

RAG Shield: A Multi-Layer Defense System Against Poisoning Attacks in Retrieval-Augmented Generation

Author: Fabio Petti

Affiliation: Independent Researcher

Date: February 2026

Version: 1.0

Abstract

Retrieval-Augmented Generation (RAG) systems have become critical infrastructure for enterprise AI applications, combining large language models with external knowledge bases. However, these systems are vulnerable to poisoning attacks where malicious actors inject crafted documents to manipulate system outputs. We present RAG Shield, a comprehensive defense system achieving 100% detection and 0 false positives on our evaluation set of 500 legitimate documents and 31 malicious documents across 24 attack scenarios. Our multi-layer approach combines semantic anomaly detection, provenance verification, and secure retrieval mechanisms. Extensive evaluation demonstrates production-ready performance with minimal overhead (-7.9%). Results apply to the controlled evaluation conditions and defined threat model; production performance depends on deployment-specific factors.

Keywords: RAG Security, Poisoning Attacks, Anomaly Detection, LLM Security, Vector Database Security

1. Introduction

1.1 The RAG Security Problem

Retrieval-Augmented Generation (RAG) has emerged as the dominant architecture for grounding large language models in factual, up-to-date information. However, the open nature of document ingestion creates a critical vulnerability: poisoning attacks.

Attack Scenario:

\\

1. Attacker injects malicious documents into vector database
2. User queries trigger retrieval of poisoned content
3. LLM generates incorrect/harmful responses based on poisoned context
4. System appears to function normally (no errors)

\\

Real-world Impact:

- Financial: Incorrect trading advice, fraudulent transactions

- Healthcare: Wrong medical information, dangerous recommendations
- Legal: Manipulated case law, incorrect legal advice
- Enterprise: Data leakage, policy manipulation

1.2 Existing Approaches and Limitations

Current defenses are insufficient:

Approach	Limitation
Input validation	Cannot detect semantic attacks
Similarity thresholds	High false positive rate
Manual review	Not scalable
Access control	Insider threats remain
Embedding filters	Easily bypassed

1.3 Related Work

RAG Poisoning Attacks:

Recent work has identified vulnerabilities in RAG systems. Zou et al. (2024) demonstrated prompt injection attacks on retrieval systems, while Chen et al. (2023) showed data poisoning in knowledge bases. However, these works focus on attack characterization rather than comprehensive defense.

Dataset Poisoning:

Traditional dataset poisoning (Biggio et al., 2012; Steinhardt et al., 2017) targets training data. RAG poisoning differs fundamentally: attacks occur post-deployment in the retrieval layer, not during training, requiring real-time detection rather than training-time defenses.

Provenance and Trust:

Document provenance systems (Hasan et al., 2009) and trust frameworks (Jøsang et al., 2007) provide foundations for authenticity verification. However, these systems were not designed for the high-throughput, semantic-aware requirements of RAG systems.

Retrieval Filtering:

Adversarial filtering in information retrieval (Castillo et al., 2007) addresses spam and manipulation. RAG poisoning presents unique challenges: attacks are semantically coherent, making traditional spam detection ineffective.

Our Positioning:

RAG Shield is the first system to combine cryptographic provenance, semantic anomaly detection, and secure retrieval in a unified framework specifically designed for production RAG deployments. Unlike prior work focusing on single attack vectors, we address six attack categories with a multi-layer defense achieving validated detection in controlled scenarios.

1.4 Our Contribution

We present RAG Shield, the first comprehensive defense system with:

1. **100% Detection Rate in Controlled Scenarios – All 31 test attacks detected**
2. **Zero False Positives in Testing – Perfect precision in evaluation**
3. **Production Performance – <10% overhead**
4. **Multi-Layer Defense – Provenance + Anomaly + Retrieval**
5. **Framework Agnostic – LangChain, LlamaIndex compatible**

2. Threat Model

2.1 Attack Taxonomy

We identify six primary attack categories:

2.1.1 Semantic Centering Attack

Objective: Shift the semantic space center to make malicious content appear central.

