International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Cryptographically Verifiable Retrieval-Augmented Generation: A Tripartite Architecture for Decentralized Provenance, Compute-to-Data Privacy, and Automated Fact-Checking

Sheetal Laroiya
Department of Computer Science and Engineering
Chandigarh University Punjab,India
Sheetal.e15433@cumail.in

Purnendu kumar Ghosh
Department of Computer Science and Engineering
Chandigarh University Punjab, India Rumag789@gmail.com

Ketan
Department of Computer Science and Engineering
Chandigarh University Punjab, India
Ketansaini8765@gmail.com

Subhanshu Raj
Department of Computer Science and Engineering
Chandigarh University Punjab, India
Subhanshuraj49@gmail.com

Abstract—Contemporary Large Language Models (LLMs) deployed within Retrieval-Augmented Generation (RAG) pipelines suffer from three distinct vulnerabilities: epistemological opacity (hallucinations), non-consensual data exploitation, and a lack of granular provenance. This paper proposes a tripartite architecture to resolve these deficits. First, we introduce an Indexer and Provenance Layer utilizing Decentralized Identifiers (DIDs), Verifiable Credentials (VCs), and on-chain mapping to establish immutable audit trails for retrieved context. Second, we present a Privacy-Preserving Compute-to-Data paradigm leveraging tokenized access control to facilitate economic incentivization without exposing raw data to consumers. Finally, we formalize a Verifiable RAG Pipeline equipped with a multi-strategy Verifier Agent to autonomously audit LLM generation against cryptographically anchored evidence.

Index Terms—Retrieval-Augmented Generation, Decentralized Identifiers, Verifiable Credentials, Compute-to-Data, Natural Language Inference, Hallucination Mitigation, Zero-Trust AI, Blockchain, Privacy-Preserving AI.

I. Introduction

The proliferation of Retrieval-Augmented Generation (RAG) systems has democratized access to domain-specific knowledge by grounding generative models in external databases [1]. However, current implementations operate on implicit trust. When an LLM generates a claim, the provenance of the underlying data is easily obscured, and the intellectual property rights of data creators are frequently bypassed. Furthermore, the ingestion of sensitive data into centralized vector databases presents severe privacy risks.

To achieve a zero-trust, economically viable, and mathematically verifiable generative ecosystem, we propose an integrated architecture bridging Web3 cryptographic primitives with advanced NLP orchestration. Three core deficits are addressed:

- **Epistemological opacity**: LLMs fabricate facts not supported by retrieved context.
- **Non-consensual data exploitation**: Raw data ingested without owner consent or compensation.
- **Lack of granular provenance**: No immutable record links a generated claim to its evidence.

Our tripartite solution — an Indexer and Provenance Layer, a Compute-to-Data access framework, and a Verifiable Pipeline with an agentic Verifier — resolves each deficit through a unified cryptographic trust chain.

II. Component I: Indexer & Provenance Layer

The foundational layer shifts data addressing from location-based URLs to content-based cryptographic hashes, ensuring immutability and verifiable ownership.

A. Content Addressing and Decentralized Storage

Raw corpora undergo sanitization and chunking before deployment to decentralized storage networks (e.g., IPFS or Arweave). This yields a Content Identifier (CID) — a permanent, immutable pointer to the dataset payload. Because a CID is derived from the content's cryptographic hash, any modification produces a distinct CID, rendering tampering self-evident.

B. Verifiable Credentials and Decentralized Identifiers

To establish ownership and usage rights, the data proprietor generates a W3C-compliant Verifiable Credential (VC) [7]. This JSON-LD document asserts the

dataset's CID, licensing parameters, and optional datatoken pricing metrics. The VC is cryptographically signed using the proprietor's Decentralized Identifier (e.g., *did:ethr*), ensuring that any modification to the metadata invalidates the proof.

