

Citation Receipts: Auditing Retrieval-Backed Citation Events in Digital Libraries

Kumar Abhinav

rushtoabhinavin@gmail.com

AiDash <https://orcid.org/0009-0009-1839-841X>

Research Article

Keywords: Digital libraries, Data provenance, AI citations, Transparency logs, Retrieval-augmented generation, Auditability

Posted Date: October 28th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11514620/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.
[Read Full License](#)

Additional Declarations: The authors declare potential competing interests as follows: The author personally owns a filed patent application related to the Citation Receipt Protocol. The application is disclosed as a potential financial competing interest.

Citation Receipts: Auditing Retrieval-Backed Citation Events in Digital Libraries

Kumar Abhinav

AiDash Inc., Palo Alto, United States

Email: rushtoabhinavin@gmail.com

ORCID: <https://orcid.org/0009-0009-1839-841X>

Abstract

Purpose: AI-assisted discovery can present a scholarly citation without leaving a publisher-verifiable event record. We study how digital libraries can audit one narrow class of source use: visible citations backed by authenticated retrieval. **Methods:** The Citation Receipt Protocol signs versioned source and response bindings, commits epoch sets to a Merkle log and audits sampled omissions. We separate provider declarations, client observations and publisher delivery. Evaluation includes controlled selective suppression, audit recognition, withholding, inflation, current-schema benchmarks, two distinct live-model studies and a saved-output suppression replay. **Results:** The preserved 300-call post-cutoff study produced 190 eligible receipts, all verified. A fresh 60-call context-intervention study produced 20 verified receipts in local replay with the target available and none when it was excluded or context was empty. A local replay of twenty saved answers detected two deliberate receipt omissions, with no false alarms. Selective-suppression experiments exposed the difference between pooled and publisher-conditioned sampling. Recognized probes passed despite omissions in organic traffic; valid committed receipts could still be withheld or fabricated. The current fixture measured a 935-byte payload and a 1,171-byte signed envelope. **Conclusion:** CRP supplies auditable provenance records and conditional sample-based checks. It does not certify global completeness, authentic user demand, semantic citation support or training influence. Digital-library adoption requires explicit sampling, visibility, delivery and privacy policies.

Keywords: Digital libraries, Data provenance, AI citations, Transparency logs, Retrieval-augmented generation, Auditability

1 Introduction

Language models are now used to write literature reviews, related-work sections and complete research drafts [19, 20]. Their citations matter because they point to sources and make claims about provenance. Some generated references are fabricated or wrong [33]. Even when a reference is real, an outside reader cannot tell whether the serving system fetched that paper for the answer, cited it through another source or generated the reference without a recorded source event.

A measurement gap is emerging: publications can influence thousands of generated answers without gaining a conventional scholarly citation. A crawler fetch is visible to the publisher, but a citation inside an answer is normally visible only to the AI provider and the user. Fetch and citation counts differ. One cached fetch can support many answers, while many fetched papers never appear in an answer. Existing page analytics therefore cannot by themselves produce a trustworthy paper-level AI citation count.

We address one part of this problem: *inference-time citations backed by source retrieval*. Suppose a model has a January 2026 training-data cutoff, a paper is published in July 2026 and a user asks about its result in August. If the system retrieves and cites the paper, the paper should receive a machine-readable credit event. That event

should be tied to the exact paper version and answer, rather than depending on the provider's unaudited aggregate report.

For digital libraries, this is a provenance problem before it is a bibliometric problem. A repository access log records acquisition of an artifact. A citation in a generated response records a later presentation event, potentially after caching or extraction. These activities have different identifiers, retention policies and observers. A library needs to preserve their connection without treating either activity as proof of intellectual influence.

CRP supports a library's audit record of recorded, retrieval-backed citation events. A publisher can check content and license bindings against its own source history. An auditor can challenge a sample of events. An index can store version-specific records before work-level aggregation. Separating source entities, generation activities and responsible providers is consistent with provenance modeling such as PROV-DM [22]. Research-quality assessment remains a separate task.

We present the Citation Receipt Protocol (CRP). A provider creates a signed receipt when its serving pipeline renders a visible citation to a fetched source. The receipt is delivered to the source owner and committed to a public append-only log. An auditor can then test whether the provider is omitting receipts by sending ordinary-looking probe queries and later asking the provider to open the matching commitments.

The paper makes five contributions:

1. A signed receipt format connecting a versioned scholarly source, authenticated fetch, license, generation event and response commitment.
2. A conditional omission test and a sampling analysis that distinguishes the probe population from a publisher's eligible citation traffic.
3. A client-observation verification profile and sampled delivery reconciliation, with explicit limits on visibility, inflation and recipient completeness.
4. A reference implementation tested with controlled selective-suppression, audit-recognition, withholding and fabricated-event attacks.
5. Separate live-model studies and measurements of the current receipt schema, including all enabled artifact and event fields.

CRP addresses recorded citation events. Training attribution and citation faithfulness require separate evidence and methods.

2 Related work and scope

Usage, attention and bibliographic links.

COUNTER Release 5.1 standardizes usage reports for online collections. Altmetrics extends measurement to activity beyond bibliographic citations [12, 25, 26]. The Altmetric Attention Score weights tracked mentions, while PlumX separates citations, usage, captures, mentions and social media [2, 13]. These counting units differ from a source displayed in one observed AI answer. CRP asks whether that event has a matching signed receipt under a prior epoch commitment.

OpenCitations provides reusable scholarly citation data, and COCI records open DOI-to-DOI links derived from Crossref metadata [16, 24]. Links between works differ from repeated source displays by an AI service. Crossref Event Data historically collected online mentions, but its API was retired on 23 April 2026. Its successor concerns deposited data-citation relationships [28]. Recent digital-library research evaluates LLMs as information-access layers [10]. CRP addresses the evidence required to account for the citations they display.

Transparent statements.

The transparency architecture of SCITT is specified in RFC 9943 [8]. Registration can authenticate a statement without proving its truth or requiring the issuer to register every event. Similar considerations apply to CRP's citation-event accounting and sampled client observations. The current JSON/Ed25519 implementation is not COSE/SCITT-conforming.

Crawler identity and licensing.

An active IETF Internet-Draft describes HTTP message signatures for automated traffic. Such signatures allow servers to verify the identity of bots that retrieve content [21]. RSL expresses machine-readable content terms [30]. These systems concern who retrieved content and under which terms. They do not by themselves record whether that content later appeared as a citation in an answer.

Content-use reporting.

Content Telemetry version 1.0 defines events for retrieval, grounding, citation, presentation and engagement [32]. The Aegon design connects license tokens, a Certificate Transparency-style ledger and signed provenance events through stages that include retrieval and citation [6]. Both describe reported source-use events. A signed event proves that a reported event was not altered, but it cannot reveal a visible citation event that the provider never reported. CRP adds a probe audit for that omission case.

Receipts and transparency logs.

Certificate Transparency implements an append-only Merkle-log model [17, 18]. Similar concepts are applied in Sigstore for software signing [23]. Figuera's Notarized Agents design has a receiving service sign what it observed and publish the receipt to a witness-backed log [14]. CRP uses the opposite trust direction for citation events: the provider signs a receipt, the cited publisher verifies it against its own fetch history, and an auditor tests the provider's reporting completeness.

Citation quality.

Earlier work investigated whether generated citations support their claims [9, 15]. CRP records that a visible citation was rendered for a source that had been fetched. Semantic citation checks assess a different property.

Content Telemetry permits origin/CDN corroboration for retrieval, while citation and grounding remain agent-reported. Event signatures do not provide a mandatory omission audit. Notarized Agents uses receiver-attested evidence and describes an unimplemented heartbeat approach for suppression detection. Its trust boundary differs from provider-signed citation reports. Table 1 contrasts the stated event and audit mechanisms rather than treating the systems as equivalent.

Table 1. *Closest systems. "Omission audit" means a method for testing whether a provider failed to report visible citation events.*

System	Citation/use event	Fetch/license link	Public commitment	Omission audit
Content Telemetry	Yes	Partial	No	No
Aegon	Yes	Yes	Yes	No
Notarized Agents	General actions	Authorization-bound	Yes	Future heartbeat sketch
CRP	Visible citations	Yes	Yes	Yes, under A1–A4

3 Citation Receipt Protocol

3.1 Parties and event definition

Four parties take part. A provider V runs a search-enabled generation service. A publisher P serves a paper or other source, keeps an authenticated fetch log, and receives receipts for its content. A transparency log L commits to hourly receipt sets. An auditor A sends queries through the same channel as ordinary users. Fig. 1 shows the main message flow.

