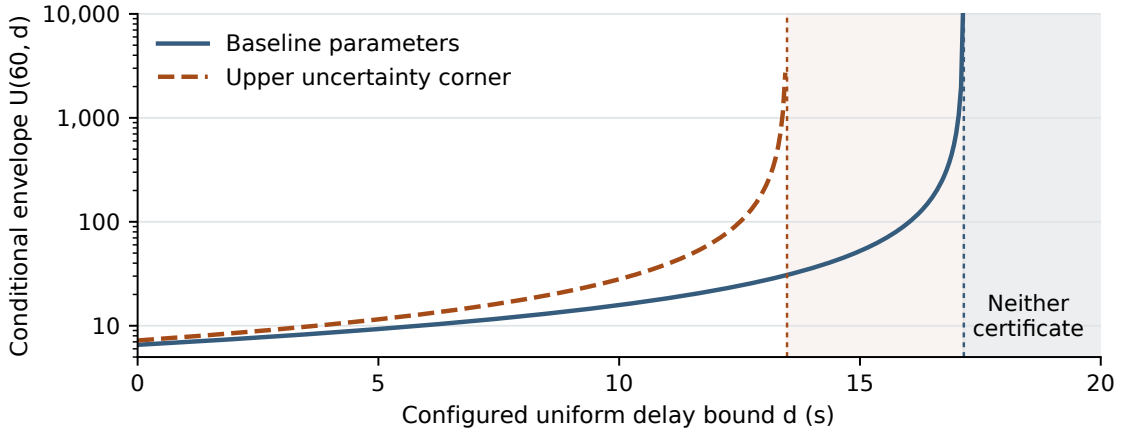
Supported delay (s)	$\rho(K)$	U_{60} (harm units)	Enlarged-parameter envelope
2.7	0.27536	7.80393	9.05168
7.2	0.50147	11.36262	15.55806
12.7	0.77696	25.36457	124.09250
14.5	0.86705	42.50278	Certificate unavailable

The enlarged envelope increases every offspring-intensity bound by 15%, detection-mean bounds by 20%, and the supported delay by 0.5 seconds. These changes define a sensitivity rectangle, not a statistical confidence region. At the last point its spectral radius is 1.05839. Exact rational inversion gives $U_{60} = 420117/53834$ for the first baseline row and $61119/1438$ for the last. No Monte Carlo estimation is involved in these matrix values.

Continuing roots and uncertainty in the harm envelope

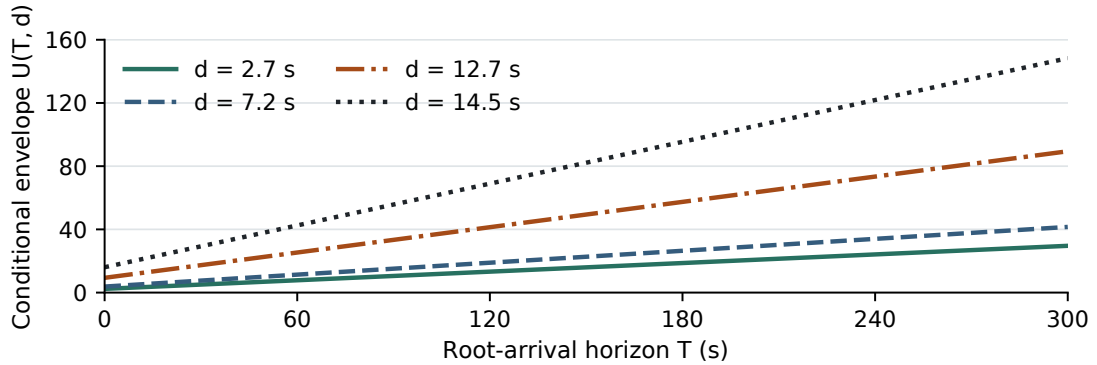
Illustrative parameters and hypothetical, uniformly supported influence-delay bounds

A Delay sensitivity at a 60-second root-arrival horizon



Only $\rho(K) < 1$ values are plotted; dotted lines mark the respective certificate thresholds.

B Continuing root arrivals increase the bound with the horizon



$U = h_{pre} + c^T(I - K)^{-1}b(T)$; the uncertainty rectangle is a parameter set, not a confidence interval.

Unavailable subcritical certificates do not imply infinite harm or growth in an arbitrary network.

Figure 6: Conditional harm envelopes with continuing roots. The delay d is a hypothetical uniformly supported influence bound, not one inferred from measured workload means or a particular initial watermark. (a) U_{60} on a logarithmic scale. The upper sensitivity corner raises offspring-intensity bounds by 15%, detection-mean bounds by 20%, and delay by 0.5 seconds. Dotted lines mark the respective spectral thresholds. This parameter rectangle is not a confidence region. (b) Baseline envelopes as the root-arrival horizon increases. The inverse counts complete descendant forests of selected roots, including descendants born after the horizon. These are upper envelopes rather than exact time-resolved expectations. An unavailable subcritical certificate implies neither infinite observed harm nor exponential growth in every finite network.

9.5 Pending tails and operational interpretation

Let $P(t)$ contain unresolved logical operations. For each operation, a valid conditional upper bound $q_o(t)$ must concern its future attributable harm under the receiver contract. It is not inferred from the fact that a sender timed out. Conditional linearity gives

$$\mathbb{E}[H_{pending} \mid \mathcal{F}_t] \leq \sum_{o \in P(t)} q_o(t). \quad (25)$$

If an operation can cause at most one counted harmful release, an upper bound on its conditional harmful-release probability bounds its conditional expected release count. Otherwise the operation requires a conditional expected count bound; severity requires a corresponding cost bound. If an operation is already charged to an episode cost in Proposition 7, it cannot be added again through h_{pre} . An unreceived terminal receipt can leave accounting unresolved even when the effect has physically completed. Conversely, a cancelled sender task can leave a future receiver effect possible. These are distinct states.

For a named nonnegative harm and a valid U_T , Markov's inequality gives $\Pr(H_T \geq z) \leq \min\{1, U_T/z\}$ for $z > 0$. At $U_{60} = 7.80393$, the bound for at least 100 harm units is approximately 0.07804. The same formula is uninformative for sufficiently small thresholds. A low mean does not exclude a rare severe event, and correlated root shocks can change the tail while preserving the root mean. Stronger tail estimates require additional distributional premises and their own validation.

Reading the risk statement. The first numerical bound, about 7.8 harm units, is an upper bound on an expectation under the stated model. It is not a rule that every incident causes at most 7.8 units, and the units are not automatically a count of released files. A rare large event can coexist with that expectation. Engineers must specify the severity measure and the unacceptable-event threshold, then assess whether the available probability bound is informative enough for the decision being made.

The resulting research obligation is concrete. First justify the receiver-enforced influence lifetime and the conditional detection, offspring and cost envelopes. Then assess whether the robust matrix is subcritical and whether its finite-horizon harm and tail bounds meet a stated criterion. The synthetic workload can check scheduling logic within its configured envelope. It does not supply the external rate or severity premises needed to turn the conditional theorem into a deployment claim.

10 Integrated evaluation and statistical interpretation

10.1 What connects the three studies

The protocol, workload and privacy experiments answer different questions. The transition model asks whether specified guards preserve invariants over its reachable states. The workload experiment asks how stipulated publication and service rules change modelled enforcement and completion. The decoy experiment asks what an observer can infer under a declared distribution of complete histories. Agreement among their results is useful only where their interfaces agree. A fast publication trace cannot repair a missing execution guard, and a correct receiver cannot establish a false issuer assertion.

Each implementation claim therefore needs a traceable sequence of evidence. A credential representation determines concrete bytes. Parsing, authentication and dependency processing

consume work on a stated machine. Scheduling converts that work into installation and enforcement delays under a load envelope. The receiver's effect record identifies what was actually prevented. A conditional cascade calculation uses only parameters whose meaning matches the model. The present experiments instantiate parts of this sequence with declared constants; they do not measure every stage of a deployed system.

Table 12: Completed research evidence and remaining deployment obligations.

Research question	Result supplied here	Evidence still required for deployment
Causal decoys	A fixed-cohort generator, conditional privacy results and synthetic mismatch tests.	Incident-derived hazard laws, hidden-allocation controls and a full publication implementation.
Value of masking	A paired comparison on identical semantic histories and the known-initialisation limit.	A credible restricted observer for any claimed additional benefit.
Protocol invariants	Conditional proofs, bounded reachable-state checks and omission witnesses.	Refinement of the actual resource adapter, persistence layer and message handlers.
Multiplicity and containment	Equivalent-policy work parameters and a matched service model.	Byte-to-work calibration, verified service contracts and measured work costs.
Continuing incidents and uncertainty	A finite-horizon immigration bound and parameter sensitivity.	Supported conditional rate and cost bounds for the intended environment.
Ledger versus mirrors	Local-state alternatives under identical freshness and effect semantics.	End-to-end publication, recovery, synchronization and total resource measurements.
Truth and completeness	Explicit authority and discovery premises with their failure interpretations.	Governance and independent evidence capable of supporting those premises.

10.2 Units of replication and uncertainty

The independent unit in the workload fixture is an incident run generated from one seed. Requests within that run share the incident, service state and failure condition. They are not independent experimental replicates. Backend contrasts use the same incident inputs. Report the difference within each pair before aggregating across runs. This preserves the controlled comparison and avoids treating request volume as additional statistical evidence.

The privacy fixture similarly separates cohort-level replication from positions and snapshots inside a cohort. A fixed count of genuine positions makes labels dependent. Repeated snapshots also share absorbing state histories. The exact posterior calculation in Section 8 accounts for the declared fixed-count prior. For an empirical attack trained on future operational data, entire incidents or epochs must be held out. Randomly splitting neighbouring snapshots would expose related histories to both training and test sets.

Uncertainty has three sources. Sampling uncertainty concerns the finite set of runs from the stipulated generator. Parameter uncertainty concerns the values used to describe a real deployment. Model error concerns omitted mechanisms or incorrect structural assumptions. Increasing the number of model runs reduces only the first. A narrow simulation interval around

an assumed finality delay says nothing about whether a public ledger will meet that delay during a real incident.