C. On-Chain Registry A minimal registry deployed on a Layer-2 network (e.g., Scroll, Base, or Arbitrum) maps a unique *dataset_id* to its CID, the VC hash, and the datatoken address. The on-hain struct is defined as:

{ datasetId, CID, VC_Hash,
Datatoken_Address,Owner_Address, Timestamp }

A RAG consumer independently fetches the VC from IPFS, verifies the DID signature offline, and cross-references VC_Hash with the immutable ledger — requiring no trusted intermediary.

TABLE I: On-Chain Registry Fields

Field	Purpose
datasetId	Unique dataset identifier
CID	IPFS content address
VC_Hash	Hash of Verifiable Credential
Datatoken_Address	Access-gate ERC token contract
Owner_Address	DID-linked proprietor wallet
Timestamp	Block timestamp of registration

III. Component II: Privacy-Preserving Data Access

Standard RAG ingestion requires raw text to be transferred to the consumer's environment for embedding, compromising data sovereignty. We resolve this via a Compute-to-Data (C2D) architecture [3], adapted for LLM workloads.

A. Tokenized Compute Orchestration

The data proprietor provisions a secure, containerized compute endpoint exposing a strictly controlled REST API. Access is gated through three on-chain stages:

1. Token Escrow: Consumer transfers ERC-20/ERC-721 datatokens to an AccessEscrow smart contract.

2. Event Emittance: Contract emits a cryptographically signed AccessGranted event upon successful transfer.

3. Validation & Execution: Compute container validates the event and returns only processed embeddings or evidentiary extracts — raw dataset egress is strictly prohibited.

B. Trusted Execution Environments (TEEs)

Future iterations require the compute container to be hosted within a TEE (e.g., Intel SGX or AWS Nitro Enclaves) [8]. The endpoint thereby generates a remote attestation proof, guaranteeing to the consumer that the prescribed retrieval algorithm was executed without interference.

IV. Component III: The Verifiable RAG Pipeline

A. Retrieval and Reranking

Documents are ingested in overlapping chunks (200–800 tokens) and embedded via dense vector models (e.g., all-mpnet-base-v2). The vector database maps each embedding to provenance metadata: {CID, datasetId, chunkId, offset}. A Cross-Encoder Reranker (e.g., ms-marco-MiniLM) re-scores the initial distribution to isolate the top-m passages [11].

B. LLM Generation with Inline Citations

The Reasoning LLM is constrained via system prompting to operate exclusively upon the retrieved context. It appends a structured citation tag (e.g., [CID:chunkId]) to every factual assertion, creating a machine-readable provenance trail within the generated text.

C. Verifier Agent and Formal Abstention Logic

A deterministic Verifier Agent audits the LLM draft using NLI entailment [9] and iterative retrieval. Let $C = \{c_1,...,c_n\}$ be discrete factual claims and $P = \{p_1,...,p_k\}$ the retrieved passages. For each claim ci, we compute:

$$S_{max}(c_i) = \max_j E(c_i, p_j)$$

where E is the DeBERTa-v3 NLI entailment score. The verification status is:

$$V(c_i) = \text{SUPPORTED if } S_{max} \geq \tau_{entail}$$

$$V(c_i) = \text{UNSUPPORTED otherwise}$$

If UNSUPPORTED, the agent initiates targeted secondary retrieval. The pipeline implements a hard Abstention Protocol: the system yields ABSTAIN (triggering human intervention) if:

$$[\sum \blacksquare(V(c_i)=\text{UNSUPPORTED})] / |C| > X$$

where X is the maximum acceptable unsupported-claim ratio. This prevents any response with an unacceptable hallucination rate from reaching the user.

TABLE II: Verification Pipeline Stages

Stage	Model / Tool	Output
Claim Extraction	LLM + parser	Set C
NLI Scoring	DeBERTa-v3	S_max per claim
Secondary Retrieval	Dense retriever	Augmented P
Abstention Check	Threshold X	PASS / ABSTAIN

V. Evaluation Methodology and Risk Mitigation

Empirical validation requires benchmarking across three primary vectors:

Latency Overhead. The cumulative penalty from on-chain verification, reranking, and NLI auditing must be quantified. Mitigations: asynchronous verification for non-blocking UI rendering and low-latency Layer-2 rollups [12].