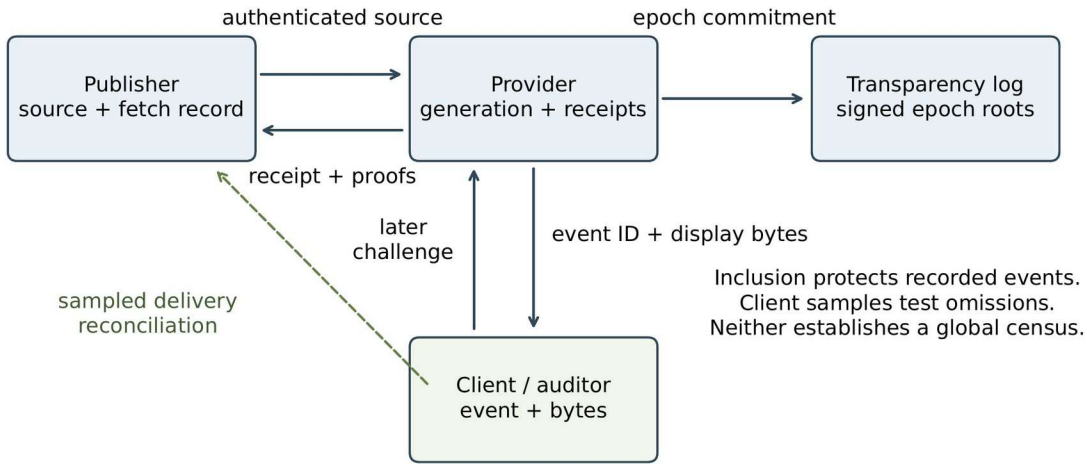


Fig. 1 CRP flow. The publisher observes the authenticated fetch; the provider observes the visible citation; the signed receipt connects the two and the public epoch root freezes the receipt set before an audit challenge

A visible citation event is a tuple (g, r, s) , where g is a unique generation-event identifier, r is the rendered response byte string, and s is a source that the provider displays as a citation in r . A receipt is not created merely because s appeared in the retrieval context. Hidden grounding can be reported as a separate event class, but its omission is not observable to the audit in Section 4; it is not counted as a visible AI citation in this paper.

A client or auditor also records the response it receives, the exposed generation-event identifier and the source markers it observes. This client record is outside the provider's receipt-construction path. At epoch close, the client verifies and pins the signed, logged manifest before any challenge. Its configuration names the expected log signer separately from the provider signer; accepting any registered key as the log would allow a provider to substitute its own signed tree head. Distinct keys establish the configured identity boundary, not operational independence. In the strict observation profile, the opening must match this prior anchor, expected provider issuer, event identifier, epoch, canonical source and use class. A receipt supplied for another event or epoch root is rejected even if its signature is valid. Missing event identifiers or prior manifest anchors are reported as unauditible rather than silently matched by response text.

The client's record supports an audit of the response it observed. E9 uses plain response bytes, and E11 uses a deterministic rendering of the JSON answer field. The delivery design below outlines how a production client could expose and retain the event handle. Section 13 collects the evaluation boundaries.

Even when responses are mostly different, a generation-event identifier is still needed. The post-cutoff experiment produced 11 groups of byte-identical responses covering 26 calls. A commitment over response bytes alone could not distinguish those calls. The current protocol uses a random UUIDv4 event identifier created before generation.

3.2 Event exposure and the client trust boundary

A production integration needs an event handle outside the receipt reply. We propose a versioned response envelope containing an answer string, a citation list and reserved `crp_meta` fields for the full event UUID, provider issuer, epoch, artifact profile and answer digest. The HTTP CRP-Event-ID header repeats the full UUID. A client-owned answer pane may display a short CRP event token beside a control that shows the full identifier. The short token serves as a lookup aid, not a collision-resistant identity. Verification always uses the full UUID.

For an answer-utf8-v1 profile, the committed artifact r is the exact UTF-8 encoding of the JSON-decoded answer string, including citation markers and its received whitespace. No Unicode normalization, line-ending conversion or HTML interpretation is applied. The JSON envelope, citation-list metadata, HTTP headers, `crp_meta`, displayed event token and digest field are excluded from r . This avoids a digest containing itself. The client renders the string as text in its own answer component and records the same bytes before an audit challenge. Rich HTML or a different visible-text serialization would require a separate declared profile; neither is silently equated with raw model JSON.

The client cross-checks the full UUID in the HTTP CRP-Event-ID header, reserved `crp_meta` field and token's full-ID view, recomputes the answer digest and records the visible source markers. After epoch close, it verifies the provider-signed manifest and its inclusion under the separately configured log signer, then pins that manifest before any challenge. A challenged receipt must match this prior manifest, expected issuer, epoch, full UUID, artifact bytes, source and use class. Provider-supplied identifiers, digests and manifests are necessary inputs but remain untrusted until these checks pass. Missing or conflicting aliases are coverage failures, not grounds for inferring an identifier from answer text.

Streaming needs an initial UUID followed by a completion record for the final answer bytes. A retry of the same delivered generation retains its UUID; a replacement answer receives a new one. An unfinished stream is recorded separately and is not counted as a completed event under this profile. The provider can control answer content and metadata, omit fields or recognize audit traffic. The audit assumes an honest client recording component outside the provider's receipt-construction path. TLS authenticates delivery from the endpoint; it does not establish truthful provider behavior. A compromised client, forged user demand and equivocation between client views require the additional controls discussed in Section 13.

3.3 Receipt contents

For visible citation event (g, r, s) in epoch e , the provider creates

$$\rho = (id, e, g, src(s), u, c, H(Lic), seq), \quad (1)$$

where id is a receipt identifier, $u = \text{visible_citation}$, and seq is an epoch sequence number. The source object contains:

- a canonical identifier, such as a DOI or arxiv:2607.12345v1;
- the source URL and authenticated fetch reference;
- a hash of the exact artifact, such as the arXiv v1 PDF; and
- a hash of the bytes served to the model, such as extracted text.

The version is part of the identifier: arXiv v1 and v2 are distinct protocol objects. Work-level merging belongs to the later index.

The response commitment is

$$\begin{aligned} c &= \text{SHA-256}(salt \parallel g \parallel r), \\ salt &= \text{HMAC-SHA-256}(k_e, id). \end{aligned} \tag{2}$$

with epoch key k_e held by the provider. The provider signs the RFC 8785 canonical JSON form [31]. The receipt-signing key should be purpose-specific and linked to the provider's verified crawler identity; it need not be the same key used for HTTP requests.

The new observation profile requires the generation-event identifier and the explicit `visible_citation` use class. Legacy receipts may omit event identifiers for byte compatibility, but they do not acquire the strict profile's event-attribution guarantee. The response byte representation must be agreed before verification. A semantic rendering, a JSON serialization and a browser document are different artifacts.

There is no user ID or query text in the receipt. A challenge reveals the salts for the corresponding citation receipts and the recorded response. It does not reveal the epoch key or unrelated response openings.

3.4 Epoch commitment and verification

At epoch close, the provider builds an RFC 6962 Merkle tree over canonical receipt payloads and signs a manifest containing the provider identity, epoch, receipt count, and Merkle root. The manifest is appended to L . A publisher receives its receipts with proofs from each receipt to the epoch root and from the epoch manifest to the public log.

A verifier checks, in order: the receipt signature; the manifest signature and log inclusion; receipt inclusion in the epoch tree; uniqueness of $(issuer, receipt_id)$; source, artifact, and license hashes; and the authenticated fetch reference. A fetch-log screen can also flag receipt totals that exceed a license-declared reuse cap. This screen limits some forms of inflation, but it is not a cryptographic proof that every signed receipt corresponds to a real user-visible event because the provider is the signer.

Global sequence gaps mix receipts from different publishers. Sampled delivery reconciliation compares a verified, committed receipt identifier from an observed citation with the identifiers received by the publisher. It reveals withholding within that observed sample. Publisher-wide delivery accounting requires the publisher-scoped commitments described below and independent event observation.

The journal revision also closes an implementation gap in artifact verification. When a receipt claims a PDF hash, publisher verification now compares it against the publisher's known PDF digest and rejects a missing or mismatched reference. The served-text digest is checked separately. A legacy receipt without a PDF field remains valid under the legacy schema, without a PDF-binding guarantee. Negative tests include a correctly signed receipt containing the wrong PDF digest.

3.5 Publisher-scoped delivery commitments

A publisher can reconcile a known sampled receipt identifier, but global sequence gaps mix receipts across publishers. A possible extension builds a separate Merkle tree for each publisher's committed receipt payloads, in the global epoch sequence order. For publisher p and epoch e , a directory leaf contains the publisher identifier, epoch, receipt count n_p and subroot R_p . A second Merkle tree commits these directory leaves in a predeclared canonical publisher-identifier order, including zero-count entries. A versioned signed epoch descriptor would bind both the existing global receipt root and the publisher-directory root. The recipient pins that descriptor before requesting delivery.

The publisher obtains its directory inclusion proof and complete receipt-payload list, rejects duplicate receipt identifiers, checks every signature and ownership binding, and recomputes n_p and R_p in the declared order. A

missing delivery changes the count or reconstructed subroot. A receipt-level proof would contain a path in the publisher tree and one in the directory tree. This establishes completeness of delivery relative to the pinned publisher commitment, subject to a trusted source-to-publisher registry. Unissued citations and authentic user demand require independent observations.

For N receipts and P roster entries, materializing cached trees from the ordered inputs takes $O(N + P)$ hash work and tree storage. A receipt's two inclusion paths contain at most $\lceil \log_2 \max(1, n_p) \rceil + \lceil \log_2 \max(1, P) \rceil$ sibling hashes in a balanced binary layout. At 32 bytes per SHA-256 hash, their sibling material is 32 times that sum, before identifiers, signatures and serialization. Recomputing a complete publisher list costs $O(n_p)$ hash work and requires delivery of its payload bytes. These are design costs, not measured E10 timings.