For a later study with n independent and identically distributed (i.i.d.) incident trials, suppose no trial has a named harmful release. The exact one-sided binomial upper bound at confidence $1 - \alpha$ is obtained by solving the probability of zero events:

$$p_{upper} = 1 - \alpha^{1/n}. \quad (26)$$

This bounds the probability of the binary incident outcome under the sampled distribution. It does not bound the number of releases in a harmful incident, their severity or risk under arbitrary deployment conditions. To obtain a zero-event upper bound no larger than ε , the required sample count satisfies:

$$n \geq \left\lceil \frac{\log \alpha}{\log(1 - \varepsilon)} \right\rceil. \quad (27)$$

At 95% confidence, the required counts are 299 for a 1% target and 2995 for a 0.1% target. These are exact design calculations conditional on observing zero events. They are not a power calculation or a promise that the study will observe none. Different fault conditions require separate estimates or a declared sampling mixture; pooling convenient low-risk runs with difficult incidents changes the quantity being estimated.

10.3 Progression to a deployment experiment

A deployment study should first reproduce the transition witnesses against the actual receiver and storage implementation. It should then replace one stipulated quantity at a time with a measured, versioned quantity. Signature work, decompression, dependency traversal and queue scheduling need separate measurements because their cost can respond differently to credential multiplicity. The benchmark must preserve equivalent policy decisions while changing credential decomposition.

The hosting experiment should include cold startup, sustained publication, missed events, origin failure, peer isolation and recovery after a checkpoint gap. It should count synchronization and consensus traffic on declared links rather than reporting only the compact application update. The privacy experiment should expose the same headers, writers, event fields and batch timing that the deployed publisher exposes. Failed runs, censored operations and missed deadlines remain results. An implementation that returns to service by discarding unresolved operations has changed the security contract rather than improved recovery.

11 Practical implementation guidance

11.1 Protected gates and persistent operations

Pattern P1 places the decision and effect guard at the receiver's last preventable boundary. An approval contains the complete-record binding, audience, resource, dependency versions and expiry. The adapter either executes within the guarded transition or supplies an equivalent mechanism that preserves its ordering. Where an external service performs the irreversible action later, the receiver needs its own expiry and cancellation contract. The sender's database transaction cannot supply that guarantee on the receiver's behalf.

Pattern P6 retains the corresponding operation identity through retries, agent retirement and recovery. The sender stores the admitted record and debit before dispatch. The receiver stores an idempotency tombstone and terminal outcome under its execution contract. Closure accepts evidence for that exact logical operation and distinguishes execution, cancellation and another explicitly agreed terminal result. Neither an elapsed request timeout nor an agent's disappearance is terminal evidence. Storage retention and identity reuse rules must outlast every message that could still cause a replayed effect.

These controls also determine the meaning of recovery. A restarted service restores accepted epochs, operation records and supported clock state before admitting new ordinary work. If persistence integrity or clock bounds are uncertain, recovery proceeds under a separately authorised restricted mode. A controller may issue cancellation requests while it refuses new releases. Such behaviour preserves the purpose of fail-closed containment without assuming that every maintenance action is prohibited.

11.2 Caches and dependency processing

Pattern P2 shares origin-authenticated state inside a relying domain. The cache validates authority, profile, ordering and freshness before exposing a version to decisions. Push signals accelerate invalidation; reconciliation repairs missed delivery. A signal with insufficient authority or content can trigger retrieval without changing the semantic state directly. A cache hit records which accepted version was used. Counting hits without their watermark ages can conceal stale reliance.

Pattern P3 maintains a typed reverse index from evidence and policy dependencies to affected decisions. An update invalidates cached conclusions about that dependency. Re-evaluation can preserve a record when an admissible alternative still proves the same complete continuation. The index must include prohibitions and relevant adverse evidence. Where discovery or index maintenance is uncertain, a conservative restriction takes precedence over an unsupported claim that unaffected records have been found exactly.

A shared cache is also a privacy boundary. Tenant identity, subscription patterns and internal invalidation logs can reveal activity even when the external status service receives no per-decision query. Separate operational logging from unnecessary credential-level telemetry. Retain the evidence needed for incident reconstruction under a declared access and retention policy. A private local interface does not justify exporting detailed query traces to an unrelated observability service.

11.3 Service admission and operational thresholds

Pattern P4 translates a deadline into an admission condition. Assume a first-in, first-out reserved service class, or an equivalent non-overtaking guarantee. The service contract supplies the reserved rate after the bounded latency over each continuously backlogged interval, including permitted blocking and interference. Let q be work already ahead of a control job, w_o its supported work bound, $R_{srv} > 0$ the reserved service rate and θ the maximum scheduling latency. If b_o is the remaining budget for this service stage, the permitted queue headroom is:

$$q \leq R_{srv} (b_o - \theta) - w_o. \quad (28)$$

The right-hand side must be nonnegative, and its work units must match the measured or proved service contract. The rule requires a bound on interference, non-preemptible blocking and admitted job size. A priority label or low average processor utilisation cannot establish it.

If the detailed path cannot meet the inequality, a separately supported broad suspension or protected expiry path must prevent the covered effect. Recovery work can continue under its own bound while ordinary admission remains restricted.

For status object l at verifier v , use Section 3's conservative age $A_{v,l}(t) = \hat{t}_v(t) - \omega_{v,l} + \kappa_{v,l}(t)$, where $\hat{t}_v(t)$ is the protected local clock estimate and the margin covers source and verifier clock uncertainty. The estimate must dominate the true elapsed age of the authoritative completeness cut throughout the claimed interval. Define residual headroom $h_{v,l}(t) = \tau_l - A_{v,l}(t)$. Ordinary admission requires positive headroom for every mandatory object and compliance with all other validity and lease limits. Refresh alerts should start before headroom falls below the supported refresh-and-application budget. Loss of a finite refresh bound does not permit the age cap to increase.

Operational telemetry therefore joins source age with completed enforcement. Record the authoritative decision, accepted publication, validated installation, generation change, effect decision and terminal receipt as distinct events. Report clock intervals when comparing domains. Queue work, refresh failures, unknown-status decisions and the age of unresolved operations are direct control indicators. Aggregate requests per second and cache-hit ratios describe load but cannot substitute for those indicators.

11.4 Privacy lifecycle and algorithm changes

Pattern P5 treats a cover cohort as a finite resource with an explicit lifecycle. Protect the master allocation; deliver only the required per-credential decoding information; use a cryptographic random source in any deployed mask. Monitor remaining decoy reserve against the declared hazard model. Rotation, replenishment, changes in writer identity and emergency publication all enter the observer's transcript. A generator that matches routine independent revocations may fail during a common issuer compromise, precisely when many real entries change together.

Pattern P7 binds accepted algorithms and key versions to the relying policy. A migration inventory includes credential, event, transport, transaction, checkpoint and consensus authentication. Hybrid rules state which signatures must verify and how a downgrade is rejected. Key withdrawal is exercised during publication and recovery, when stale trust in an old key could otherwise revive rejected state. Post-quantum object signatures change specific authenticity assumptions; they do not establish the truth of signed claims or the quantum resistance of an unchanged ledger stack.

11.5 From the abstract protocol to implementation tests

Begin with a receiver test harness for one specified file-release operation. Reproduce the abstract checks in Section 4, then connect the same obligations to actual storage and the resource adapter. Use full evidence reevaluation as the initial decision baseline; P3's dependency index is an optimisation to test against it. Start with the pinned ordinary status profile and signed snapshots. Changing hosting or adding experimental privacy should follow a working enforcement boundary, because otherwise a favourable backend result can conceal a defective gate. The following sequence describes required deployment work, not validation completed in this paper.

First, define the implementation-to-model mapping. For every protected transition in Section 3.2, name the responsible component, accepted input, persistent state and last preventable effect. Record the operation ID, complete-record binding, receiver, evidence generations, policy version and absolute expiry. Identify the actual storage transaction or other mechanism that supplies each claimed atomic step. A log message saying “committed” is insufficient unless its

relation to the irreversible effect is established. If effect execution and durable outcome cannot share the model's boundary, extend the state machine with the uncertain outcome and analyse recovery explicitly. Do not assert Proposition 3 for an adapter that lacks its premise.

Second, control the schedules that ordinary load tests miss. Add test-only pause points after evaluation, before final comparison, after dispatch, at the last preventable effect, and before terminal-receipt processing. Exercise Section 3.3's two local orders while keeping the complete operation fixed. An installed relevant withdrawal before the receiver's final transition must exclude its old witness. If the effect commits first, record it as an earlier committed effect. If the source has withdrawn but installation is delayed, an effect may remain permitted until the accepted-age cap; this is residual exposure, not a violation of Proposition 1. A fresh alternative witness must be identified separately from reuse of the invalidated approval.

Third, expand beyond the finite checker's failure domain. Run sender and receiver as independently stoppable processes with separately delayed status views. Interrupt each persistent write and message boundary. Redeliver requests after receiver restart and failover; delay cancellation until after execution; deliver an old execution request after a cancellation tombstone. Suppress terminal receipts while retaining evidence of receiver execution. The sender must preserve unresolved accounting. The receiver must prevent duplicate effects within the declared identity and retention contract. These tests address behaviours absent from the single shared up/down flag and installed generation of Section 4.

Fourth, qualify timing and optimisation separately. Partition all refresh paths, retain the old watermark and attempt repeated lease renewal and restart. Exercise the declared clock-error envelope, then deliberately invalidate that envelope and verify the restricted response. Add competing work and withdrawal bursts while recording service headroom and completed receiver enforcement. If the detailed path cannot meet its budget, test the separately supported broad hold or expiry path. A broad hold requires its own bounded implementation; one atomic assignment in the simulation is not that evidence. Only then compare incremental reevaluation, publication adapters and privacy variants under the same policy and incident schedules.

The table gives concrete acceptance observations and their scope. A “pass” refers to the exercised fixture and its independent observations. It is not a proof for arbitrary schedules or deployments.

Table 13: Implementation tests and the limits of passing observations.

Requirement	Test or injected fault	Passing observation	Remaining limit
Exact record and current receiver state	Change recipient or payload after evaluation; install a relevant withdrawal at each pause point.	Changed binding is rejected; earlier local installation excludes the old witness at the covered effect.	Does not prove source truth, dependency completeness or instantaneous inter-domain withdrawal.
Parent freshness and durable expiry	Remove all update paths; renew repeatedly; restart before and after the cap.	No covered ordinary effect occurs at or after a required strict cap under the supported clock convention.	Availability can end; clocks and authoritative watermark semantics remain premises.

