WP-022 | Agentic RAG Citation Integrity | June 2026

Cryptographic Citation Integrity in Agentic RAG Pipelines: Per-Chunk Retrieval Proofs and Merkle-Anchored Source Attestation

AffixIO Research June 2026 WP-022 Download PDF

Agentic RAG systems cite sources they never retrieved. Regulators notice. This paper walks through AffixIO's per-chunk zero-knowledge retrieval proofs: each cited passage is cryptographically bound to a Merkle-anchored document chunk before the model sees it. Auditors verify citations offline. Article 13 transparency stops being a slide deck promise.

CONTENTS

1. The RAG Citation Problem

Retrieval-Augmented Generation was developed to address a fundamental limitation of language models: their knowledge is frozen at training time and they confabulate when asked about information not present in their training data. By retrieving relevant documents at inference time and presenting them to the model as context, RAG grounds the model's responses in current, specific information and substantially reduces hallucination rates for factual questions.

The RAG paradigm has succeeded to the point where it is now the default architecture for enterprise AI systems in 2026. Knowledge bases, document retrieval systems, and agentic multi-hop retrieval pipelines are ubiquitous in enterprise AI deployments for legal research, financial analysis, compliance review, medical information, and customer service.

Yet RAG has introduced a new problem that is underappreciated in the industry: the citation problem. When a RAG system cites a source, it is making a claim about the relationship between the cited content and the generated response. This claim can fail in multiple ways: the cited source may not have been retrieved, the cited source may have been retrieved but not used in generating the specific claim, the cited passage may be inaccurately paraphrased, or the claim may be a synthesis of multiple sources that is not faithful to any individual source.

In low-stakes contexts, RAG citation errors are annoying. In high-stakes contexts, they are consequential. A legal AI system that cites cases it did not actually retrieve creates a professional conduct risk for the lawyers who rely on it. A medical information system that attributes a clinical claim to a guideline it did not read creates a patient safety risk. A financial analysis system that fabricates citations to regulatory documents creates a compliance risk. The EU AI Act's Article 13 transparency requirements, which mandate that users be given sufficient information to interpret and use AI outputs appropriately, cannot be satisfied by AI systems that fabricate citations.

2. Agentic RAG Architecture in 2026

Modern enterprise RAG has evolved substantially from the simple retrieve-and-generate pattern of early implementations. Agentic RAG, the dominant enterprise pattern in 2026, embeds retrieval inside multi-agent systems where specialised agents handle query decomposition, retrieval, validation, synthesis, and citation verification in parallel or in sequence.

Query Decomposition

Complex queries are decomposed by a planning agent into sub-queries, each targeted at a specific aspect of the question. A query about "the regulatory requirements for AI systems in financial services under the EU AI Act" might be decomposed into sub-queries about Article 9 scope, Article 10 data governance, Article 11 technical documentation, Article 12 record-keeping, and Article 13 transparency. Each sub-query is dispatched to a retrieval agent.

Multi-Source Retrieval

Retrieval agents query multiple retrieval systems in parallel: a dense vector index for semantic search, a sparse BM25 index for keyword search, a graph database for relationship queries, and specialised domain indexes for regulatory text, technical standards, or proprietary knowledge bases. Each retrieval system returns ranked chunks; the retrieval agent scores, filters, and deduplicates the results.

Validation and Corrective RAG

Self-reflective RAG and corrective RAG patterns, prominent in 2026 enterprise deployments, add a validation step: a validation agent checks whether the retrieved chunks are sufficient to answer the query, whether they are relevant, and whether they are consistent with each other. If validation fails, the validation agent triggers additional retrieval or re-ranking. This corrective loop substantially reduces hallucination rates for complex multi-hop queries.

Synthesis and Citation

The synthesis agent receives the validated chunks and generates the response. It is expected to cite specific chunks, not just acknowledge that sources were used. The synthesis agent's ability to produce faithful citations depends on the quality of the source context, the model's instruction-following capability, and the citation format required by the application. All three of these factors are subject to failure; none of them is amenable to cryptographic verification in the current RAG architecture.

---

3. Three Categories of RAG Citation Failure

RAG citation failures fall into three distinct categories, each with different causes and different consequences for users who rely on cited sources.

Category 1: Phantom Citations

A phantom citation occurs when the language model generates a citation to a source that was not retrieved. This is a form of hallucination specific to RAG systems: the model generates content that appears to be grounded in a source but is in fact generated from parametric knowledge or confabulated entirely. Phantom citations are particularly insidious because they provide false reassurance: the presence of a citation increases user trust even when the cited source does not support the claim.

Research on legal AI systems has found phantom citation rates of 17% to 34% in early RAG deployments, with some models generating plausible-looking but non-existent case citations. Even with more recent models and improved RAG pipelines, phantom citations have not been eliminated; they occur at lower rates but in unpredictable circumstances that are difficult to anticipate.