The directory alone does not prove that its publisher sets form a complete, correctly classified partition of the global root. An independent auditor would need to rebuild both commitments from the complete epoch leaves and a pinned ownership registry, with $O(N)$ receipt access, or use an additional proof system. Directory counts also disclose publisher activity; access controls or a different privacy mechanism would be needed where that disclosure is unacceptable. The current implementation has no publisher-subroot profile. The proposal needs a new manifest schema, partition validation, empty-roster handling and tests before its delivery guarantee can be used.

4 Auditing omitted receipts

A provider can omit a citation receipt by never creating it. A Merkle log cannot reveal an event that was never entered. CRP therefore uses an active audit.

The auditor submits queries and records each response and its visible citations. It pins the provider identity, epoch and signed manifest hash after epoch close and before revealing its challenges. The budget n counts eligible auditable source events, rather than all submitted queries. After the epoch closes, it challenges the provider to return a receipt for each probed citation, the salt that opens Equation 2, and the two inclusion proofs. A missing or invalid answer is one failure.

We assume: (A1) the provider cannot distinguish probes from ordinary traffic after conditioning on query content; (A2) receipt signatures are unforgeable; (A3) the hash and HMAC assumptions hold; and (A4) the configured log provides the standard Certificate Transparency accountability property, with the auditor checking the expected log signer and pinning the accepted signed epoch manifest before its challenge. Append-only logging alone does not enforce one provider manifest per epoch; the opening must match the pinned manifest. A1 includes the condition that suppression is not chosen based on whether a request is a probe.

Theorem 1 (Evasion bound).

If the provider independently suppresses each eligible visible-citation receipt with probability q , then the probability that all n eligible probed citations receive valid answers against the pinned commitment is at most $(1 - q)^n$. Equality holds in a suppression-only model where every nonsuppressed event receives a valid answer.

The proof is short. Once the epoch root is public, a missing leaf cannot be added without changing the committed tree. Under A1, each eligible probe independently survives suppression with probability $1 - q$. A suppressed event cannot receive a valid opening against the pinned root. Additional verification failures can only reduce the all-pass probability. Appendix A gives the formal statement.

Corollary 1 (Audit budget).

To exclude suppression at rate q_0 or higher after zero failures with one-sided confidence $1 - \alpha$, it is enough to run

$$n^*(q_0, \alpha) = \lceil \frac{\ln \alpha}{\ln(1 - q_0)} \rceil \quad (3)$$

clean probes. For $q_0 = 0.01$ and $\alpha = 0.05$, $n^* = 299$.

The zero-failure one-sided Clopper–Pearson upper bound is $1 - \alpha^{1/n}$ [11]. If the per-probe rates vary as q_i and suppression remains independent, the suppression-only evasion probability is $\prod_i (1 - q_i)$, which is at most $(1 - \bar{q})^n$ for average rate $\bar{q}$. The theorem does not cover probes a provider can identify, nor hidden grounding events that users cannot observe.

4.1 Population and sampling unit

The unit in Theorem 1 is an eligible observed citation, not a submitted query. A query can yield no citations, one citation or several correlated citations. A publisher's quota therefore advances only when an observed response visibly cites an eligible source from that publisher. A provider's refusal to cite a source is outside this omission test. Queries that fail to elicit an eligible citation remain in the query-cost report but are not counted as clean citation probes.

The audit reports observed, auditable and unauditable source-event totals separately. Missing event identifiers or manifest anchors are coverage failures, not clean probes. The confidence statement applies only to the auditable population. Selective identifier exposure or anchoring requires separate coverage reporting. Multiple sources in one response require a correlation-aware sampling design unless their suppression is independently justified. A missing or invalid answer indicates an audit failure, whose cause requires investigation.

The bound applies to the distribution from which probes are sampled. Let w_j denote publisher j 's share of eligible traffic and q_j its conditional omission rate. Under independent suppression and matching probe/traffic distributions, a pooled probe has omission probability $q_{pool} = \sum_j w_j q_j$. A small pooled rate can coexist with severe omission for a small publisher. Publisher-specific bounds require publisher-specific sampling.

For example, a publisher with share $w_j = 0.1$ and conditional omission $q_j = 0.01$, with no omissions elsewhere, produces $q_{pool} = 0.001$. A pooled budget of 299 has detection probability $1 - 0.999^{299} = 0.2586$. Achieving 95% detection against this specific alternative takes 2,995 pooled eligible citations. A publisher-conditioned sample requires 299 eligible citations from that publisher under the original assumptions. The larger query budget needed to elicit those citations is not fixed by this calculation.

For simultaneous statements about J prespecified publishers, assign per-publisher error levels α_j with $\sum_j \alpha_j \leq \alpha$.

If each publisher contributes n_j independent eligible clean probes, its upper limit is $1 - \alpha_j^{1/n_j}$. The union bound controls the familywise error without requiring independence across publisher statements. For ten publishers and equal error allocation at familywise 95% confidence, the one-percent threshold requires 528 eligible citations per publisher. Selection of publishers and stopping rules must be declared before inspecting failures.

Account identity, query wording, timing and service tier can reveal audit traffic even when the text appears ordinary. An operational sampling plan must address this risk. Probe indistinguishability remains an assumption. The audit-recognition experiment below tests its failure.

5 Implementation and earlier evaluation

We used AI for literature checks, implementation changes, test development, experiment orchestration and manuscript editing. We checked outputs and reported quantities against source code and saved result artifacts. The author is responsible for the final manuscript and submission. The system under test in the live-model experiments was Claude Sonnet 5.

The Python implementation contains provider, publisher, receipt and verifier components, an append-only log, Merkle proofs, auditor functions and adversarial test modes. It uses Ed25519 signatures [7] and canonical JSON. The preserved workshop baseline has 372 passing tests. The current combined suite passes 412 tests: 407 protocol tests, including negative artifact-binding and client-observation cases, plus five E12 replay tests. Original experiment records are retained separately from the new evaluations.

The reference implementation and public reproducibility artifacts are archived as software version 0.2.0 [1].

Experiments E1–E5 test the original receipt format, log, audit, inflation screen, and privacy behavior. The legacy v0.1 receipt measured 771 bytes as a payload and 1,007 bytes as a signed envelope; median issuance took 79 microseconds and signature verification 64 microseconds. These figures do not include the optional generation-event and arXiv artifact fields added for the post-cutoff experiment, so we report them as a baseline rather than as the cost of the final envelope. At the 299-probe budget against 1% suppression, full-stack detection was 0.98 over 50 trials versus 0.95 predicted. A reuse-cap screen detected $5 \times$ or greater inflation in the tested cell but allowed $2 \times$ inflation, which is an explicit limit rather than a success case.

E6 replaced the simulated model and corpus with 500 live Claude Sonnet 5 calls over 60 OpenAlex papers [27], using BM25 top-5 retrieval [29]. The run produced 792 receipts, all of which verified. Deterministic replay across 24 suppression settings tracked the predicted audit bound; 949 honest-path trials produced no false detection. The transport and transparency log ran in process with the same signature and proof checks. E6 supplied candidate papers in context, so it tested live receipt handling rather than post-cutoff discovery or model-memory attribution.

An archived intermediate study, E7, used 20 questions and 300 calls over a 60-paper DOI corpus. Target citations appeared in 76/100 parametric responses and 95/100 responses in each retrieval condition, with 190 receipts marked valid in the preserved records. The study lacked verified post-cutoff status and used byte-only response commitments. Identical-answer ambiguity and truncated abstract evidence informed the later event-bound, version-pinned E8 design.

6 Post-cutoff citation experiment (E8)

6.1 Design

This experiment tests the recording and verification of retrieval-backed citation events for papers first submitted after the model's documented training-data cutoff. Its output is a count of recorded events per versioned paper identifier. Claude Sonnet 5's model-specific documentation lists both its training-data cutoff and reliable knowledge cutoff as January 2026 [4]. The versioning documentation describes the model ID as a pinned snapshot and distinguishes fixed weights from serving infrastructure and safety behavior [5]. The AWS model card lists the same January 2026 knowledge cutoff [3].

We froze a corpus of 100 arXiv v1 papers first submitted between 1 and 27 July 2026: 20 targets and 80 distractors across cs.AI, cs.CL, cs.LG and cs.IR. All papers were identified uniformly by versioned arXiv identifiers, and 94 had no DOI. Receipts signed both the v1 PDF hash and the served-text hash. One question per target was written from a concrete main-text statement before model calls, checked by code and frozen by hash.

Each question was run five times in each of three conditions, for 300 calls in total.

No retrieval.

The model received only the question and output format. It had no tools or web access. Any source it named could not receive a CRP receipt because no authenticated fetch existed.

Natural retrieval.

BM25 searched the 100-paper local corpus and supplied the top five papers. The target was not forced into the result set.

Oracle retrieval.

The target and four high-ranked distractors were supplied, with the target rotated across positions 1–5.

The tested endpoint was configured as anthropic.claude-sonnet-5 through Amazon Bedrock and invoked as us.anthropic.claude-sonnet-5. Bedrock Converse returns no model field, so the claude-sonnet-5 echo was captured with one setup InvokeModel probe against the same model. No tool configuration or non-default sampling field was sent. Every response had a unique generation-event identifier. A receipt was issued only when the citation was visible in the answer, declared in the structured citation list, and matched an authenticated source in the supplied context.