Continued on next page

Table 13 (continued)

Requirement	Test or injected fault	Passing observation	Remaining limit
Persistent identity and terminal closure	Crash sender and receiver separately; duplicate dispatch; suppress receipts; replay after cancellation.	No duplicate observed effect or debit per bound ID; unresolved sender state persists until valid terminal evidence.	General correctness still needs justified atomicity, failover fencing and retention bounds.
Wire-profile interpretation	Use known valid objects, altered signed bytes, wrong purpose or epoch, and bit indices 0, 7, 8, 255 and 256.	One declared decoding and authority interpretation; invalid objects fail within bounded input-processing limits.	Model counts do not establish cryptographic or standards conformance.
Dependency optimisation	Compare indexed reevaluation with full reevaluation after each policy or evidence mutation.	Both reach the same complete-record decision on the declared coherent inputs.	Agreement cannot detect a premise omitted from both implementations.
Supported enforcement service	Saturate ordinary work, inject a withdrawal burst and delay selected publication paths.	The named effect boundary meets its supported path budget or the declared restrictive fallback applies.	Sampled tail latency does not prove an unrestricted deadline; legitimate loss is reported separately.

Retain the input objects, software and configuration versions, fault schedule and operation-level traces for every failed and passed fixture. Observe released bytes or the corresponding effect through a fixture independent of the decision log, so missing instrumentation cannot silently count an execution as blocked. Report tested schedules, unexplored cases and clock uncertainty. Separate profile conformance, invariant violations, observed delay, unresolved outcomes and legitimate service loss. Negative tests should also show that authentic but false source assertions remain outside the protocol's truth guarantee. Where the required evidence cannot be established, restrict the action's scope or keep the deployment claim at the tested level.

12 Research extensions and external validity

The next protocol step is refinement of a real effect adapter against Section 3's transition system. The mapping must identify where an accepted status change takes effect, where an operation becomes irreversible and which durable records survive each crash boundary. Testing message handlers against an abstract model can reveal deviations. A complete implementation argument additionally covers parsing, cryptography, persistence, scheduling and the resource mechanism. The bounded checker supplies a reproducible starting point for this work.

The main privacy extension is continuous issuance with correlated incidents. A deployed issuer creates and retires credentials while its incident process changes. Decoy generation must respond causally without publishing an identifying signal or using future revocations to construct a retrospective match. The study needs an explicit model for replenishment, cohort age and knowledge of already issued positions. An active attacker that requests selected credentials or induces incidents requires a separate intervention experiment. Extending a passive guarantee without modelling that interaction would change its meaning.

Accumulator-backed non-revocation already has implementations in AnonCreds; it is not solely a research proposal [19, 22]. The remaining work here is its integration and matched evaluation under Section 3’s enforcement contract. Section 5.10 and Appendix F define that work. Article A discusses current adoption, coexistence with bitstrings and additional privacy alternatives.

State channels and layer 2 publication require a public fallback for unilateral withdrawal. A new external verifier must establish an authoritative accepted state without assuming that it participated in every prior update. Dispute periods, censorship, watcher failure and data availability enter the deadline and unresolved-operation model. A prompt sequencer response cannot be substituted for the settlement condition selected by policy. These questions belong in a dedicated implementation extension; they are not silently resolved by the small ledger-path model here.

Large language model (LLM) agents can later supply richer proposal and communication workloads. Their inclusion should preserve a deterministic resource oracle and the same protected boundaries. Record the model, prompt, tool and budget configuration for each run. Greater agent diversity may change contact rates and incident patterns; it does not make those rates uniform conditional bounds automatically. Physical actuation requires a further account of stopping distance, stored energy and irreversible effects, which the engineering-file case does not provide.

13 Scope and conditions for article claims

The formal results apply to the stated transition and observation models. They establish local ordering, bounded stale reliance, persistent accounting and conditional inference or expected-harm statements under explicit premises. The exhaustive checks cover only their finite configurations. The workload and privacy studies are executed model experiments with disclosed generators and constants. No latency, gas cost, throughput or privacy result in this article is a measurement of a production credential service or public ledger.

The principal operational limits are truthful authoritative state, complete evidence under the discovery contract, protected gate integrity, supported clock bounds and receiver-side effect control. A source that signs false fresh assertions defeats the completeness premise. A bypassed adapter defeats complete mediation. Uncontrolled outgoing messages defeat the corresponding influence-delay assumption. A system may detect some of these failures through additional mechanisms, but that detection has its own evidence and timing requirements.

The privacy limits are equally specific. Known indices permit status tracking for those credentials. Known active initialization exposes a fixed mask. Matching marginal transition rates does not match the joint public transcript. The evaluated common-shock and hazard-mismatch cases show why routine cover generation cannot be described as a general anonymity guarantee. Its usefulness must be assessed jointly with service demand and the operational cost of maintaining the assumed observation law.

The manuscript supports a protocol and evaluation method for a bounded class of distributed agent workflows. It does not certify general AI safety, solve the semantic truth of credentials or establish universal superiority of either publication backend. Deployment claims require the actual implementation and environment to supply the premises that are configured, abstracted or independently assumed here.

Research transparency

The computational studies use synthetic, non-personal inputs. The reproducibility package contains executable sources, configuration files, result tables and figure sources. The author is the founder of Spherity GmbH, which develops identity and trust infrastructure. Independent implementation and review would strengthen the assessment of deployment relevance. The first article and this manuscript form a separate two-paper study; neither is a replacement for the previously published Spherity AI Safety Series.

References

- [1] Carsten Stöcker. Cascading Trust Failures in AI Agent Networks: A Trust Architecture for Scalable Revocation and Bounded Containment. 2026-10-10. Supplied revised manuscript.
- [2] W3C Verifiable Credentials Working Group. Recognized Entities v1.0. 2026-09-06. W3C Working Draft. <https://www.w3.org/TR/2026/WD-vc-recognized-entities-1.0-20260906/>
- [3] W3C Verifiable Credentials Working Group. Recognized Entities Threat Model v1.0. 2026-10-08. W3C Group Note Draft. <https://www.w3.org/TR/2026/DNOTE-vc-recognized-entities-threat-model-1.0-20261008/>
- [4] Scott Rose, Oliver Borchert, Stu Mitchell, Sean Connelly. Zero Trust Architecture. 2020-08. NIST Special Publication 800-207. <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf>
- [5] W3C Verifiable Credentials Working Group. Bitstring Status List v1.0. 2025-05-15. W3C Recommendation. <https://www.w3.org/TR/2025/REC-vc-bitstring-status-list-20250515/>
- [6] W3C Verifiable Credentials Working Group. Bitstring Status List v1.1. 2026-09-24. W3C First Public Working Draft. <https://www.w3.org/TR/2026/WD-vc-bitstring-status-list-1.1-20260924/>
- [7] A. Tulshibagwale, T. Cappalli, M. Scurtescu, A. Backman, J. Bradley, S. Miel. OpenID Shared Signals Framework Specification 1.0. 2025-08-29. https://openid.net/specs/openid-sharedsignals-framework-1_0-final.html
- [8] T. Cappalli, A. Tulshibagwale. OpenID Continuous Access Evaluation Profile 1.0. 2025-08-29. https://openid.net/specs/openid-caep-1_0-final.html
- [9] Felix Hoops, Jonas Gebele, Florian Matthes. CRSet: Private Non-Interactive Verifiable Credential Revocation. 2025-09-17. arXiv:2501.17089v3. <https://arxiv.org/html/2501.17089v3>
- [10] Patrick Herbke, Wolf Rieder, Christian René Sechting, Huaning Yang, Sid Lamichhane, Philip Raschke, Axel Küpper. ShadowPath: Lookup-Private Credential Status Verification over Authenticated State. 2026-08-20. arXiv:2608.19937v1. <https://arxiv.org/html/2608.19937v1>
- [11] Maurice P. Herlihy, Jeannette M. Wing. Linearizability: A Correctness Condition for Concurrent Objects. 1990-07. ACM Transactions on Programming Languages and Systems 12(3), 463–492. <https://cs.brown.edu/~mph/HerlihyW90/p463-herlihy.pdf>
- [12] Martín Abadi, Leslie Lamport. The Existence of Refinement Mappings. 1991-05. Theoretical Computer Science 82(2), 253–284. <https://lamport.azurewebsites.net/pubs/pubs.html#refinement>

- [13] L. Peter Deutsch. GZIP file format specification version 4.3. 1996-05. RFC 1952. <https://www.rfc-editor.org/rfc/rfc1952.html>
- [14] Simon Josefsson. The Base16, Base32, and Base64 Data Encodings. 2006-10. RFC 4648. <https://www.rfc-editor.org/rfc/rfc4648.html>
- [15] Philipp Bolte, Dennis von der Bey, Lauritz Leifermann. ERC-7506: Trusted Hint Registry. 2023-08-31. Draft, created 31 August 2023; inspected 9 October 2026. <https://eips.ethereum.org/EIPS/eip-7506>
- [16] A. Backman, M. Scurtescu, P. Hunt. Push-Based Security Event Token (SET) Delivery Using HTTP. 2020-11. RFC 8935. <https://www.rfc-editor.org/rfc/rfc8935.html>
- [17] National Institute of Standards and Technology. Module-Lattice-Based Digital Signature Standard. 2024-08-13. FIPS 204. <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.204.pdf>
- [18] National Institute of Standards and Technology. FIPS 204 publication page and planning note. 2026-07-31. <https://csrc.nist.gov/pubs/fips/204/final>
- [19] AnonCreds Working Group. AnonCreds Specification, version 1.0, draft. Accessed 10 October 2026. <https://anoncreds.github.io/anoncreds-spec/>
- [20] Hyperledger Indy. Indy SDK for Node.js: revocation registry and witness interfaces. Accessed 10 October 2026. <https://hyperledger-indy.readthedocs.io/projects/sdk/en/latest/wrappers/nodejs/README.html>
- [21] A. Fraser and S. Schneider. A Formal Security Analysis of Hyperledger AnonCreds. IEEE European Symposium on Security and Privacy, 2025. IACR ePrint 2025/694. <https://eprint.iacr.org/2025/694>
- [22] OpenWallet Foundation. AnonCreds Credential Revocation. ACA-Py documentation. Accessed 10 October 2026. <https://aca-py.org/latest/gettingStarted/CredentialRevocation/>
- [23] Google. Colaboratory Frequently Asked Questions. Accessed 2026-10-09. <https://research.google.com/colaboratory/faq.html>
- [24] M. Rosenberg, J. White, C. Garman, and I. Miers. zk-creds: Flexible Anonymous Credentials from zkSNARKs and Existing Identity Infrastructure. IEEE Symposium on Security and Privacy, 2023. <https://obj.umiacs.umd.edu/ieeesp23/zk-creds.pdf>

A Reproduction contract

A.1 Computational artifacts and fixed inputs

The source package separates the finite transition checker, workload simulation, decoy inference and exact matrix calculations. Reproduction starts from the recorded configurations, rather than from values transcribed from a plotted curve. Each result file identifies its generating program. Seeds control simulation inputs; the analytical matrix calculations and reachable-state counts do not depend on random sampling. The package records software versions and source digests so a changed program can be distinguished from another execution of the same program.