Cryptographic Integrity. Resilience of VC validation must be stress-tested against DID spoofing, signature forgery, replay attacks, and malformed DID documents.

Verifier Efficacy. τ_{entail} must be rigorously calibrated on a held-out validation set to balance false negatives (rejecting valid deductions) against false positives (permitting hallucinations). F1 scoring across stratified hallucination types is recommended.

TABLE III: Benchmark Dimensions

Vector	Metric	Mitigation
Latency	P99 e2e (ms)	Async verify + L2
Crypto Integrity	Spoof success %	Adversarial DID suite
Verifier F1	Hallucination F1	τ calibration on val

VI. Related Work

Prior work on factual grounding in LLMs has largely focused on post-hoc retrieval augmentation [1] or output calibration. Lewis et al. [1] introduced the foundational RAG paradigm, demonstrating that grounding generation in retrieved documents reduces hallucination on open-domain QA benchmarks.

Decentralized provenance mechanisms have been explored in data marketplaces [3] and scientific publishing [6], but have not been integrated into LLM inference pipelines. The W3C DID [2] and VC [7] specifications provide the cryptographic substrate upon which our Indexer Layer is built.

NLI-based fact verification has been applied to claim checking [5], yet prior systems operate as standalone pipelines rather than integrated pipeline guards. Self-RAG [4] introduces reflection tokens for retrieval decisions but lacks formal abstention logic and cryptographic provenance. Our Verifier Agent closes this gap.

TABLE IV: Comparison with Related Approaches

Approach	Provenance	Privacy	Verification
RAG [1]	None	None	None
Self-RAG [4]	None	None	Partial
Ocean [3]	On-chain	C2D	None
Ours	**DID/VC/CID**	**TEE+C2D**	**NLI+Abstain**

VII. Conclusion

By unifying Decentralized Identifiers, compute-to-data tokenomics, and an ensemble-driven Verifier Agent, this architecture resolves the fundamental trust deficits inherent in modern RAG systems. It provides an immutable, cryptographically verifiable bridge between raw data originators and generative AI consumers, simultaneously protecting intellectual property and ensuring rigorous factual grounding.

Future work includes empirical deployment on production RAG workloads, integration with emerging DID methods (e.g., did:ion), formal security proofs for the escrow protocol, and red-team evaluation of the NLI verifier against adversarially crafted hallucinations.

References

[1] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," NeurIPS, 2020.

[2] W3C, "Decentralized Identifiers (DIDs) v1.0," W3C Recommendation, 2022.

[3] T. McConaghy et al., "Ocean Protocol: A Decentralized Substrate for AI Data & Services," arXiv:1911.07143, 2019.

[4] A. Asai et al., "Self-RAG: Learning to Retrieve, Generate, and Critique Through Self-Reflection," ICLR, 2024.

[5] J. Thorne et al., "FEVER: A Large-Scale Dataset for Fact Extraction and VERification," NAACL, 2018.

[6] A. Tardif et al., "Blockchain-Based Provenance for Scientific Data," IEEE Access, 2021.

[7] M. Sporny et al., "Verifiable Credentials Data Model v1.1," W3C Recommendation, 2022.

[8] P. Micikevicius et al., "Trusted Execution Environments: A Survey," ACM Computing Surveys, 2023.

[9] Y. He et al., "DeBERTa: Decoding-Enhanced BERT with Disentangled Attention," ICLR, 2021.

[10] S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," Found. & Trends IR, 2009.

[11] L. Wang et al., "Cross-Encoders for Re-Ranking in Dense Retrieval," arXiv:2205.09060, 2022.

[12] A. Kirilenko et al., "Layer-2 Scaling Solutions for Ethereum," IEEE Blockchain, 2023.