6.2 Results

All 300 calls completed. Table 2 and Fig. 2 show the main contrast. The target appeared in 0 of 100 no-retrieval answers. It appeared in 95 of 100 natural-retrieval answers and 95 of 100 oracle answers. The paired difference between natural retrieval and no retrieval was +0.95, with a 95% bootstrap interval of [0.85, 1.00] over the 20 question–paper pairs. Response-level Wilson intervals describe the recorded calls; repeated calls to the same twenty questions do not constitute one hundred independent papers or a sample of ordinary user traffic.

Table 2. *Post-cutoff experiment results. Brackets are 95% Wilson intervals. Oracle counts are reported separately and are not part of the natural end-to-end count.*

	No retrieval	Natural	Oracle
Responses	100	100	100
Target visibly cited	0	95	95
Citation rate	0.000 [0.000,0.037]	0.950 [0.888,0.978]	0.950 [0.888,0.978]
Valid receipts	0	95	95
Out-of-context citations	–	0	0
Distractor citations	–	0	0
Automated gold-string match	0.000	0.900	0.900

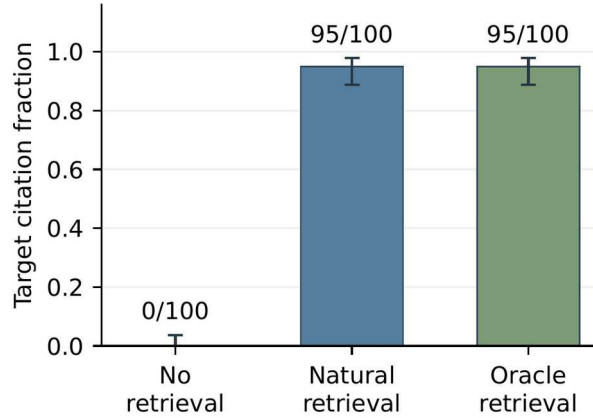


Fig. 2 Target citation rates in the post-cutoff experiment. Papers were first submitted six months after the model's documented training-data cutoff. Error bars are 95% Wilson intervals

Across the two retrieval conditions, 190 visible target citations were eligible for receipts. CRP issued and verified all 190. There were no mismatched, out-of-context or distractor receipts, and none in the no-retrieval condition. Natural retrieval produced 95 verified post-cutoff citation events across twenty target identifiers, with zero to five per paper. Oracle receipts served as an availability control and were counted separately. One no-retrieval answer to Q20 cited the non-corpus source arxiv:2306.08121v3 and received no receipt.

Both retrieval conditions had five citation failures, all for Q17, whose target was P17 (arxiv:2607.18712v1). The required figure-caption detail lay outside the supplied window, so the model returned insufficient evidence rather than a distractor citation. Q12's exact prespecified span was also absent, but another supplied passage contained the answer and all ten Q12 calls succeeded. Each retrieval condition cited the target in all ninety calls for the eighteen questions whose exact evidence span was present. The main citation result remains the prespecified, hash-frozen 95/100 per condition.

Across both retrieval conditions, Q06 accounted for ten automated gold-match misses: its gold value was 0.881, while every answer returned the equivalent 88.1%. Q17 accounted for the other ten misses. Treating those numeric forms as equivalent gives 95/100 answer matches per retrieval condition; the remaining five misses are the Q17 insufficient-evidence cases.

In the natural retrieval condition, every target paper ranked first. The paper-specific questions contained distinctive terms, so this condition tests the path from receipt to count after controlled retrieval.

The post-cutoff experiment also exercised the generation-event fix. Eleven groups of byte-identical responses covered 26 calls in the same epoch. With the event identifier in Equation 2, the audit returned exactly 190 receipt matches for 190 claims and produced zero attribution failures. A negative test also confirmed that a duplicate (*issuer, receipt_id*) is rejected.

7 Controlled adversary and client-observation study (E9)

E9 tests selective omission, audit recognition, receipt withholding and fabricated events in a controlled testbed. Its fixed synthetic corpus has 3,000 displayed responses, two publishers, ten DOI-bearing source fixtures, five topics and five contiguous ten-minute windows. Each response has one eligible visible citation. The target publisher has 300 events; the selected topic and time window each have 600. Assignments and sampling use fixed streams derived from seed 90210. Event identifiers are deterministic, while receipt identifiers, keys and salts are random. Aggregate results reproduce even though receipt bytes change.

Before middleware processing, a separate client fixture records the display bytes, event identifier, epoch, expected provider key and cited source. Retrieval eligibility requires signed mock fetches in the publisher’s log. At epoch close, the client verifies the manifest and inclusion under its configured log key, then pins the canonical manifest hash before challenge. This freezes one observed view, without enforcing log-wide uniqueness of the issuer/epoch root. The client fixture operates in process.

7.1 Selective omission and sampling budgets

In each attack, the provider omits exactly 10% of eligible citations in the chosen group before signing or sequencing. Other events follow the existing receipt path. The honest control omits none. Receipt issuance, salted commitments, Ed25519 signatures, epoch-tree proofs, signed manifests and transparency-log inclusion use the reference implementation. Every recorded source-event outcome is challenged and verified once. Resampling uses those verified finite-population outcomes.

We draw 200 fixed-seed subsets without replacement for each strategy and budget. Pooled sampling draws from all 3,000 eligible citations; conditioned sampling draws only from the selected group. The 200 resampling trials share one fixed finite population. With M omissions among N eligible observations, the exact finite-population detection probability for budget n is

$$P_{finite}(detect) = 1 - \frac{\binom{N-M}{n}}{\binom{N}{n}},$$

with probability one when $n > N - M$. This hypergeometric reference differs from the independent-suppression model in Theorem 1.

Table 3. E9 detection at a budget of 50 eligible citations. Detection columns report observed counts across 200 resampled subsets. The last column gives exact finite-population detection probabilities, pooled/conditioned, in percent. These probabilities refer only to the fixed synthetic populations.

Attack	Omissions/eligible	Pooled	Conditioned	Exact probability (%)
Publisher	30/300	70/200	200/200	39.75/99.69
Topic	60/600	138/200	199/200	63.89/99.59
Time window	60/600	127/200	198/200	63.89/99.59

Table 3 and Fig. 3 compare pooled and group-conditioned sampling at a budget of 50 eligible citations. At budget 299, publisher omission was detected in 192/200 pooled subsets and 200/200 conditioned subsets. Topic and time-window attacks were detected in all 200 subsets under both strategies. The honest control produced zero failures and zero detections for every strategy and budget. These detections describe the fixed population and sampled subsets.

Pooled budgets count citations across publishers. Samples of fifty contained a mean of 4.90 target-publisher citations, with a range of zero to thirteen. Samples of 299 contained a mean of 30.07, with a range of eighteen to forty-eight. Conditioned samples contained exactly fifty or 299 target citations. The separate 299-citation design result applies to eligible citations from the named population under independent suppression and indistinguishable probes. A publisher-specific interpretation requires that full eligible publisher quota.

7.2 Client binding and audit recognition

The verifier checks provider issuer, event identifier, epoch, source and use class, plus the manifest's issuer, epoch and pinned hash. Pinning and challenge checks also require the configured log identity, with a key distinct from the provider's. A receipt for another event can remain valid for identical bytes, yet client binding rejects its replay. A newly signed same-epoch root or provider-signed tree head can pass basic cryptographic checks but fails the strict profile. Negative tests reject altered bytes and mismatched identifiers, sources, issuers, log keys, epochs and anchors. Missing identifiers, anchors or trusted log configuration are unauditable, outside the detection denominator; eligibility and auditable coverage are reported separately. Duplicate events and source-event units are rejected. The E9 module adds twenty-nine tests.

In the recognition case, 1,000 organic events include 100 target-publisher citations, alongside 100 additional target-publisher audit events. The adversary recognizes the latter and preserves every audit receipt while omitting ten organic target receipts. All 100 recognized probes pass. Auditing the organic target records exposes ten failures. This in-process attack tests the consequence of violating probe indistinguishability.

7.3 Delivery and inflation limits

The delivery case commits 1,000 receipts, including 100 for the target publisher, then withholds ten target receipts. The remaining 90 verify, and all 100 target event challenges succeed. Global sequence gaps total 910, comprising 900 legitimate other-publisher receipts and ten withheld receipts. Recipient reconciliation against externally verified target receipt identifiers exposes ten missing deliveries in that observed set. With no external observations, the reconciliation input is empty. Publisher-wide completeness requires the proposed commitment extension in Section 3.5.

E9 produced twenty valid target receipts: ten for actual target events and ten for distinct fabricated events. The three-uses-per-fetch plausibility screen flagged no DOI/epoch pair. Repeating delivery raised the naive count to forty. Receipt-identifier deduplication restored twenty, but the ten fabricated events remained. Signatures and commitments bind provider declarations to records. Independent observation, delivery completeness and authentic user demand require further mechanisms. E9 supplies controlled implementation measurements.