The protocol checker uses finite immutable states and a breadth-first visited-state set. The one- and two-identifier baselines terminate at the reachable-state fixed point. Omission searches stop at the first violation. Reproduction must retain that difference. Changing the ordering of successor enumeration may select a different witness among equally short failures without changing the invariant conclusion. The published witness is checked against its corresponding transition definitions, not treated as a generic network incident.

The workload source generates one incident from each seed and reuses its inputs across configurations. It records the selected publication path, credential decomposition, batch size and fault. Per-incident records preserve affected job rank, work ahead, installation time, enforcement boundary, prohibited effects, legitimate outcomes and horizon censoring. The synthetic byte constants are identified separately. Removing excluded overheads from the boundary description would turn an accounting illustration into an unsupported network comparison.

The decoy source records cohort size, genuine count, horizon, hazard law and observation model. The exact fixed-count posterior uses all observed transition categories. Mixture cases retain the shared latent event rather than replacing it with independent marginal mixtures. Posterior ties use the declared numerical rule. Masked and unmasked representations reuse identical semantic histories. Compression fixtures record raw, compressed and encoded byte counts by snapshot, including initialization, with the same fixed codec settings for both representations.

A.2 Regeneration and interpretation

Run the component programs using the supplied reproduction instructions. The outputs include the state counts and omission traces, incident tables and summaries, decoy inference and reserve results, and harm curves with exact comparison points. Figure scripts consume those result files. They do not contain independently invented observations. Regenerating a figure must therefore follow regeneration of its data when any model parameter changes.

Expected checks include the baseline state counts, detection of each omitted-control witness, enforcement before the configured cap in the workload model, exact agreement of masked and unmasked inference under known initialization, and consistency of matrix values with rational comparison points. A check can fail because of a software defect or because the model was intentionally changed. Record that distinction before comparing outputs. Wall-clock program runtime is not reported as a credential-verification performance measurement.

The package is a research implementation of the stated abstractions. It is not a production credential issuer, a conforming status-list server or an audited ledger contract. No cryptographic keys, real design files or personal credential records are required. The standards profiles describe obligations for a subsequent implementation; passing the model checks does not establish those obligations automatically.

B Multi-agent implementation checklist

Table 14: Multi-agent implementation checklist.

Component or phase	Control and security requirement	Verification method	Target or bound
Evidence and recognition	Bind one complete record to scoped source authority and a declared discovery contract.	Conflicting continuations, missing adverse evidence and false-authority cases.	Every admitted effect has the required joint witness.
Status decoding	Pin purpose, width, bit order, allocation, epoch and supported profile.	Byte and word boundaries, malformed inputs and cross-epoch replays.	One accepted semantic interpretation; bounded rejection work.
Decoy configuration	Protect allocation and match the declared full transcript, including writers and shocks.	Known indices, public initialization, genuine-only events and reserve exhaustion.	Only the stated posterior or inference claim.
Signature profile	Bind signed bytes, context, suite and key version; reject downgrade.	Payload mutation, wrong context, key withdrawal and mixed-suite cases.	Accepted algorithms and measured wrapper costs are explicit.
Gate mediation	Serialize final dependency checks with the last preventable effect.	Invalidation before, during and after the commit boundary.	No stale witness survives an earlier relevant local installation.
Admission and persistence	Persist identity, debit and unresolved record before dispatch.	Crash at each storage and dispatch boundary; repeated logical requests.	At most one debit per bound identifier.
Receiver idempotency	Persist one effect or cancellation result and retain its tombstone.	Duplicate and delayed dispatch across restart and failover.	At most one modelled effect per identifier under the receiver contract.
Freshness and renewal	Cap each child lease by every parent age and validity limit.	Partition, old re-signing, stale renewal and clock-bound loss.	Covered effects stop when a required strict cap is reached.
Event and cache handling	Distinguish receipt, validated installation and completed enforcement.	Lost, reordered and repeated signals; reconciliation after a gap.	Freshness is never inferred from transport success alone.
Reserved control service	Bound admitted work or use a separately supported restrictive path.	Bursts, expensive malformed inputs and competing resource load.	$\theta + (q + w_o) / R_{srv} \leq b_o$ for the claimed path.

Continued on next page

Table 14 (continued)

Component or phase	Control and security requirement	Verification method	Target or bound
Remote closure	Remove pending state only after contract-valid terminal evidence.	Lost, conflicting and premature receipts; delayed downstream execution.	Timeout retains unresolved accounting.
Recovery and audit	Restore monotonic state and preserve unresolved operations.	Rollback, checkpoint loss, key replacement and cold startup.	No ordinary admission before required trust and time premises hold.

Section 11.5 gives a staged test sequence and distinguishes fixture-level acceptance from the unproved implementation premises.

C Evidence required beyond the model study

A field study should freeze its primary comparison before collecting confirmatory observations. The freeze includes the complete policy predicate, incident distribution, hardware and software revisions, chain acceptance rule, cryptographic profile, load envelope and receiver contract. A pilot can establish feasible precision and identify implementation faults. Pilot-driven changes must be separated from the subsequent confirmatory analysis rather than hidden inside repeated model tuning.

The deployment comparison requires actual signed objects and supported status semantics. It should preserve both the original authoritative decision time and the time each backend first makes that decision usable. Record initial synchronization, state validation, proof checks, event delivery, cache installation and final receiver action. A total byte account identifies each measured link and includes request, signature, retransmission and background consensus traffic where relevant. Comparing application payload alone cannot answer a total-cost question.

The privacy study requires a publisher whose observable metadata is captured faithfully. It should include known credential indices, correlated source incidents, continuing issuance, emergency publication and genuine operational rotation. Candidate attacks should be fixed before evaluating held-out incidents. An attack that performs near its baseline does not prove that every attacker fails. Conversely, a single reliable real-only signal can refute the proposed full-transcript guarantee even when bit-flip counts look well matched.

The harm study requires a documented relation between an observed incident and a counted episode. Root and parent assignments cannot be chosen after seeing which assignment gives the most favourable bound. Uniform conditional offspring and cost envelopes need support across the intended histories. Where only empirical averages are available, report finite-horizon outcomes and uncertainty without invoking a theorem whose premises are not established. This separates a useful empirical result from a stronger certificate the evidence does not support.

D Runnable model code and selected outcomes

This appendix prints the complete bootstrap, execution, result-checking and plotting code for all four model modules. It calls the original model implementations supplied in `Runtime_Revocation_Reproducibility_Package.zip`. Those implementations are not replaced by shorter

teaching models. The four source files retain their original bytes and default parameters. The printed code checks their hashes with the 256-bit Secure Hash Algorithm (SHA-256) before execution. The large encoded source archive and the model implementation bodies are supplied as executable files rather than printed as encoded text.

Two execution routes are available. The original `Paper_2_Runtime_Revocation_Colab.ipynb` remains self-contained because it embeds its model sources and recorded results. The matching `Paper_2_Printed_Appendix.ipynb` instead uses the readable bootstrap printed below and requires the supplied reproducibility ZIP to be uploaded. It produces the same model results and the three appendix figures. Neither route downloads model code or external datasets.

D.1 Environment and complete source-package bootstrap

Open <https://colab.research.google.com/>, create a Python notebook or upload `Paper_2_Printed_Appendix.ipynb`, and upload `Runtime_Revocation_Reproducibility_Package.zip` through the session Files area [23]. Run the following cells in order. Use a normal Python runtime; no graphics processing unit is needed. NumPy, SciPy and Matplotlib must be available. The code performs no automatic installation, requires no credentials and does not mount a drive. If the uploaded ZIP has another name or location, change `ZIP_PATH`. Extraction uses a fixed list of source and result paths and creates a fresh temporary directory.