Category 2: Misattributed Citations

A misattributed citation occurs when a source was retrieved but the cited claim is not supported by the source. The source was retrieved and presented to the model; the model cited it; but the model's paraphrase of the source's content is inaccurate, selective, or out of context. The model did not fabricate the source, but it fabricated the relationship between the source and the claim.

Category 3: Synthesis Without Attribution

The most subtle citation failure occurs in synthesis: the model produces a claim that is the synthesis of multiple retrieved sources, attributed to one source or to no source. Synthesis is often the most valuable output of a RAG system, combining information from disparate sources into a coherent response. But synthetic claims that are attributed to individual sources misrepresent the reasoning process, and synthetic claims with no attribution cannot be verified by the user.

---

4. Why Standard RAG Governance Is Insufficient

Current approaches to RAG citation quality rely on a combination of prompt engineering, model selection, and post-hoc evaluation. Each of these approaches has limitations that make them insufficient for high-stakes enterprise deployments.

Prompt Engineering

System prompts that instruct the model to cite only retrieved sources and to avoid fabricating citations reduce phantom citation rates but do not eliminate them. Language models are trained to follow instructions, but they are also trained to produce plausible and helpful responses, which sometimes overrides citation instructions when the model's parametric knowledge suggests a plausible citation that was not retrieved. Prompt engineering cannot provide cryptographic guarantees.

Model Selection

Some models exhibit lower phantom citation rates than others, and fine-tuning on citation-accurate data can improve citation fidelity. However, citation accuracy is not a universal property: a model that cites accurately in one domain may be less accurate in another, and citation accuracy is not systematically tested in standard model evaluations. Model selection cannot substitute for runtime citation verification.

Post-Hoc Evaluation

Post-hoc citation evaluation, either by automated fact-checking systems or by human reviewers, can identify citation errors after generation. This is useful for quality monitoring but does not prevent citation errors from reaching users. In high-throughput enterprise deployments, post-hoc evaluation of every response is impractical. In time-sensitive workflows, the delay introduced by post-hoc evaluation is unacceptable.

---

5. Cryptographic Citation Integrity: The Approach

AffixIO's cryptographic citation integrity architecture introduces a verification layer between retrieval and generation that produces a cryptographic record of which chunks were retrieved, when, from which source, and with what query context. This record enables independent verification of citations after generation without requiring post-hoc evaluation.

> **The Core Claim**
>
> Instead of asking "did the model cite this source?" (unverifiable), cryptographic citation integrity asks "was this source retrieved in the session that produced this response?" (provable). The distinction is significant: we cannot prove what the model used internally, but we can prove what was available to it. A citation to a source that was not retrieved is a phantom citation, detectable by proof verification.

What Cryptographic Citation Integrity Proves

For each response, AffixIO's system produces a citation integrity proof asserting:

- Each cited source identifier corresponds to a chunk that was retrieved in this session
- Each retrieved chunk was retrieved with the query that is committed in the session record
- The retrieval timestamp is within the session time window

- The chunk content has not been modified between retrieval and citation

These four assertions together eliminate phantom citations (by requiring each citation to correspond to a retrieved chunk), detect retrieval tampering (by committing to chunk content at retrieval time), and provide a verifiable session record (by linking citations to a specific retrieval session).

---

6. Per-Chunk ZK Retrieval Proofs

The foundation of AffixIO's citation integrity architecture is the per-chunk retrieval proof: a ZK proof generated at retrieval time for each chunk returned by the retrieval system.

What a Retrieval Proof Contains

For each retrieved chunk, the AffixIO retrieval proof system generates a proof asserting:

- The chunk content hash matches the hash of the chunk as stored in the retrieval index
- The query vector used for retrieval is committed (without revealing the query text)
- The retrieval timestamp falls within the session's authorised time window
- The source identifier is within the set of authorised sources for this query context
- A session nonce prevents replay of retrieval proofs across sessions

The proof is generated by the retrieval system's AffixIO middleware layer and is returned alongside the chunk content. The generating-side middleware holds the retrieval index's signing key; the consuming application verifies the proof before presenting the chunk to the synthesis model.

Chunk Content Commitment

The chunk content hash included in the retrieval proof is a SHA-256 hash of the chunk text as it exists in the retrieval index at the time of retrieval. If the chunk is cited in the

response, the verifier can confirm that the cited content matches the chunk's committed hash. If the cited content differs from the hash, either the chunk was modified after retrieval or the model paraphrased the chunk in a way that has been misattributed as a direct quote.

Source Authority Verification

The source identifier committed in the retrieval proof is mapped to a source authority record: the organisation, URL, timestamp, and licence type of the original source. This enables not just citation integrity verification but source authority verification: the verifier can confirm that the cited source is an authorised, known source rather than an adversarially injected document.