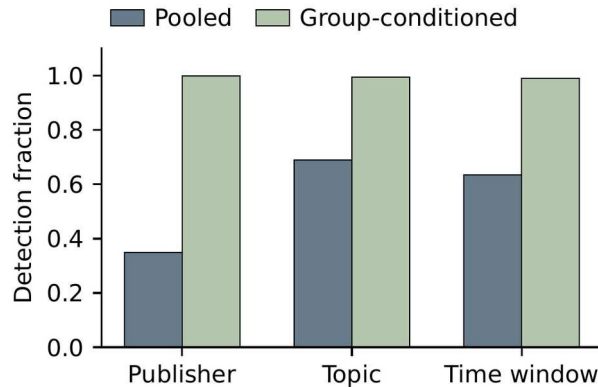


Fig. 3 Detection in 200 fixed-population resamples of 50 eligible citations. Group-conditioned sampling refers to the named attacked group; these observations do not estimate deployed traffic behavior

8 Current-format overhead (E10)

E10 measures the current receipt profile with the generation-event identifier, versioned arXiv source, served-text hash and PDF hash enabled. The fixture uses an 8,192-byte source body, a 3,909-byte synthetic PDF and a

1,024-byte response. Source and key setup are outside the timed regions. Sizes refer to canonical UTF-8 JSON before compression, HTTP framing or filesystem overhead. Identifier and URL lengths affect those sizes.

The canonical payload is 935 bytes and the signed envelope is 1,171 bytes. A complete one-receipt delivery package including proof material and signed tree head is 1,969 bytes. Disabling only the optional event and PDF fields in the same fixture reduces the payload by 146 bytes. These are current-fixture measurements, not an attribution of all differences from the earlier 771-byte payload to the schema alone.

Table 4 reports component timings. Each microbenchmark uses 200 warmup calls and 2,000 individually timed samples. The host is an Apple M4 with ten logical CPUs and 24 GiB of memory, running CPython 3.14.7 on macOS 26.6.2, PyNaCl 1.6.2 and rfc8785 0.1.4. The process is unpinning and background activity is uncontrolled. We report medians and nearest-rank 95th percentiles. Components are timed separately, so their medians should not be added to estimate full-path latency.

Table 4. *Current-profile component times in microseconds. Signing and signature verification use a precanonicalized payload; full package verification additionally checks the signed manifest, log and epoch inclusion, license and artifact hashes.*

Operation	Median (μ s)	p95 (μ s)
Canonicalize payload	23.46	23.83
Sign canonical payload	149.88	164.04
Verify signature	463.83	533.83
Issue current receipt	197.00	400.42
Verify full one-receipt package	1429.62	1545.12

Table 5 reports the full local path: event UUID creation, a signed mock fetch, queueing, receipt signing, epoch-tree construction, manifest signing, transparency-log append, individual inclusion proofs, in-process publisher delivery and full package verification. Batches of one, ten and one hundred receipts use twenty-five repetitions after two warmups. The 1,000-receipt batch uses eleven repetitions after two warmups. Every package passed verification.

Table 5. *Measured local full-path time per epoch batch and corresponding receipt throughput. No model generation, network transport, durable storage or concurrent clients are included.*

Receipts	Median (ms)	p95 (ms)	Receipts/s
1	2.709	2.828	369.2
10	15.247	18.548	655.9
100	147.918	162.869	676.0
1000	2536.426	2587.145	394.3

The prototype repeatedly builds sibling paths for individual inclusion proofs, a cost that grows much faster than root construction alone. A 1,000-receipt batch takes 2,536.426 ms at the median. Building a 100,000-leaf root takes 118.720 ms median and 131.730 ms p95. These root timings do not measure the issuing and delivery of 100,000 verified receipts. A cached tree and a persistent proof service remain engineering work for a scalable deployment.

Storage accounting depends on scope. The 188-byte epoch-manifest leaf excludes the 311-byte serialized signed tree head and receipt storage. Each serialized receipt-identity registry pair occupies 108 bytes in the fixture before Python object overhead. Manifest leaves alone therefore do not describe total provider storage. At 1,000

epoch leaves, local append, root update, tree-head signing and the latest inclusion proof take 1.845 ms median, about 542 serial appends per second. This in-memory measurement excludes disk persistence and replication.

9 Fresh context-intervention study (E11)

E11 is a separate study dated 1 October 2026. It uses the same frozen twenty question-paper pairs and hundred-paper corpus as E8. Each question was run once in each of three prespecified context conditions: the top-five BM25 sources, the top-five ranked sources after excluding the target and no source blocks. All conditions used the same instruction to answer from supplied evidence. The empty-context condition is an abstention and receipt-eligibility control, distinct from E8's instruction to answer from existing model knowledge.

The call plan, source selections and every prompt hash were saved before calls. The requested inference profile was `us.anthropic.claude-sonnet-5` in `us-west-2` through Bedrock Converse, with maximum output 4,000 tokens, no sampling override, no thinking override and no toolConfig. Three workers submitted independent calls, one per question and condition. The run retained raw responses, HTTP request identifiers, token usage and parsing outcomes. Converse does not echo the resolved model identifier. A separate `InvokeModel` setup probe returned `claude-sonnet-5`; this provider-reported identity is recorded separately from the requested inference profile and the sixty experimental calls.

Table 6. E11: one fresh response per question per context condition. A receipt is eligible only for a visible, declared source in authenticated context. Evidence insufficiency is the model-reported flag, not an independent quality assessment.

Context	Calls	Target cited	Valid receipts	Insufficient
Available	20	20	20	1
Target excluded	20	0	0	20
Empty context	20	0	0	20

Table 6 reports the three context conditions. All sixty calls completed and parsed. With the target available, twenty responses visibly cited the target and twenty eligible receipts verified. Neither target-excluded nor empty-context responses cited a target or generated an eligible receipt. No out-of-context visible citation was observed. One available-context answer cited the target while marking evidence insufficient. Its receipt counts that observed citation event.

Receipt issuance and authenticated mock fetches ran locally after the API responses were saved. The replay used the parsed answer field as the deterministic client display artifact. Its UTF-8 bytes and hash were stored separately from raw model JSON, and the commitment was opened against those bytes. The signed, logged epoch manifest was pinned before challenges. Source identifier, expected issuer, event identity, prior manifest, license, fetch reference, served-text hash and PDF hash were checked against the client record, frozen corpus and publisher record. Every eligible event had exactly one matching verified receipt, with zero event-binding or coverage failures. E8 instead committed to the complete model JSON response.

The available-context result differs from E8's 95/100 repeated responses. E11 ran later and used twenty calls in that condition as a controlled intervention on the same questions. The studies describe their own recorded calls. Excluding the target deliberately tests source absence.

10 Client-witness suppression replay (E12)

E12 replays a local provider using saved live-model answers with deliberate receipt suppression. It reuses the twenty Available responses from E11, with no additional model calls. Each response has one eligible visible

target citation. The plan fixed two conditions before replay: issue all eligible receipts, or suppress issuance for the first two lexicographically ordered call identifiers, E11-Q01-AVAILABLE and E11-Q02-AVAILABLE. The input hashes, omission mask and script hash were frozen before execution.

An in-process client fixture records each saved answer's UTF-8 bytes, source markers and existing generation UUID outside receipt construction. Every condition uses real signatures for 100 local mock fetches, then constructs and logs an epoch manifest. All twenty client observations pin the signed manifest and configured log identity before the first challenge. The audit challenges every eligible source-event pair, so this is a census of a fixed twenty-event fixture rather than a sampling-power experiment. It verifies source and artifact bindings, receipt signatures, epoch and log inclusion, and event-to-observation agreement.

Table 7. *E12 census of saved E11 Available responses under local receipt issuance and a predeclared two-event suppression mask. A detected gap is a missing or invalid strict-profile citation opening, not a global sequence gap.*

Condition	Eligible	Omitted	Valid	Detected gaps	False alarms
Baseline	20	0	20	0	0
Suppress Q01/Q02	20	2	18	2	0

Table 7 summarizes the replay results. The baseline returned twenty valid citation openings with no gaps or false alarms. Suppression left eighteen valid openings and exposed both omitted events, with no false alarms among the eighteen retained citations. All returned proofs and delivery packages verified. Both conditions had zero unauditible citations, zero invalid returned proofs and zero global sequence gaps. The client witness detected unissued receipts despite the absence of sequence gaps.

Authenticated fetches, receipt issuance, logging and challenges ran in process after the original API calls. The twenty saved answers and fixed two-event omission mask define the replay population. The one answer that flagged insufficient evidence still had an observed citation and remained eligible for audit. Section 13 states the deployment and population boundaries.

11 What the count means

Let $R_p(T)$ be the accepted records in epoch window T under a declared verification profile P . For source s , define a set of unique provider-generation pairs

$$E_p(s, T) = \{(issuer(\rho), g(\rho)) : \rho \in R_p(T), \\ src(\rho) = s, u(\rho) = u_{vis}\}, \quad (4)$$

where u_{vis} =visible_citation and a missing generation identifier is excluded from this profile. The deduplicated source-event count is

$$C_p(s, T) = |E_p(s, T)|. \quad (5)$$

Repeated delivery or a second receipt identifier for the same $(issuer, g, s)$ does not increment this count. A distinct fabricated event can still increment a provider-declaration profile's count. A profile that requires independent client observation counts only records with that additional evidence. We write $C(s, T)$ below with the profile understood.