The printed cells were executed in order in a fresh local Python process using Python 3.12.14, NumPy 2.3.5, SciPy 1.17.0 and Matplotlib 3.10.8. They have not been tested in a Google-hosted runtime. Runtime versions are printed because codec and floating-point details can vary. The checks compare regenerated outputs with the recorded results at their stated tolerance. Elapsed program time is a local reproduction measurement, not a deployed revocation-service benchmark.

```
from pathlib import Path
import csv
import hashlib
import json
import math
import platform
import shutil
import subprocess
import sys
import tempfile
import time
import zipfile
import numpy as np
import scipy
import matplotlib
import matplotlib.pyplot as plt

# Upload this supplied ZIP to Colab's session Files area first.
# Change only this path if the uploaded file has a different name.
ZIP_PATH = Path('Runtime_Revocation_Reproducibility_Package.zip')
PREFIX = 'Runtime_Revocation_Source/'
assert ZIP_PATH.is_file(), 'Upload the supplied reproducibility ZIP first.'

SOURCE_HASHES = {
    'research/protocol_check.py':
        '09c6db03f3760b23ae75ea15ad549355af5fe9911efa6081933a4a9f10826be4',
    'code/run_workload.py':
        'c418b009d72e2c27817f942a4011453fe7f64bdfe3c6e59f01248af7c3109e5f',
    'experiments/decoys/decoy_experiment.py':
```

```

        'c153d0e1813dd6a99bcb72428f9b2e087691e28d4d7db9534c1a30b2e7561869',
        'code/harm_envelope.py':
            'able498393c0d2c8bf9efeb7ed5a0f00ef8eee411f381f9390c3b040011363b0',
    }
    FROZEN_RESULTS = (
        'research/results/protocol_results.json',
        'data/workload_config.json',
        'data/workload_summary.csv',
        'data/workload_paired_contrasts.csv',
        'experiments/decoys/results.json',
        'experiments/decoys/privacy_summary.csv',
        'experiments/decoys/shock_conditional_summary.csv',
        'data/harm_exact_points.json',
    )
    MODEL_ROOT = Path(tempfile.mkdtemp(prefix='paper2_printed_appendix_'))
    with zipfile.ZipFile(ZIP_PATH) as archive:
        for relative, expected in SOURCE_HASHES.items():
            payload = archive.read(PREFIX + relative)
            assert hashlib.sha256(payload).hexdigest() == expected, relative
            target = MODEL_ROOT / relative
            target.parent.mkdir(parents=True, exist_ok=True)
            target.write_bytes(payload)
        for relative in FROZEN_RESULTS:
            target = MODEL_ROOT / 'frozen' / relative
            target.parent.mkdir(parents=True, exist_ok=True)
            target.write_bytes(archive.read(PREFIX + relative))
    (MODEL_ROOT / 'data').mkdir(exist_ok=True)
    FIGURE_DIR = MODEL_ROOT / 'appendix_figures'
    FIGURE_DIR.mkdir()
    VERSIONS = {
        'python': platform.python_version(), 'numpy': np.__version__,
        'scipy': scipy.__version__, 'matplotlib': matplotlib.__version__,
    }
    RESULTS, RUN_SECONDS = {}, {}
    print('Current versions:', VERSIONS)
    print('Four original source hashes checked. Results:', MODEL_ROOT)
    print('Full defaults; no external data or model network calls.')

    def read_json(relative, frozen=False):
        base = MODEL_ROOT / 'frozen' if frozen else MODEL_ROOT
        return json.loads((base / relative).read_text())

    def read_csv(relative, frozen=False):
        base = MODEL_ROOT / 'frozen' if frozen else MODEL_ROOT
        with (base / relative).open(newline='') as handle:
            return list(csv.DictReader(handle))

    def run_model(relative, *arguments):
        print('Running', relative, 'with original defaults...', flush=True)
        start = time.perf_counter()
        command = [sys.executable, str(MODEL_ROOT / relative)]
        command.extend(map(str, arguments))
        completed = subprocess.run(
            command, cwd=MODEL_ROOT, check=True, text=True, capture_output=True
        )
        RUN_SECONDS[relative] = time.perf_counter() - start
        log = MODEL_ROOT / (Path(relative).stem + '_stdout.txt')
        log.write_text(completed.stdout)
        print(f'Completed in {RUN_SECONDS[relative]:.2f} seconds; log:', log)
        if completed.stderr:

```

```

        print('Program diagnostics:', completed.stderr)
    return completed.stdout

def compare_csv(relative, tolerance=1e-10):
    actual = read_csv(relative)
    frozen = read_csv(relative, frozen=True)
    assert len(actual) == len(frozen), (relative, 'row count')
    for actual_row, frozen_row in zip(actual, frozen):
        assert actual_row.keys() == frozen_row.keys()
        for key, expected in frozen_row.items():
            value = actual_row[key]
            if value == expected:
                continue
            try:
                close = math.isclose(
                    float(value), float(expected),
                    rel_tol=tolerance, abs_tol=tolerance
                )
                assert close, (relative, key, value, expected)
            except ValueError:
                raise AssertionError((relative, key, value, expected))
    print('Matches recorded results within tolerance:', relative)
    return len(actual)

plt.rcParams.update({
    'font.family': 'DejaVu Sans', 'font.size': 9,
    'axes.titlesize': 10, 'axes.labelsize': 9,
    'xtick.labelsize': 8.5, 'ytick.labelsize': 8.5,
    'legend.fontsize': 8, 'svg.fonttype': 'none', 'pdf.fonttype': 42,
    'axes.spines.top': False, 'axes.spines.right': False,
})

def finish_plot(name, source_files):
    fig = plt.gcf()
    fig.savefig(FIGURE_DIR / (name + '.png'), dpi=300)
    fig.savefig(FIGURE_DIR / (name + '.svg'))
    fig.savefig(FIGURE_DIR / (name + '.pdf'))
    for relative in source_files:
        shutil.copyfile(MODEL_ROOT / relative, FIGURE_DIR / Path(relative).name)
    plt.show()
    plt.close(fig)

```

D.2 Protocol invariants and omission traces: Sections 3–4

Question before execution: does a sender timeout establish cancellation at the receiver? This module exhaustively explores the declared finite transition systems. It prints the baseline counts, shortest omission witnesses, packed-word update results and the allowed trace in which a source withdrawal remains in transit. Read the action sequence as an ordering of abstract transitions, not measured network timestamps.

```

run_model('research/protocol_check.py', '--output', MODEL_ROOT / 'research/results')
protocol = read_json('research/results/protocol_results.json')
frozen_protocol = read_json('research/results/protocol_results.json', frozen=True)
assert protocol['source_sha256'] == SOURCE_HASHES['research/protocol_check.py']
for key in ('protocol_baselines', 'omission_runs', 'packed_word'):
    assert protocol[key] == frozen_protocol[key]
for row in protocol['protocol_baselines']:

```

```

    print(f"{row['operations']} ID(s): {row['reachable_states']} states; "
          f"{row['enabled_transition_instances']} transition instances; "
          f"{row['invariant_violations']} detected invariant violations")
assert protocol['protocol_baselines'][1]['reachable_states'] == 6651
for row in protocol['omission_runs']:
    trace = row['counterexample']
    print('\nOmitted control:', row['variant'])
    print('Shortest witness:', row['counterexample_length'], 'steps')
    print('Fault:', trace['fault'])
    for step in trace['steps']:
        print('  ->', step['action'])
print('\nPacked-word updates:')
for row in protocol['packed_word']:
    print(row['variant'], row['invariant_holds'])
allowed = read_json('research/results/trace_allowed_in_transit_withdrawal.json')
print('\nAllowed trace while withdrawal is in transit:')
for step in allowed['steps']:
    print('  ->', step['action'])
RESULTS['protocol'] = {
    'baseline_states': [
        row['reachable_states'] for row in protocol['protocol_baselines']
    ],
    'baseline_transitions': [
        row['enabled_transition_instances']
        for row in protocol['protocol_baselines']
    ],
    'omission_lengths': {
        row['variant']: row['counterexample_length']
        for row in protocol['omission_runs']
    },
    'packed_atomic_safe': protocol['packed_word'][0]['invariant_holds'],
    'packed_replacement_safe': protocol['packed_word'][1]['invariant_holds'],
    'frozen_results_match': True,
}

```

Expected outcome: the one- and two-identifier baselines contain 477 and 6,651 reachable states, with 1,487 and 26,905 enabled transition instances and no monitored invariant violations. Removing the final generation check, parent-age cap or terminal-evidence requirement yields witnesses of 6, 7 and 3 transitions. Atomic semantic OR preserves committed bits; stale whole-word replacement does not. The allowed in-transit trace shows why these local properties do not establish instantaneous global withdrawal. The checker does not verify cryptographic code, independent process failures or a production effect adapter.

D.3 Paired synthetic workloads: Section 6

Question before execution: can a wider protective hold reduce prohibited releases while also blocking more legitimate work? The program runs all 6,144 configured incidents: 128 paired seeds in each of 48 cells. The displayed comparison fixes twenty credentials and eight affected-record jobs. Credential decomposition is represented through an imposed semantic label and configured work cost; the program does not parse or verify real credentials. The plotting code reports means, with no claim that they estimate production performance.

```

run_model('code/run_workload.py')
assert read_json('data/workload_config.json') == read_json(
    'data/workload_config.json', frozen=True
)

```

```

)
assert compare_csv('data/workload_summary.csv') == 48
compare_csv('data/workload_paired_contrasts.csv')
workload_rows = read_csv('data/workload_incidents.csv')
assert len(workload_rows) == 6144
assert all(float(row['enforcement_delay']) <= 14.5 for row in workload_rows)
assert all(int(row['remote_per_decision_reads']) == 0 for row in workload_rows)
selected = [
    row for row in read_csv('data/workload_summary.csv')
    if row['k_pop'] == '20' and row['batch'] == '8'
]
print('k=20, batch=8: fault | path | mean seconds | releases | loss %')
for row in selected:
    print(f"{row['fault']:18s} {row['backend']:7s} "
          f"{float(row['enforcement_delay_mean']):8.3f} "
          f"{float(row['harmful_releases_mean']):9.3f} "
          f"{100 * float(row['legitimate_loss_fraction_mean']):8.3f}")
contrasts = [
    row for row in read_csv('data/workload_paired_contrasts.csv')
    if row['k_pop'] == '20' and row['batch'] == '8'
    and row['metric'] == 'enforcement_delay'
]
print('\nPaired ledger-minus-mirror delay in seconds; 128 incidents:')
for row in contrasts:
    print(row['fault'], round(float(row['ledger_minus_mirror']), 6))

faults = ['push', 'lost_push', 'origin_dns_outage', 'partition']
labels = ['Push delivered', 'Push lost', 'Origin / DNS outage', 'Full partition']
fig, axes = plt.subplots(2, 1, figsize=(6.6, 4.4), sharex=True)
styles = [('mirror', 'o', '#365f80'), ('ledger', 's', '#267566')]
for backend, marker, color in styles:
    rows = [
        next(row for row in selected
              if row['backend'] == backend and row['fault'] == fault)
        for fault in faults
    ]
    axes[0].plot(
        range(4), [float(row['enforcement_delay_mean']) for row in rows],
        marker, color=color, markersize=6, label=backend + ' model'
    )
    axes[1].plot(
        range(4),
        [100 * float(row['legitimate_loss_fraction_mean']) for row in rows],
        marker, color=color, markersize=6, label=backend + ' model'
    )
axes[0].axhline(14.5, color='#777777', ls=':', lw=1)
axes[0].set(ylabel='Mean enforcement delay (s)', ylim=(0, 16.5),
            title='A Paired synthetic incidents: k = 20, batch = 8')
axes[0].legend(loc='upper left', ncol=2, frameon=False)
axes[1].set(ylabel='Legitimate work lost (%)', ylim=(-4, 100),
            title='B Availability cost under the same configured faults')
axes[1].set_xticks(range(4), labels)
for ax in axes:
    ax.grid(axis='y', alpha=.2)
    ax.set_xlim(-.25, 3.25)
fig.tight_layout(pad=1.0, h_pad=1.0)
finish_plot('figure_D1_workload', [
    'data/workload_summary.csv', 'data/workload_paired_contrasts.csv'
])
RESULTS['workload'] = {