---

7. Merkle-Anchored Response Attestation

After retrieval and before synthesis, AffixIO constructs a response session Merkle tree. The leaves of this tree are the retrieval proofs for all chunks retrieved in the session. The Merkle root commits to the complete set of retrieved chunks; the response to the user includes this root as a citation integrity anchor.

Response-Level Citation Map

When the synthesis model generates a response citing specific sources, the AffixIO synthesis middleware constructs a citation map: a structured record linking each citation in the response text to the leaf index of the corresponding chunk in the session Merkle tree. The citation map is included in the response metadata alongside the Merkle root.

User-Facing Verification

A user who receives a RAG response with AffixIO citation integrity can verify any citation as follows:

1. Obtain the session Merkle root and the citation map from the response metadata

2. For each citation, request the Merkle inclusion proof for the cited chunk from the AffixIO verification API
3. Verify the inclusion proof against the session root
4. Compare the cited content against the chunk content hash committed in the retrieval proof

This verification process confirms that the cited source was retrieved in the session, that the chunk content has not been modified, and that the citation is not a phantom. It does not confirm that the model accurately represented the chunk's content, but it provides a verifiable starting point for human review of the cited material.

---

8. Multi-Hop RAG and Proof Chaining

Agentic RAG pipelines frequently use multi-hop retrieval: the result of one retrieval step informs a subsequent query, building a chain of evidence across multiple retrieval steps. Multi-hop RAG introduces additional citation complexity because claims in the final response may be derived from a chain of retrieved documents rather than from any single document.

Proof Chain Construction

In a multi-hop RAG pipeline, AffixIO maintains a proof chain across retrieval hops. Each hop's retrieval proofs are committed in a hop-level Merkle tree; the root of each hop-level tree is a leaf in the session-level Merkle tree. The session root therefore commits to the complete multi-hop retrieval history, enabling an auditor to trace any final response claim back through the chain of evidence that produced it.

Intermediate Synthesis Commitment

In multi-hop reasoning, intermediate synthesis steps produce intermediate conclusions that are used as inputs to subsequent retrieval queries. These intermediate conclusions are committed as part of the proof chain, enabling auditors to verify not just the final response's citations but the intermediate reasoning steps

that led to it. This is particularly important for complex analytical tasks where the reasoning chain is as significant as the final conclusion.

Cross-Agent Citation Attribution

In multi-agent RAG systems where different agents retrieve and synthesise different aspects of the response, AffixIO's proof chain architecture maintains cross-agent attribution: each agent's retrieval proofs are recorded in the overall session proof chain, and the final response's citation map attributes each claim to the specific agent's retrieval that supports it. This enables precise attribution in systems where the same query may be answered by contributions from multiple specialised agents.

---

9. Corrective RAG and Self-Reflective RAG Integration

Corrective RAG and self-reflective RAG patterns, which add validation loops to the retrieval and synthesis process, integrate naturally with AffixIO's citation integrity architecture and enhance its guarantees.

Corrective RAG Integration

In corrective RAG, a validation agent assesses the relevance and quality of retrieved chunks and triggers additional retrieval if the initial results are insufficient. AffixIO's per-chunk retrieval proofs are generated for every retrieval step, including corrective retrieval steps. The session Merkle tree includes leaves for all retrieval events, including those that were triggered by the corrective loop. An auditor can see not just the sources used in the final response but the sources that were considered and rejected or supplemented during the corrective process.

Self-Reflective RAG and Claim-Level Verification

Self-reflective RAG adds a post-synthesis validation step where the model checks its own output for factual consistency with the retrieved sources. When self-reflective RAG is integrated with AffixIO's citation integrity system, the validation step generates a claim-level verification proof: for each claim in the response, a proof asserting that at

least one retrieved chunk supports the claim within a specified semantic similarity threshold. This provides a stronger citation guarantee than the basic retrieval proof, approaching the claim-level attribution that human citation practice requires.

---

10. EU AI Act Article 13 Transparency Compliance

EU AI Act Article 13 requires that high-risk AI systems provide users with sufficient information to interpret the system's output and use it appropriately. For RAG-based AI systems, this transparency requirement has direct implications for citation practice: users must be able to understand what sources the AI used and how those sources relate to the AI's output.

The Current Transparency Gap

Most RAG systems provide transparency at the source level: they tell users which documents were retrieved and used. This is useful, but it does not satisfy Article 13's requirement to enable users to interpret and use the output appropriately. A list of retrieved documents does not tell the user which specific content in the response was derived from which document, whether the citations are accurate, or whether the synthesis is faithful to the sources.