For the natural-retrieval condition in E8, Fig. 4 traces the 100 responses through the receipt pipeline. Fig. 5 gives the resulting 95 verified receipts by target paper. The zero count for P17 corresponds to question Q17; each of its five calls lacked the required figure-caption evidence. Oracle receipts remain separate from this count. The

experiment calculates $C(s, T)$ inside one controlled run. A *verified post-cutoff count* adds a temporal label when the source's first public date is later than the tested model's documented training-data cutoff and the receipt is tied to authenticated retrieval. The label states that documented time relation and observed retrieval path.

A public Machine Citation Index would sit above CRP and aggregate receipts across providers, dates, source versions, authors and institutions. It would need rules for linking arXiv preprints, accepted manuscripts and journal DOIs to a scholarly work. It would also need policies for deduplicating repeated event records, maintaining provider identities, monitoring public logs and separating natural traffic from controlled audit or oracle traffic. Equation 5 is an event count, not a final research-impact score.

Table 8 separates the evidence supplied by the protocol from claims that need other methods. A signed receipt records a provider's declaration of a citation event. Explaining why the model selected that source requires other evidence.

Table 8. *Meaning of the evidence. Each row is narrower than the next claim a reader might be tempted to make.*

Evidence	What it supports
Authenticated fetch	The recorded identity obtained source bytes under the recorded terms; the controlled transport uses a signed mock request.
Signed receipt	The provider declared a visible citation for that source and generation event.
Log inclusion	The receipt existed in the committed epoch set.
Successful audit opening	The recorded client event and source have a matching committed receipt when strict observation binding is checked.
Receipt aggregation	A count of verified records for a source identifier and reporting window, conditional on deduplication and traffic selection.
Not supplied by CRP	Semantic support, causal dependence, training-data influence, or global citation frequency.

The count aggregates signed declarations of machine-mediated citation presentation, with sampled client checks where available. Bibliographic citations by later papers are a separate measure. Publisher-wide delivery reconciliation could use the subroot design in Section 3.5. Training-data attribution would require provider-side corpus records and a defensible method for connecting a generated claim to one paper rather than to copies, summaries or related sources.

For historical comparison, Table 9 retains the earlier v0.1 fixture costs. The current-format measurements use a different schema and fixture.

12 Digital-library integration and governance

A repository could present CRP records alongside COUNTER reports and attention indicators, with each measure retaining its unit, provider coverage and observation window [2, 12, 13]. DOI links could connect events to OpenCitations metadata without converting repeated answer displays into new bibliographic links [16, 24]. Mappings to PROV-DM and SCITT remain integration work [8, 22]; neither mapping has been implemented in this study.

12.1 Artifacts, versions and work-level aggregation

An ingestion service would keep the receipt's canonical scheme, versioned identifier, artifact digest, served-content digest and reporting epoch. A DOI identifies an object under a publisher's version policy, while a content digest identifies bytes. An arXiv version identifies a particular preprint revision. These identifiers can describe

related versions of a scholarly work with different bytes. CRP preserves artifact identity but leaves work-level relationships to a separate catalog layer.

A library can represent a PDF as a source entity, extracted text as a derived entity and generation as an activity that uses that text. Provider identity records the party responsible for the declaration. A work-authority table can link a preprint, accepted manuscript and version of record. The evidence and rule used for the merge must also be retained. A matching title alone is insufficient. Version-specific audit counts should remain accessible after the work-level view is produced.

Cached retrieval requires an authenticated original fetch reference and a specified reuse policy. A single fetch can support several later generation events. Subsequent receipts must identify the source version and cached artifact actually used rather than claim that each citation triggered another publisher request. Hash agreement verifies bytes against a known reference. Publisher authority also requires authenticated delegation. In the current evaluation, these fetches occur at a local signed mirror endpoint. Proxy delegation, cache freshness and publisher-authorized extraction require operational agreements.

12.2 Counting and traffic separation

The ingestion layer first rejects repeated receipt identifiers and then counts at most one visible citation for each $(issuer, g, s)$ tuple. Repeating the same citation marker inside one answer is one source-event pair. A transport retry that does not produce a new user-visible generation reuses the event identifier; a new delivered generation receives a new identifier. A display or client protocol must make this distinction, especially when streaming or partial responses are possible.

Audit, test, oracle and naturally occurring service traffic need separate accounting. The experiments' receipts report protocol behavior within test traffic. An auditor can retain operational audit labels after observations rather than expose them to the provider and make probes identifiable. The service needs to document and validate its traffic-classification mechanism.

Expandable source panels, footnotes and bibliography sections qualify only under a declared visibility rule. The evaluated profile uses citation markers in the displayed answer field. Sources present only in retrieval context, hidden grounding metadata or inter-agent messages are excluded. A broader presentation profile could define those events separately, but it would need its own client observation and audit policy. A handoff between agents must preserve authenticated source custody if a later agent cites a summary rather than the original document.

Deduplication removes repeated records, not invented events. A provider can mint distinct event identifiers, sign fabricated declarations and keep the count below a license reuse cap. E9 illustrates this residual threat. An index that attaches monetary or bibliometric value to the count needs independent observation, abuse detection or contractual auditing beyond receipt cryptography. We therefore recommend reporting the count with its profile, provenance, traffic class and audit window rather than presenting it as an unconditional metric.

12.3 Privacy and retention

Receipts omit user IDs and query text, but source identities and epochs can reveal interest in a rare topic. A stable event identifier can link sources cited in the same response. This metadata linkage persists even when a salted commitment hides response bytes. Public logs can contain epoch roots and signed manifests, with receipt payloads and sampled openings access-controlled under publisher and auditor policies.

Per-receipt salts derived from an epoch key allow a challenge to open one commitment without revealing the entire epoch key. Client records and response openings may contain sensitive queries or generated text and need separate access and deletion rules. Source artifacts may also be restricted by their licenses. The reproducibility package contains published scholarly sources and synthetic adversarial queries rather than private user traffic.

The earlier batching experiment explored the tradeoff between the ambiguity set and delivery delay. The ambiguity set counts distinct fetch candidates, with a cached fetch counted only once. A batching threshold should therefore use distinct reference observations and specify an expiry policy for low-volume traffic. Differential privacy, resistance to auxiliary information and operational usefulness still require a deployment study.

13 Limitations

The omission guarantee remains conditional on independent suppression and probes drawn from the relevant traffic population without provider recognition. Publisher-conditioned sampling improves sensitivity only when eligible citations can be obtained. A provider can reduce the visible citation rate, ground silently, identify auditor accounts or change behavior by topic and time. The receipt protocol cannot reveal every form of source use. Signed event bindings do not establish semantic citation support, causal dependence or training-data influence.

The strict observation profile binds evidence to a client's recorded event and bytes. It does not prove an ecosystem-wide census of real users. Sampled receipt-ID reconciliation detects withholding for events already observed and successfully opened, but no publisher-scoped total commitment is implemented. Correct signatures can coexist with fabricated events, as the low-inflation experiment demonstrates. The client delivery design assumes an honest recording component and does not provide remote attestation or a census of users. Provider-controlled identifiers cannot authenticate user demand; different signing keys cannot establish organizational independence. The publisher-subroot design is an unimplemented extension for delivery of committed sets, with the partition and privacy requirements in Section 3.5. Logs also need external monitoring, witness agreement and gossip to expose equivocation between clients; a single in-process log view cannot demonstrate operational accountability.

The live studies use one model, a local corpus and highly specific questions. E8's every-target-first retrieval and E11's deliberate context exclusion limit conclusions to controlled receipt handling and citation behavior. The model's documented cutoff is a temporal label, not proof of absence from every provider system or evidence of training influence. Repeated-question response intervals are descriptive, while E12 is a census of twenty saved answers under a fixed local suppression mask.

Performance measurements exclude real HTTP/TLS traffic, model generation, disk durability, replication and concurrent consumers. Individual proof construction is a scaling bottleneck. Artifact hashing depends on trusted reference bytes and a documented extraction process. Receipt metadata can disclose topic interest and link source sets, and deduplication cannot authenticate independent user demand. No provider, publisher or digital-library deployment is claimed.

14 Conclusion

CRP connects an authenticated scholarly source fetch to a provider-signed citation declaration, a committed epoch set and a conditional sampled audit. The journal evaluation shows why the sampling population matters: targeted omission can be difficult to detect in pooled traffic, recognized probes can all pass while organic events are omitted, and valid log proofs can coexist with withheld delivery or fabricated events. Client observation and receipt-ID reconciliation improve checks for observed samples.

Separate live-model studies found that eligible citations can produce verifiable receipts in the controlled pipeline. Current-format measurements cover event and artifact fields and detail the full path of proof generation and its costs. The result is a versioned provenance record with stated audit and traffic boundaries. A public citation index also requires independent observation, recipient-scoped completeness mechanisms, privacy governance and evaluated deployment.

Statements and Declarations

Funding

This research was self-funded by the author.

Competing interests

The author personally owns a filed patent application related to the Citation Receipt Protocol. The application is disclosed as a potential financial competing interest.

Data and code availability

The reference implementation, analysis scripts, frozen plans, result summaries and sanitized evidence exports are publicly available as software version 0.2.0 [1]. The version-specific Zenodo archive is <https://doi.org/10.5281/zenodo.23092601>; the GitHub repository is <https://github.com/Abhinav0905/citation-receipt-protocol-research>. The cited GitHub release is <https://github.com/Abhinav0905/citation-receipt-protocol-research/releases/tag/v0.2.0>, at commit 0acd710b840fd0442e8f2f3f105c1c6e25418dcb. The public package omits bulk third-party PDFs and extracted text, full source abstracts and source-bearing request prompts. The frozen public version 0.2.0 predates the additional E12 saved-output suppression replay. The complete private peer-review package, including original saved request/output records, frozen source inputs and E12 records, is available from the author on request. Versioned source identifiers, hashes and reconstruction instructions document the excluded inputs. Third-party source content remains subject to its applicable terms.