```

```

'runs': len(workload_rows), 'cells': 48, 'seeds_per_cell': 128,
'frozen_results_match': True, 'selected_k20_batch8': selected,
'paired_enforcement_contrasts': contrasts,
}

```

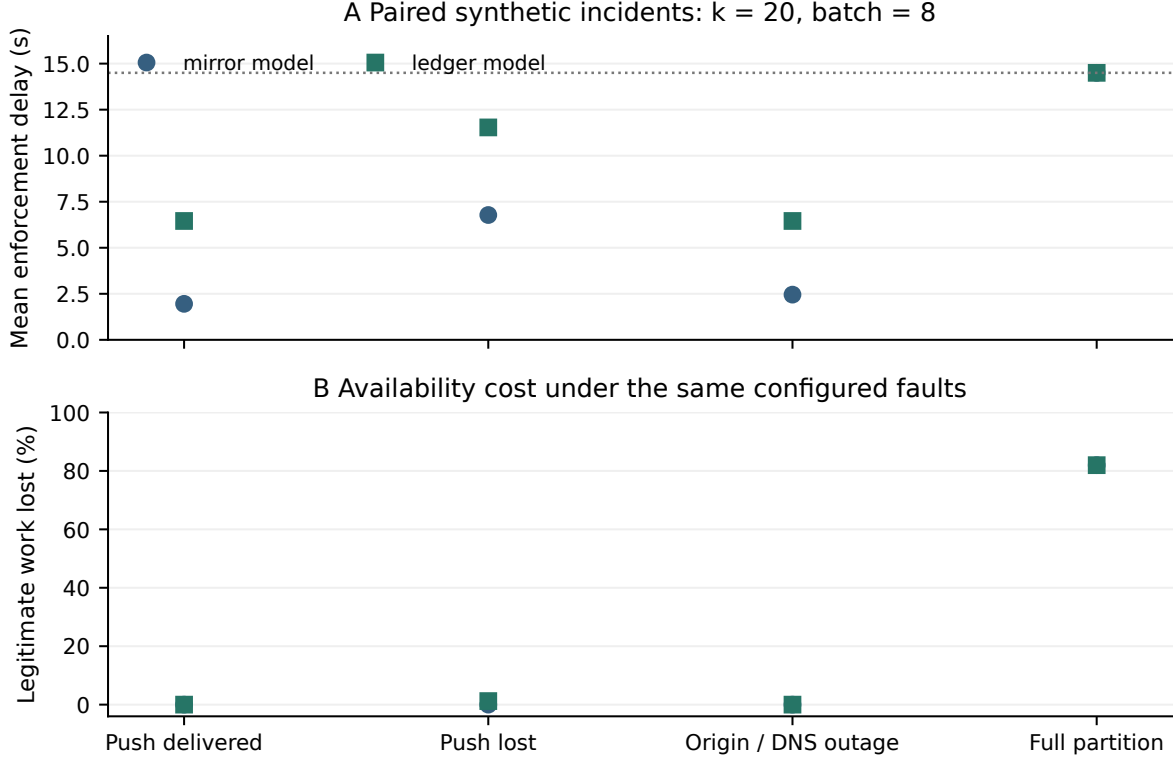


Figure D1: Selected outputs of the original paired workload model. Each plotted mean uses 128 synthetic incidents with $k_{\text{pop}} = 20$ and eight affected-record jobs. Panel A shows mean enforcement delay; the dotted line is the configured 14.5-second parent-age cap. Panel B shows the fraction of legitimate jobs not completed by their deadlines, excluding deadlines beyond the observation horizon. The two markers coincide under full partition. Differences follow the stipulated publication and fault rules; they are not measurements of a deployed ledger or proof of either backend’s universal superiority.

Expected outcome: delivered-push mean delay is about 1.955 seconds for the mirror model and 6.455 seconds for the ledger model. Both full-partition cases stop ordinary effects at 14.5 seconds and lose approximately 81.97% of legitimate work in this workload. Resident reads require no per-decision remote query in either model. These outcomes illustrate the trade-off between enforcing stale-state limits and maintaining service; they do not calibrate real-network timing or total distribution cost.

D.4 Fixed-cohort inference, masking and shared shocks: Sections 7–8

Question before execution: do equally frequent real and cover events remain indistinguishable when only genuine credentials share an incident? This module uses the original exact fixed-count posterior calculation, five sets of 1,000 synthetic cohorts and 30 codec trials. Each inference cohort contains exactly 32 genuine and 32 decoy positions. Classification is marginal: a posterior of at least 0.5 predicts genuine membership, with the program’s declared twelve-decimal tie handling. It does not require recovery of exactly 32 genuine labels.

```

run_model('experiments/decoys/decoy_experiment.py')
decoy = read_json('experiments/decoys/results.json')
assert decoy['cohorts_per_case'] == 1000
assert decoy['codec']['trials'] == 30
assert decoy['codec']['snapshots_per_trial'] == 181
assert compare_csv('experiments/decoys/privacy_summary.csv') == 5
compare_csv('experiments/decoys/shock_conditional_summary.csv')
assert decoy['exact_checks']['max_error_vs_bruteforce_L6_M3'] < 1e-12
assert all(row['masked_semantic_reconstruction_failures'] == 0
            for row in decoy['summary'])
print('case | marginal accuracy | mean maximum posterior')
for row in decoy['summary']:
    print(f"{row['case']:28s} {row['accuracy']:.6f} "
          f"{row['mean_maximum_posterior']:.6f}")
shock_rows = read_csv('experiments/decoys/shock_conditional_summary.csv')
shock_active = next(
    row for row in shock_rows
    if row['case'] == 'genuine_only_common_shock' and row['latent_shock'] == '1'
)
print('\nEvaluator-only genuine-shock stratum:', shock_active['cohorts'])
print('Stratum accuracy:', shock_active['accuracy'])
print('Mask reconstruction discrepancies across all 5,000 cohorts: 0')
print('History encoded-byte ratio:', decoy['codec']['ratio_total_means'])
frozen_codec = read_json('experiments/decoys/results.json', frozen=True)['codec']
assert math.isclose(decoy['codec']['ratio_total_means'],
                    frozen_codec['ratio_total_means'], rel_tol=1e-10)

fig, axes = plt.subplots(2, 1, figsize=(6.6, 4.5))
case_labels = ['Matched', 'Half-rate\ncover', 'Double-rate\ncover',
               'Shared\nshock', 'Genuine-only\nshock']
axes[0].bar(range(5), [row['accuracy'] for row in decoy['summary']],
            color='#365f80', width=.55)
axes[0].axhline(.5, color='#555555', ls='--', lw=1)
axes[0].set_xticks(range(5), case_labels)
axes[0].set(ylim=(0, 1), ylabel='Marginal accuracy',
            title='A Five cases; 1,000 fixed-count cohorts per case')
codec_rows = read_csv('experiments/decoys/codec_curve.csv')
styles = [('unmasked', 'Unmasked', '#365f80', '-'),
          ('fixed_mask', 'Fixed mask', '#a84c22', '--')]
for profile, label, color, line in styles:
    rows = [row for row in codec_rows if row['profile'] == profile]
    axes[1].plot(
        [int(row['tick']) for row in rows],
        [float(row['mean_encoded_bytes']) / 1024 for row in rows],
        color=color, ls=line, label=label
    )
axes[1].set(xlabel='Publication tick (not calibrated seconds)',
            ylabel='Encoded snapshot (KiB)', ylim=(0, 24),
            title='B Same semantic histories; 30 codec trials')
axes[1].legend(loc='lower right', frameon=False)
for ax in axes:
    ax.grid(axis='y', alpha=.2)
fig.tight_layout(pad=1.0, h_pad=1.1)
finish_plot('figure_D2_decoys_codec', [
    'experiments/decoys/privacy_summary.csv',
    'experiments/decoys/shock_conditional_summary.csv',
    'experiments/decoys/codec_curve.csv'
])
RESULTS['decoys'] = {

```

```

'cohorts_per_case': 1000, 'summary': decoy['summary'],
'codec': decoy['codec'], 'shock_active_stratum': shock_active,
'mask_reconstruction_failures': 0, 'frozen_results_match': True,
'checks': decoy['exact_checks'],
}