Cryptographic Citation as Transparency Infrastructure

AffixIO's citation integrity architecture provides transparency at the claim level: for each claim in the response, users can verify which retrieved chunk supports it and whether the citation is accurate. This is transparency that is cryptographically verifiable, not merely asserted. It enables users to exercise their Article 13 right to interpret and use the AI output appropriately, because they can verify the basis for each claim rather than trusting the AI's assertion that sources were consulted.

Audit and Regulatory Reporting

For regulated enterprise deployments, AffixIO's session Merkle roots and citation maps provide an audit record that satisfies regulatory inspection requirements. A regulator

investigating an AI system's use of specific sources in a regulated domain can verify, through the citation integrity proof chain, that the cited sources were retrieved, that the retrieved content matches the citations, and that the response was generated in a session with a verifiable retrieval history. This evidence-quality audit record is substantially stronger than a log of document titles.

---

11. Implementation and Performance

Integration Points

AffixIO's citation integrity middleware integrates at two points in the RAG pipeline: the retrieval layer and the synthesis layer. At the retrieval layer, the middleware intercepts each retrieval result and generates a chunk retrieval proof before returning the chunk to the requesting agent. At the synthesis layer, the middleware intercepts the generated response and constructs the citation map, linking each citation to the corresponding retrieval proof, before returning the response to the user.

The integration is compatible with all major RAG frameworks, including LangChain, LlamaIndex, Haystack, and custom retrieval implementations using Azure AI Search, Elasticsearch, Pinecone, or Weaviate as the underlying vector store.

Retrieval Proof Generation Latency

Per-chunk retrieval proof generation adds between 3 and 12 milliseconds per chunk, depending on chunk size and circuit complexity. For a typical RAG response that draws on 10 to 20 retrieved chunks, the total proof generation overhead is 30 to 240 milliseconds. This is acceptable for most enterprise applications where the total latency budget is measured in seconds, but may require optimisation for sub-500ms latency targets.

Proof Generation Parallelisation

Retrieval proofs for chunks returned in a single retrieval batch can be generated in parallel, reducing the effective latency overhead to the maximum proof generation

time for the batch rather than the sum. AffixIO's retrieval middleware implements parallel proof generation by default, with a configurable thread pool size that can be tuned to balance proof generation throughput against CPU usage.

Session Merkle Root Construction

Session Merkle root construction, which occurs after retrieval and before synthesis, takes between 1 and 5 milliseconds for sessions with up to 1,000 retrieved chunks. This is negligible in the context of the overall response latency. For sessions with more than 1,000 chunks, a streaming Merkle construction approach maintains the root incrementally as chunks are retrieved, adding no perceptible latency.

---

12. Conclusions

Agentic RAG is not just a retrieval architecture; it is a trust architecture. Users of RAG systems extend trust to AI responses precisely because those responses are supposedly grounded in verifiable sources. When that trust is misplaced, because citations are phantom, misattributed, or synthesis without attribution, the consequences range from user frustration to professional and patient safety risks to regulatory non-compliance.

AffixIO's cryptographic citation integrity architecture addresses the citation problem at the infrastructure layer rather than the model layer. By generating per-chunk ZK retrieval proofs at retrieval time and constructing Merkle-anchored response attestations that verifiably link output claims to retrieved sources, the architecture makes citation verification a first-class capability of the RAG pipeline rather than an afterthought.

The architecture satisfies EU AI Act Article 13's transparency requirements through cryptographic source attribution, provides multi-hop proof chains for complex agentic RAG workflows, integrates with corrective and self-reflective RAG patterns to enhance citation guarantees, and generates evidence-quality audit records for regulated enterprise deployments.

As agentic RAG becomes the standard enterprise AI architecture and regulatory expectations for AI transparency increase, the ability to prove citation integrity will become as fundamental as the ability to reduce hallucination. AffixIO's citation integrity middleware is available as an open-source component compatible with all major RAG frameworks, with enterprise support for deployment in regulated environments.

Related reading

- WP-005: Source Verification as a Zero-Knowledge Circuit Input
- WP-015: Cryptographic Audit Trails for Autonomous AI Agents
- WP-021: Beyond C2PA: ZK Content Provenance That Survives Metadata Stripping

---

Frequently asked questions

What is RAG citation integrity?

It means every citation in an agentic RAG response can be traced to a specific retrieved chunk, with cryptographic proof that chunk existed in your corpus at query time.

How do zero-knowledge proofs help RAG?

They let you prove a chunk was retrieved and hashed without exposing the full document store or user queries to third-party auditors.

Does this satisfy EU AI Act requirements?

Per-chunk attestation supports Article 13 transparency and Article 12 record-keeping by producing verifiable retrieval evidence, not just model output logs.

All White Papers

▶ About

▶ Solutions

▶ Legal

▶ Trust & Security

Contact

truth layer | yes | no | proof