Ethics and consent

The evaluated traffic consists of synthetic queries and controlled API calls over scholarly sources. No human participants or private user traffic were recruited or observed.

Author contributions and AI assistance

The author developed the protocol and directed the study. Section 5 records the use of AI assistance and the checks on its outputs. The author is responsible for the final manuscript and submission.

Prior workshop version

A shorter version titled "Citation Receipts: A Protocol for Verifiable AI Citation Counts" was accepted for a poster at the non-archival AI4MetaScience workshop at NeurIPS 2026. Its permanent OpenReview record is <https://openreview.net/forum?id=rThrMcXwsa>. This journal version adds selective-adversary evaluation, strict client observation, sampled delivery reconciliation, current-profile benchmarks, a separate context-intervention study and the saved-output suppression replay.

A Formal model and proof details

Definition 1 (Visible citation event).

A visible citation event is (g, r, s) , where g is a unique generation-event identifier, r is the rendered response bytes, and s is a source visibly cited in r and present in the authenticated retrieval context.

Proposition 1 (Third-party unforgeability).

Assuming Ed25519 is existentially unforgeable under chosen-message attack, a publisher or auditor cannot create a receipt that verifies under the provider's receipt key except with negligible probability.

Proposition 2 (Append-only epoch set).

Assuming collision resistance and Certificate Transparency log accountability, once an epoch manifest is publicly committed, the provider cannot add, remove, or alter a receipt without either changing the committed root or presenting inconsistent signed tree heads.

Proof of Theorem 1.

For each eligible probed citation i , let $X_i = 1$ when the provider suppresses its receipt. By A1, the provider cannot condition on probe status, and the theorem assumes X_i are independent Bernoulli(q) variables. A valid challenge for all probes requires $X_i = 0$ for every i . With the pre-challenge manifest pinned, a suppressed leaf cannot be supplied by appending a different epoch manifest. Thus

$$\Pr[\text{all probes pass}] \leq \prod_{i=1}^n \Pr[X_i = 0] = (1 - q)^n.$$

Equality requires valid answers for every nonsuppressed event. A receipt absent from the pinned epoch commitment cannot be added without changing its Merkle root. A second signed epoch root may be append-only logged, but it does not answer this pinned challenge. The signature, hash, and log assumptions prevent an outside party from supplying a different valid opening.

One-sided interval.

With zero failures in n probes, the one-sided exact upper confidence limit is $q_U = 1 - \alpha^{1/n}$. Solving $q_U \leq q_0$ yields Corollary 1. For varying independent rates q_i , the evasion probability is $\prod_i (1 - q_i) \leq (1 - \bar{q})^n$ by the arithmetic–geometric mean inequality.

B Earlier experiment details

Table 9. Legacy v0.1 cost measurements. The final protocol adds optional event and arXiv fields; these rows are retained as a backward-compatible baseline.

Quantity	Measured value
Receipt payload / signed envelope	771 B / 1,007 B
Issue one receipt	79 μ s
Verify one signature	64 μ s
Merkle root over 10^5 receipts	426 ms
Transparency-log growth	188 B/epoch

E3 ran six full-stack audit cells and a Monte Carlo grid. At $(q, n) = (0.01, 299)$, the observed detection rate was 0.98 over 50 trials against 0.95 predicted. At $(0.05, 59)$ it was 0.967 against 0.952, and at $(0.10, 29)$ it was 0.947 against 0.953. Every tested point was within two binomial standard errors of the closed form.

E4 generated 200 honest responses over 60 identifiers and then inflated the most-cited identifier with no matching fetches. A reuse cap of 3 flagged $5 \times$, $10 \times$, and $50 \times$ inflation with no false flags across 59 honest identifiers. It did not flag $2 \times$ inflation because the forged count remained below the configured cap. E5 measured timing linkage and batched delivery. Privacy from batching requires at least k distinct fetch references in the release window; k receipts from one cached fetch do not provide a k -sized ambiguity set.

C Post-cutoff protocol path and paper-level counts

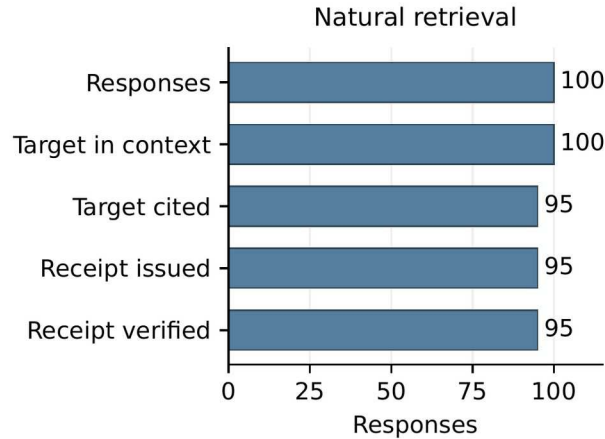


Fig. 4 Natural-retrieval path in the post-cutoff experiment. Each target citation that reached the visible answer produced one valid receipt

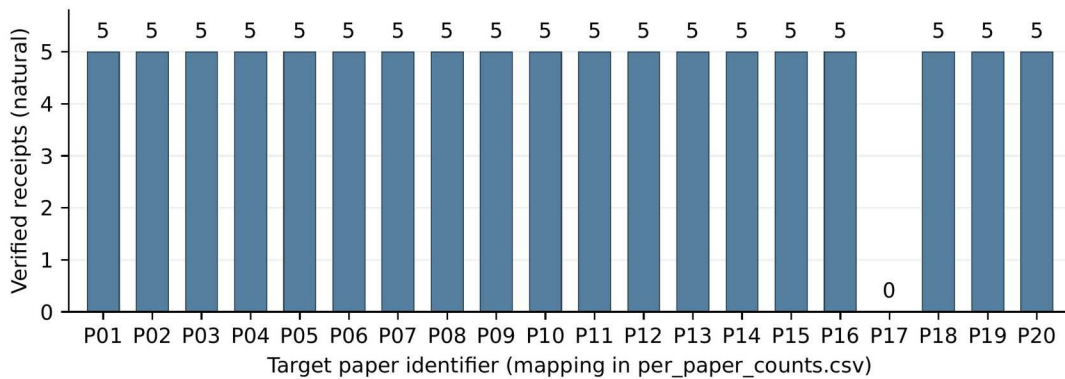


Fig. 5 Verified natural-retrieval receipts by target paper. P17 is the target of Q17 and received zero because all five calls lacked the needed figure-caption evidence in the supplied context

The corpus and question set were frozen under SHA-256 digests before the full run; the exact digests are included in the accompanying experiment artifacts. The plan initially supplied one best chunk per retrieved paper. Before the smoke test, a reachability check found the exact evidence in only 11 of 20 target contexts, so the frozen full-run rule became the abstract plus a BM25-centered, contiguous window capped at 6,000 words for every target and distractor. The response-format example `arxiv:2607.12345v1` appeared in all three conditions, including no retrieval; startup checks confirmed that it was not a corpus identifier.

The model was called through `bedrock-runtime.converse` in `us-west-2` with `max_tokens=4000`. No temperature, top-p, top-k, stop sequence, tool configuration, or web access was supplied. The experiment seed was 20260820. All 300 calls completed with no structured-output parse failure.

D Implementation notes

We canonicalize the receipt payload under RFC 8785 and sign it with Ed25519. The log uses RFC 6962 leaf and node hashing and supports inclusion and consistency proofs. We added versioned arXiv identities, separate PDF and served-content hashes, random generation-event identifiers, event-bound response commitments and a registry that rejects duplicate receipt identifiers across epochs. The event field is optional for backward compatibility. When it is absent, the serialized v0.1 receipt and its commitment remain byte-for-byte unchanged.

The evaluated fetch authentication and transparency log operate in process with real signatures and proof checks. HTTP transport and hosted-log operation remain deployment work, as described in Section 13.

E Reproducibility and analysis boundaries

The private peer-review research package contains the reference implementation, unit tests, fixed-seed E9 runner, E10 benchmark, frozen E11 plan, raw API outputs, response records and machine-readable summaries. Historical E6 and E8 records remain separate. The E12 extension includes its frozen replay plan, suppression mask, input hashes, client anchors, returned proofs and machine-readable outcomes. The code and measured results were prepared in an isolated copy of the workshop project. The package includes a file hash manifest and excludes credentials, virtual environments and local account configuration.

E9 fixed-seed resampling evaluates a finite population of actual cryptographic challenge outcomes and uses the finite-population sampling model for its hypergeometric probabilities. E10 reports single-host measurements rather than a model of a deployed service. E11's sixty API results were saved before receipt reprocessing, and the analysis configuration explicitly specifies the rendered answer artifact. Reprocessing saved outputs runs local protocol checks with zero model calls.