```

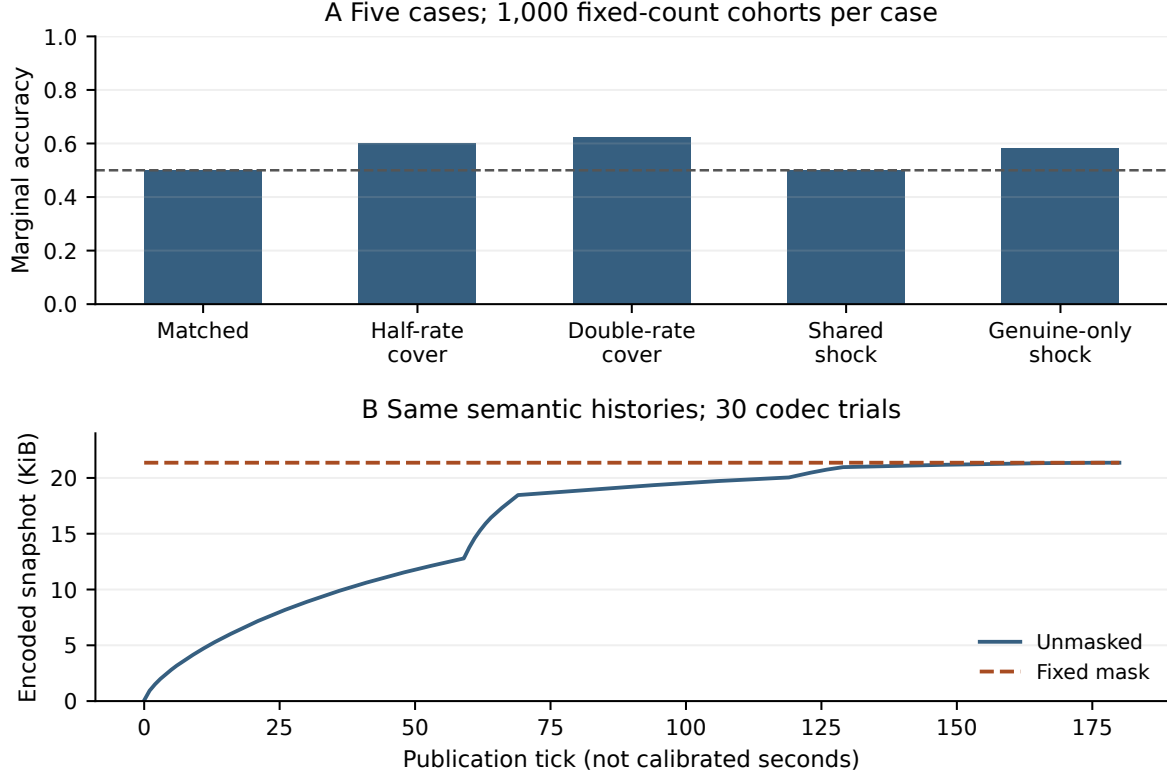


Figure D2: Original synthetic privacy and codec results. Panel A reports mean marginal classification accuracy for 1,000 cohorts per case; the dashed line is 0.5. The genuine-only-shock bar pools shock-active and inactive cohorts. It must not be confused with the separately printed evaluator-only shock-active stratum. Panel B uses identical semantic histories for masked and unmasked publication, with 30 trials, 131,072 positions and 181 snapshots. One kibibyte (KiB) is 1,024 bytes. Sizes include GZIP, unpadded base64url and the Multibase prefix, but exclude signatures and transport. Known active initialization permits exact semantic reconstruction from the masked history; these curves are payload measurements, not privacy scores.

Expected outcome: matched hazards and a correctly shared common shock give marginal accuracy 0.5. Half-rate and double-rate cover yield approximately 0.600891 and 0.622734. The evaluator-only genuine-shock-active stratum contains 274 cohorts and has accuracy about 0.79140. All 5,000 masked histories are reconstructed without discrepancy. Mean total encoded bytes across 181 snapshots are 2,978,690.1 unmasked and 3,959,757.2 masked, a ratio of 1.32936. The growing unmasked density explains why this ratio differs from Article A’s sparse equal-population snapshot factors. A passing attack experiment alone is not an all-observer privacy proof.

D.5 Continuing roots and parameter uncertainty: Section 9

Question before execution: does a subcritical propagation model give a fixed total harm limit when new external incidents keep arriving? The final module evaluates the article’s conditional two-type matrix envelope and continuing-root input. The enlarged parameter rectangle is a

deterministic sensitivity set, not a statistical confidence interval. The program returns no all-generation certificate when the dominating matrix is not subcritical. It does not replace an unavailable certificate with an observed infinite loss.

```

run_model('code/harm_envelope.py')
harm_points = read_json('data/harm_exact_points.json')
assert harm_points == read_json('data/harm_exact_points.json', frozen=True)
selected_delays = [2.7, 7.2, 12.7, 14.5]
print('configured delay | nominal T=60 bound | enlarged-parameter bound')
for delay in selected_delays:
    nominal = next(row for row in harm_points
                    if row['configured_delay'] == delay and row['uncertain'] == 0)
    enlarged = next(row for row in harm_points
                    if row['configured_delay'] == delay and row['uncertain'] == 1)
    text = (f"{enlarged['bound']:.5f}" if enlarged['bound'] is not None
            else 'certificate unavailable')
    print(f"{delay:16.1f} {nominal['bound']:.19.5f} {text}")
first = next(row for row in harm_points
              if row['configured_delay'] == 2.7 and row['uncertain'] == 0)
assert first['exact_bound'] == '420117/53834'
last = next(row for row in harm_points
             if row['configured_delay'] == 14.5 and row['uncertain'] == 1)
assert last['bound'] is None
print('Exact first bound:', first['exact_bound'])

fig, axes = plt.subplots(2, 1, figsize=(6.6, 4.5))
delay_rows = read_csv('data/harm_delay_curve.csv')
styles = [(0, 'Nominal parameters', '#365f80', '-'),
          (1, 'Enlarged parameter rectangle', '#a84c22', '--')]
for flag, label, color, line in styles:
    rows = [row for row in delay_rows if row['uncertain'] == flag and row['bound']]
    axes[0].plot(
        [float(row['configured_delay']) for row in rows],
        [float(row['bound']) for row in rows], color=color, ls=line, label=label
    )
axes[0].set(ylim=(0, 150), xlim=(0, 16),
            xlabel='Configured covered-influence delay (s)',
            ylabel='Expected-harm bound',
            title='A T = 60 s; displayed vertical range stops at 150')
axes[0].legend(loc='upper left', frameon=False)
horizon_rows = read_csv('data/harm_horizon_curve.csv')
for delay, line in zip(selected_delays, ['- ', '-- ', '-.', ':']):
    rows = [row for row in horizon_rows if float(row['configured_delay']) == delay]
    axes[1].plot(
        [float(row['horizon']) for row in rows],
        [float(row['bound']) for row in rows], ls=line, label=f'delay {delay:g} s'
    )
axes[1].set(xlabel='Root-arrival horizon T (s)', ylabel='Expected-harm bound',
            title='B Nominal parameters with continuing external roots')
axes[1].legend(loc='upper left', ncol=2, frameon=False)
for ax in axes:
    ax.grid(alpha=.2)
fig.tight_layout(pad=1.0, h_pad=1.1)
finish_plot('figure_D3_harm', [
    'data/harm_delay_curve.csv', 'data/harm_horizon_curve.csv',
    'data/harm_exact_points.json', 'data/harm_parameters.json'
])
RESULTS['harm'] = {'points': harm_points, 'frozen_results_match': True}

```

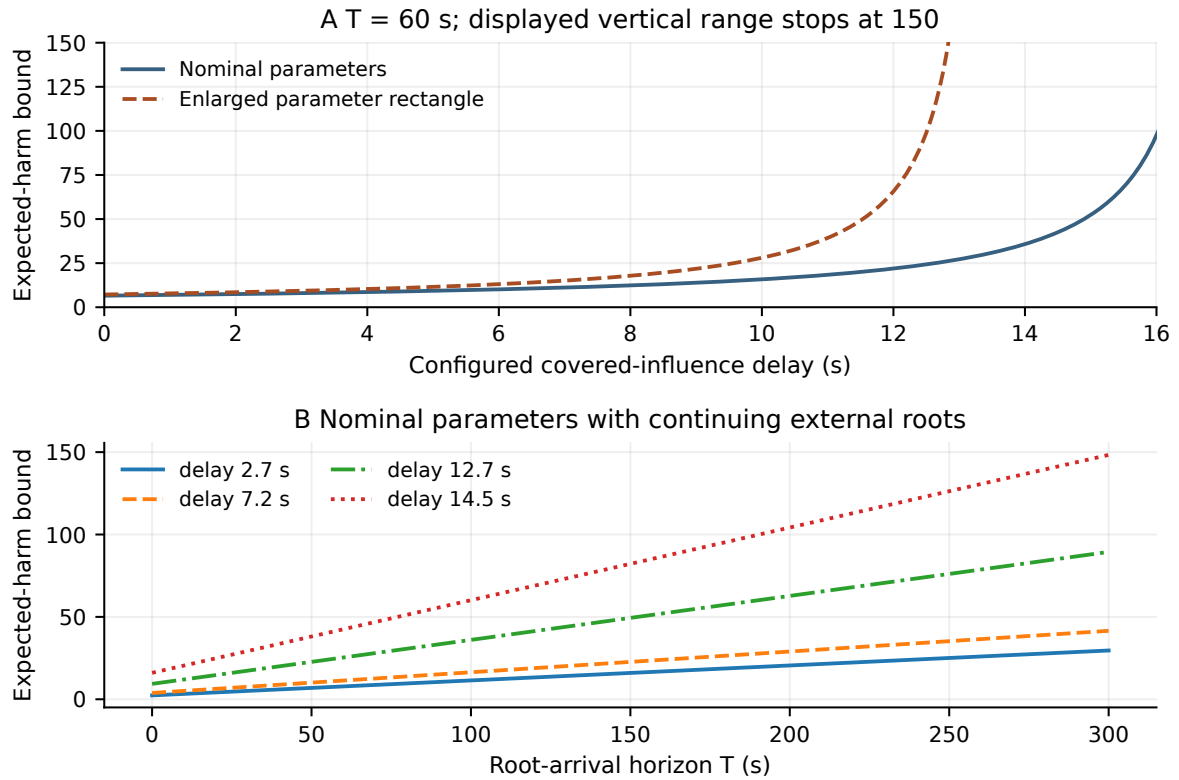


Figure D3: Calculated conditional expected-harm envelopes from the original program. Panel A fixes the root-arrival horizon at 60 seconds and compares nominal parameters with the enlarged parameter rectangle. The displayed vertical range ends at 150; values above this are clipped by the axis, and no value is defined where the subcritical certificate is unavailable. Panel B uses nominal parameters and increases the horizon during which external roots arrive; descendants of those roots remain included beyond that horizon. The curves are conditional mathematical bounds, not simulated or observed losses. Their use requires the uniform source, influence, cost and receiver-enforcement premises in Section 9.

Expected outcome: at horizon 60 seconds, nominal bounds for configured delays 2.7, 7.2, 12.7 and 14.5 seconds are approximately 7.80393, 11.36262, 25.36457 and 42.50278. The first bound is exactly 420117/53834 for the rational input fixture. The enlarged rectangle has no subcritical certificate at the last delay. Continued external roots increase the horizon bound; subcritical propagation does not remove new root incidents. These inputs are stipulated envelopes, not fitted field estimates.

D.5.1 Inspect one exact harm calculation

This cell can be run on its own without the ZIP or earlier cells. It uses only the Python standard library. It retains the original short calculation, placed inside a function to keep its variables separate from the four-module driver.

The following short cell uses the same rational parameters and matrix orientation as the original harm program for the nominal 2.7-second delay and 60-second horizon. Its exact guard applies specifically to a nonnegative two-by-two matrix: both diagonal entries must be below one and the displayed determinant must be positive. This avoids a floating-point threshold decision when readers change the parameters. It prints 420117/53834 and approximately 7.80393. It is an explanatory calculation, not a separate simulation or a measured risk estimate.

```
def exact_harm_example():
    from fractions import Fraction as F
    d, T = F(27, 10), F(60)
    D = [F(2), F(4)]
    B = [[F(2, 100), F(3, 100)], [F(4, 100), F(1, 100)]]
    K = [[B[i][j] * (D[j] + d) for j in range(2)] for i in range(2)]
    a, b = K[0]
    c, e = K[1]
    assert all(value >= 0 for row in K for value in row)
    roots = [1 + T/F(100), T/F(50)]
    det = (1-a)*(1-e) - b*c
    if not (a < 1 and e < 1 and det > 0):
        print('Certificate unavailable')
    else:
        x = [((1-e)*roots[0] + b*roots[1])/det,
              (c*roots[0] + (1-a)*roots[1])/det]
        bound = x[0] + 3*x[1] + F(1, 2)
        print(bound, round(float(bound), 5))

exact_harm_example()
```

D.6 Reproduction record and implementation boundary

Run the final cell to retain environment versions, source hashes, checks and results. Figures are exported as editable Scalable Vector Graphics (SVG), vector Portable Document Format (PDF) and Portable Network Graphics (PNG) images at 300 dots per inch. Their source tables are copied beside them. The recorded execution time measures only these local model programs. Appendix B and Section 11.5 identify the additional implementation tests; Appendix C states the field evidence still required.

```
print('Local execution seconds:', json.dumps(RUN_SECONDS, indent=2))
print('All four modules used original source files and full defaults.')
```

```

REPRODUCTION_RECORD = {
    'versions': VERSIONS, 'source_hashes': SOURCE_HASHES,
    'local_execution_seconds': RUN_SECONDS, 'results': RESULTS,
    'google_hosted_runtime_tested': False,
}
record_path = MODEL_ROOT / 'printed_appendix_reproduction_record.json'
record_path.write_text(json.dumps(REPRODUCTION_RECORD, indent=2))
print('Reproduction record:', record_path)
print('Vector figures, 300 dpi images and source data:', FIGURE_DIR)

```

Changing inputs defines a new experiment. Record those changes and do not relabel new results as reproduction of the fixed study. A mismatch should be investigated rather than bypassed by removing an assertion. In particular, equal finite test results do not establish source truth, dependency completeness, all-history privacy, standards conformance or production containment deadlines.

E Implementation conformance and standards contribution fixtures

E.1 Separate the standard, the profile and the experiment

A fixture is a saved input with a stated expected result. A conformance package needs accepted, rejected and boundary examples. Each fixture should identify the specification version, producer input, encoded object, expected decoded value and expected validation decision. Preserve the exact signed object when testing cryptographic verification. Equivalent semantic content is not necessarily the same signed byte sequence.

Article A, Appendix E, recommends where to propose W3C changes [1]. Here the immediate baseline remains Bitstring Status List v1.0, the 15 May 2025 Recommendation [5]. The v1.1 document inspected on 9 October 2026 is the 24 September First Public Working Draft [6]. Neither the model checker nor the teaching notebook establishes conformance to these documents.

Both versions recommend random index assignment in Section 2.1. Their Section 3.3 starts generation from zero bits and assigns the statuses of issued credentials. Non-normative Section 6.5 leaves decoy algorithms outside scope [5,6]. Unpredictable index allocation therefore differs from randomizing published status values. Production randomness needs an appropriate cryptographic source; fixed seeds belong to reproducible experiments.

E.2 Minimum producer and reader fixtures

These requirements are proposed for this article's deployment profile. They are not additional W3C requirements. A reader cannot generally infer from the published list whether a set position belongs to an issued credential. Producer tests therefore require the corresponding issuance inventory.

Table 15: Proposed producer and reader conformance fixtures.

Layer	Positive fixture	Negative or boundary fixture	Required interpretation
Ordinary list	Generate known issued statuses; decode the exact expected entries.	Nonzero unissued position; wrong purpose; out-of-range index; truncated compression.	Test producer output separately from reader rejection and application policy.
Bit order	Set positions 0, 7, 8, 255 and 256 and round-trip them.	Reverse byte or word bit order.	The adapter preserves the standard's left-most index-zero convention.
Terse entry	Convert 0, $2^{26} - 1$, 2^{26} and $2^{32} - 1$.	Negative or oversized index; wrong list length or supplied purpose.	Bind the reconstructed locator and purpose to the verified credential.
Experimental privacy	Decode both mask values under an explicit method; record matched cover history.	Missing mask; unsupported method; repeated decoy reversal; real-only burst.	Fail unsupported decoding; distinguish a decoding error from a privacy counterexample.
Freshness	Accept a recognized, authenticated state within its age and validity limits.	Replay old state; redistribute it with a fresh retrieval time; lose clock assurance.	Redistribution does not refresh the source watermark.
Signature profile	Verify the selected suite, signed bytes, context and recognized key.	Change payload or context; withdraw the key; strip a required proof.	Cryptographic verification and authorization remain separate decisions.

The terse cases target v1.1 Sections 2.3 and 3.5. Its fixed 2^{26} entries per list and verifier-supplied purpose must remain explicit [6]. A small laboratory list does not exercise that profile's decoding and memory requirements. Resource-limit tests should include failed decompression and verification paths, not only successful reads.

E.3 Packed-word updates require their own contract

ERC-7506 remains a draft Trusted Hint Registry. Its setHint interface accepts a bytes32 value; it does not define this article's packed status semantics [15]. Test the custom adapter with two writers revoking different bits from the same old word. Both withdrawals must survive. Include retries, changed delegates, initialization, upgrades and every route that can replace a value.

Let C be the current semantic revocation word, F the new revocation mask and r the fixed initial mask. For irreversible revocation, update current semantic state by bitwise OR as $C' = C \vee F$. With masking, publish $Y' = r \oplus C'$. A one-bit example exposes the difference: $r = 1$ and $C = 0$ give $Y = 1$. Revocation changes C to 1 and Y to 0. Thus a valid semantic withdrawal can clear a wire bit. Applying OR to the wire word would fail this case. Blind XOR retries can undo a prior change. The fixture should test semantic monotonicity and idempotence, not assume monotonic wire bits. If the contract cannot inspect semantic state, additional state or a construction-specific proof is required.

E.4 Evidence needed before a standards contribution

Submit generation and decoding questions to the bitstring specification work identified in Article A. First clarify the relationship between Sections 3.3 and 6.5 for unissued decoys. Keep any masked method separate, with explicit types, authenticated parameters and rejection rules. Do not claim that an ordinary reader's ability to parse the bytes establishes producer conformance.

Attach the fixture inventory, independently reproduced outputs and the declared observer transcript. Include known-initialization leakage, snapshot differences, mismatched hazards, cohort renewal and public update metadata. Passing finite simulations supports only their tested cases; it does not prove a full-history privacy bound.

ML-DSA under FIPS 204 supplies a signature primitive [17, 18]. A deployment still needs its credential envelope, key recognition, transport and ledger authorization profiles. A changed credential signature alone does not establish end-to-end post-quantum protection. Likewise, a valid signature or finalized ledger state does not create an authoritative completeness watermark. Freshness fixtures must test that separate source contract and the receiver's enforced expiry. The model package supplies examples and proof obligations; a conforming wire implementation and interoperability report remain deployment work.

F Evaluation plan for anonymous status adapters

This appendix defines planned implementation experiments. They have not been executed for this article, and no performance or privacy results are claimed. Pin the cryptographic library, registry method, proof format, publication policy and software revision. Compare accumulator-backed AnonCreds with the existing bitstring adapter under the same action labels, authoritative withdrawals, accepted-age caps and receiver semantics [19]. A Merkle commitment with a zero-knowledge membership proof is a separate profile; zk-creds provides a research example rather than an interchangeable AnonCreds implementation [24].

Table 16: Planned evaluation fixtures for anonymous status adapters

Test family	Controlled variation or fault	Required observations and acceptance condition
Credential and context binding	Substitute another credential's witness; replay the challenge, audience or operation context.	Reject mismatched bindings; retain valid negative fixtures and distinguish proof rejection from policy denial.
Root, epoch and freshness	Replay an older root; delay installation; inject an unjustified future watermark.	Preserve monotonic accepted state and existing age caps; record withdrawal-to-enforcement time separately from proof validity.
Witness and publication availability	Delay pending publication, lose updates, or disconnect the tails or path service.	Record each publication milestone and witness recovery cost; no effect is released after its reliance cap expires.
Hidden-index invalidation	Compare controller-visible scoped handles with registry-wide invalidation.	Measure affected decisions, revalidation work and legitimate holds; document which participant can correlate dependencies.

Continued on next page

Table 16 (continued)

Test family	Controlled variation or fault	Required observations and acceptance condition
Resource saturation	Vary registry population, credential multiplicity, update bursts and verification queues.	Report proof bytes, generation and verification time, update bytes and deadline misses; test bounded rejection paths.
Cross-domain persistence	Crash after dispatch or commit; lose closure receipts; rotate registry state during recovery.	Preserve unresolved operations and accepted generations; obtain qualified terminal evidence before closure.

For registry a at epoch e , let $D_{a,e}$ contain its live dependent decisions. Under registry-wide invalidation, a work-accounting model is

$$W_{a,e}^{epoch} = \sum_{o \in D_{a,e}} (w_o^{hold} + w_o^{revalidate}).$$

Count the work actually scheduled; completed holds and later re-admission may have separate queues and deadlines. Use this workload in the supported service envelope rather than counting only revoked credentials. Article A, Appendix F, supplies the broader byte and privacy accounting. Preserve setup, tails or path distribution, state updates and proof transfers as separate terms. Test cold and warm caches. Include observable registry identifiers, update timing and retrieval traffic in the privacy transcript. The existing $40.03\times$ and $1475.44\times$ codec ratios do not estimate these profiles' total costs.