The source PDF hash is verified against the publisher's reference only when the receipt claims that field. The strict client profile requires an event identifier, expected issuer, configured log signer and signed/logged epoch manifest pinned before the challenge. The anchor must precede the challenge reply. Legacy compatibility tests verify older serialized forms under their own schema guarantees. The test count describes the executed suite; statistical sample sizes are reported by experiment.

References

- [1] Abhinav K (2026) Citation receipt protocol: reproducibility artifacts, v0.2.0. Zenodo, <https://doi.org/10.5281/zenodo.23092601>, URL <https://zenodo.org/records/23092601>, software, version 0.2.0
- [2] Altmetric (2026) How is the Altmetric Attention Score calculated? Altmetric Help Center, URL <https://help.altmetric.com/en/articles/9800513>, edited 22 September 2026. Accessed 1 October 2026
- [3] Amazon Web Services (2026) Claude Sonnet 5. Amazon Bedrock User Guide, URL <https://docs.aws.amazon.com/bedrock/latest/userguide/model-card-anthropic-claude-sonnet-5.html>, accessed 1 October 2026
- [4] Anthropic (2026a) Claude Sonnet 5: Specifications. Claude Platform documentation, URL <https://platform.claude.com/docs/en/models/sonnet-5/overview>, January 2026 training-data and reliable-knowledge cutoffs. Accessed 1 October 2026
- [5] Anthropic (2026b) Model IDs and versioning. Claude Platform documentation, URL <https://platform.claude.com/docs/en/about-claude/models/model-ids-and-versions>, accessed 1 October 2026
- [6] Baskaran A, Pherwani N, Krishnan R (2026) Aegon: Auditable AI content access with ledger-bound tokens and hardware-attested mobile receipts. arXiv preprint arXiv:2604.06693, <https://doi.org/10.48550/arXiv.2604.06693>, URL <https://arxiv.org/abs/2604.06693>, protocol design white paper
- [7] Bernstein DJ, Duif N, Lange T, et al (2012) High-speed high-security signatures. Journal of Cryptographic Engineering 2(2):77–89. <https://doi.org/10.1007/s13389-012-0027-1>, URL <https://link.springer.com/article/10.1007/s13389-012-0027-1>
- [8] Birkholz H, Delignat-Lavaud A, Fournet C, et al (2026) An architecture for trustworthy and transparent digital supply chains. RFC 9943, RFC Editor, <https://doi.org/10.17487/RFC9943>, URL <https://www.rfc-editor.org/info/rfc9943/>, proposed Standard

- [9] Bohnet B, Tran VQ, Verga P, et al (2022) Attributed question answering: Evaluation and modeling for attributed large language models. arXiv preprint arXiv:2212.08037, <https://doi.org/10.48550/arXiv.2212.08037>, URL <https://arxiv.org/abs/2212.08037>
- [10] Cazzaro M, Silvello G (2026) Large language models as digital libraries: a multi-benchmark and multi-model study. *International Journal on Digital Libraries* 27:13. <https://doi.org/10.1007/s00799-026-00448-w>, URL <https://link.springer.com/article/10.1007/s00799-026-00448-w>, article 13. Published 8 September 2026
- [11] Clopper CJ, Pearson ES (1934) The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika* 26(4):404–413. <https://doi.org/10.1093/biomet/26.4.404>, URL <https://academic.oup.com/biomet/article-abstract/26/4/404/291538>
- [12] COUNTER Metrics (2025) COUNTER Code of Practice Release 5.1.0.1. COUNTER Metrics, URL <https://cop5.cOUNTERmetrics.org/en/5.1.0.1/>, published 14 January 2025. Accessed 1 October 2026
- [13] Elsevier (n.d.) Measuring impact with PlumX Metrics in Digital Commons. Digital Commons Help Center, URL <https://digitalcommons.elsevier.com/measuring-impact-with-plumx-metrics-in-digital-commons>, online documentation. Accessed 1 October 2026
- [14] Figuera J (2026) Notarized agents: Receiver-attested confidential receipts for AI agent actions. arXiv preprint arXiv:2606.04193, <https://doi.org/10.48550/arXiv.2606.04193>, URL <https://arxiv.org/abs/2606.04193>
- [15] Gao T, Yen H, Yu J, et al (2023) Enabling large language models to generate text with citations. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Singapore, pp 6465–6488, <https://doi.org/10.18653/v1/2023.emnlp-main.398>, URL <https://aclanthology.org/2023.emnlp-main.398/>
- [16] Heibi I, Peroni S, Shotton D (2019) Software review: COCI, the OpenCitations Index of Crossref open DOI-to-DOI citations. *Scientometrics* 121(2):1213–1228. <https://doi.org/10.1007/s11192-019-03217-6>, URL <https://link.springer.com/article/10.1007/s11192-019-03217-6>
- [17] Laurie B, Langley A, Kasper E (2013) Certificate transparency. RFC 6962, RFC Editor, <https://doi.org/10.17487/RFC6962>, URL <https://www.rfc-editor.org/rfc/rfc6962>
- [18] Laurie B, Messeri E, Stradling R (2021) Certificate transparency version 2.0. RFC 9162, RFC Editor, <https://doi.org/10.17487/RFC9162>, URL <https://www.rfc-editor.org/rfc/rfc9162>
- [19] Liang W, Zhang Y, Wu Z, et al (2024) Mapping the increasing use of LLMs in scientific papers. arXiv preprint arXiv:2404.01268, <https://doi.org/10.48550/arXiv.2404.01268>, URL <https://arxiv.org/abs/2404.01268>
- [20] Lu C, Lu C, Lange RT, et al (2024) The AI scientist: Towards fully automated open-ended scientific discovery. arXiv preprint arXiv:2408.06292, <https://doi.org/10.48550/arXiv.2408.06292>, URL <https://arxiv.org/abs/2408.06292>
- [21] Meunier T, Major S (2026) HTTP message signatures for automated traffic. Internet-Draft draft-ietf-webbotauth-httpsig-protocol-00, Internet Engineering Task Force, URL <https://datatracker.ietf.org/doc/html/draft-ietf-webbotauth-httpsig-protocol-00>, dated 1 September 2026. Accessed 1 October 2026
- [22] Moreau L, Missier P (eds) (2013) PROV-DM: The PROV data model. W3C Recommendation, World Wide Web Consortium, 30 April 2013. URL <https://www.w3.org/TR/2013/REC-prov-dm-20130430/>
- [23] Newman Z, Meyers JS, Torres-Arias S (2022) Sigstore: Software signing for everybody. In: *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. Association for Computing Machinery, pp 2353–2367, <https://doi.org/10.1145/3548606.3560596>
- [24] Peroni S, Shotton D (2020) OpenCitations, an infrastructure organization for open scholarship. *Quantitative Science Studies* 1(1):428–444. https://doi.org/10.1162/qss_a_00023, URL https://doi.org/10.1162/qss_a_00023
- [25] Priem J, Hemminger BH (2010) Scientometrics 2.0: New metrics of scholarly impact on the social Web. *First Monday* 15(7). <https://doi.org/10.5210/fm.v15i7.2874>, URL <https://firstmonday.org/ojs/index.php/fm/article/view/2874>

- [26] Priem J, Taraborelli D, Groth P, et al (2010) Altmetrics: A manifesto. Manifesto, archived on Zenodo, <https://doi.org/10.5281/zenodo.12684249>, URL <https://zenodo.org/records/12684249>, original publication 26 October 2010; archived version 1.01. Repository record created 8 July 2024
- [27] Priem J, Piwowar H, Orr R (2022) OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts. arXiv preprint arXiv:2205.01833, <https://doi.org/10.48550/arXiv.2205.01833>, URL <https://arxiv.org/abs/2205.01833>
- [28] Rittman M, Cousijn H (2026) Strengthening support for data citations and saying goodbye to Event Data. Crossref, <https://doi.org/10.64000/rzbn5-wjy58>, URL <https://www.crossref.org/blog/strengthening-support-for-data-citations-and-saying-goodbye-to-event-data/>, published 24 March 2026; edited 15 June 2026. Event Data API sunset 23 April 2026. Accessed 1 October 2026
- [29] Robertson S, Zaragoza H (2009) The probabilistic relevance framework: BM25 and beyond. Foundations and Trends in Information Retrieval 3(4):333–389. <https://doi.org/10.1561/15000000019>, URL <https://www.nowpublishers.com/article/DownloadEBook/INR-019>
- [30] RSL Collective (2025) Really simple licensing (RSL) 1.0 specification. RSL-SPEC-1.0, Industry Specification, Recommendation, URL <https://rslstandard.org/rsl>, published 10 December 2025. Accessed 1 October 2026
- [31] Rundgren A, Jordan B, Erdtman S (2020) JSON canonicalization scheme (JCS). RFC 8785, RFC Editor, <https://doi.org/10.17487/RFC8785>, URL <https://www.rfc-editor.org/rfc/rfc8785>
- [32] SPUR Coalition (2026) Content telemetry specification, version 1.0. Open specification, URL <https://github.com/SPUR-Coalition/telemetry/blob/main/SPECIFICATION.md>, published 2 September 2026. Accessed 1 October 2026
- [33] Walters WH, Wilder EI (2023) Fabrication and errors in the bibliographic citations generated by ChatGPT. Scientific Reports 13:14045. <https://doi.org/10.1038/s41598-023-41032-5>, URL <https://www.nature.com/articles/s41598-023-41032-5>