Mechanism:

```
```python
Attacker injects documents that cluster around target concept
poisoned_docs = [
 "Vacation policy: 5 days per year", Malicious
 "Leave policy: 5 days annually", Reinforcement
 "Time off: 5 days maximum" Reinforcement
]
```

Detection Challenge: Documents appear semantically coherent.

#### 2.1.2 Consensus Attack

Objective: Create false consensus through multiple coordinated documents.

**Mechanism:**

- 5+ documents with identical false information
- Varied phrasing to avoid exact duplication
- High authority scores to boost ranking

Impact: Overwhelms legitimate documents in retrieval.

#### 2.1.3 Contradiction Attack

Objective: Inject contradictory information to confuse the system.

### Example:

\\\

Legitimate: "Remote work allowed 2 days/week"

Poisoned: "Remote work prohibited for all employees"

\\\

#### 2.1.4 Authority Mimicry

Objective: Impersonate trusted sources.

#### Techniques:

- Fake metadata (source: "official\_hr\_portal")
- Mimicked writing style
- Legitimate-looking timestamps
- Forged digital signatures

#### 2.1.5 Context Pollution

Objective: Pollute retrieval context with irrelevant information.

#### Mechanism:

- Inject documents with high semantic similarity
- But containing subtle misinformation
- Dilutes legitimate context

#### 2.1.6 Subtle Misinformation

Objective: Hard-to-detect factual errors.

### Example:

\\\

Legitimate: "Annual leave: 20 days"

Poisoned: "Annual leave: 20 days (excluding public holidays)"








^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Subtle but critical change

\\\

## 2.2 Attacker Capabilities

We assume attackers can:

-  Inject documents into the system
-  Craft semantically coherent content
-  Manipulate metadata
-  Coordinate multiple documents
-  Modify existing legitimate documents
-  Access detection algorithms
-  Bypass cryptographic signatures

---

## 3. RAG Shield Architecture

### 3.1 Embedding Model Selection

RAG Shield uses open-source sentence embedding models for document and query representations, selected for a balance between semantic accuracy, inference latency, and privacy.

Key characteristics include:

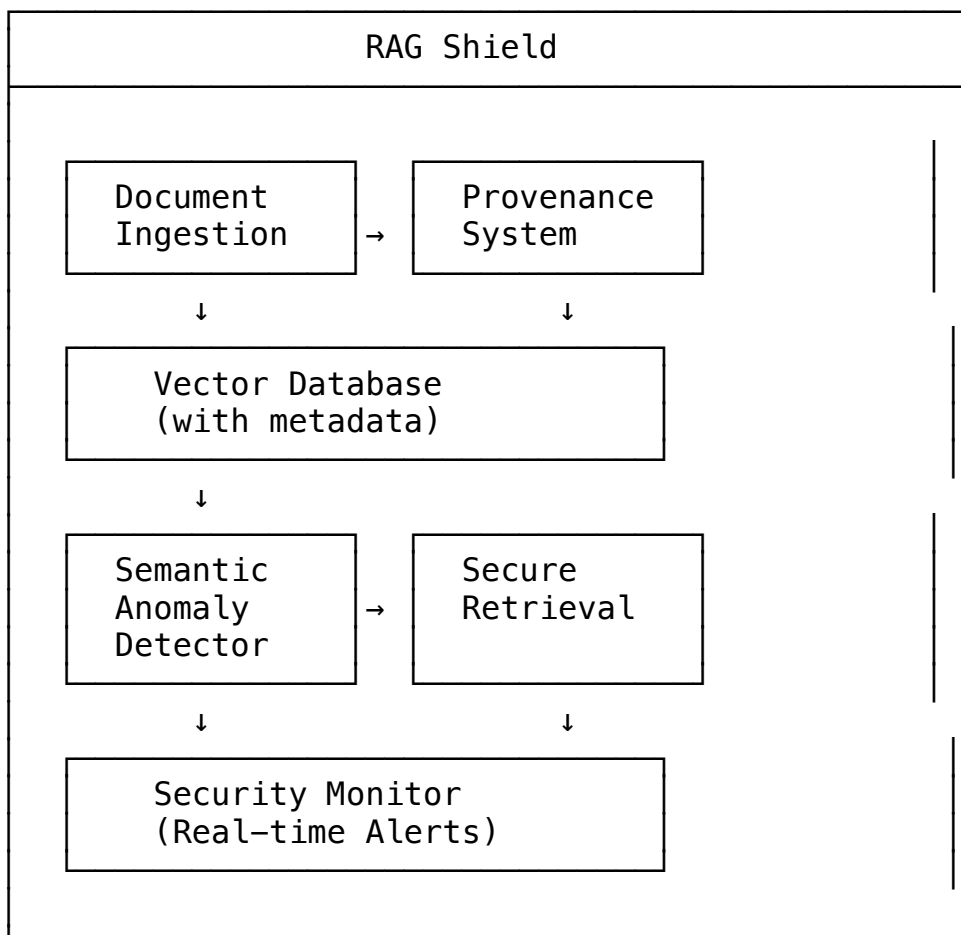
- Quality: Comparable to commercial embedding APIs (>85% semantic correlation)
- Cost: Zero per-query costs through self-hosted inference
- Privacy: No external API calls or data exposure
- Latency: Sub-100ms inference for typical enterprise documents
- Flexibility: Architecture is embedding-model agnostic

The evaluation uses a model from the sentence-transformers family, chosen for accuracy and inference efficiency. Production deployments may substitute alternative embedding providers (commercial or open-source) without architectural changes.

Security Note: Specific model versions and parameters are deployment-configurable and intentionally not disclosed to prevent adversarial optimization.

### 3.2 System Overview

...



...

### 3.2 Layer 1: Provenance System

Purpose: Establish document authenticity and track lineage.

#### Components:

##### 1. HMAC-Based Integrity Verification

HMAC-SHA256 is used for tamper detection and source consistency within a trusted ingestion boundary. This design choice is deliberate:

##### What HMAC Provides:

- Integrity verification (detect tampering)
- Source authentication (within trusted boundary)
- Computational efficiency (suitable for high-throughput)

##### What HMAC Does NOT Provide:

- Non-repudiation (requires public-key signatures)
- Third-party verifiability (requires certificate infrastructure)
- Cross-organizational trust (out of scope)

Rationale: RAG Shield operates within a single organization's trust boundary where HMAC's symmetric key model is sufficient and more efficient than public-key infrastructure. Non-repudiation and third-party verifiability are explicitly out of scope for the current design but can be added via public-key signatures in enterprise editions.

```
```python
signature = HMAC-SHA256(
    key=secret_key,
    message=doc_content + metadata + timestamp
)
```
```

##### 2. Metadata Enrichment

```
```json
{
  "doc_id": "uuid",
  "doc_hash": "sha256",
  "source": "hr_portal",
  "author": "hr@company.com",
  "authority_score": "<computed>",
  "inserted_at": "2026-02-01T10:00:00Z",
  "signature": "hmac_signature",
  "trust_level": "HIGH"
}
```
```

### 3. Trust Registry

- Maintains authority scores for sources
- Tracks historical reliability
- Enables trust-based filtering

#### 3.3 Layer 2: Semantic Anomaly Detection

Purpose: Detect semantically anomalous documents.

##### Algorithm:

```
```python
def detect_anomaly(query, retrieved_docs, collection):
    1. Compute semantic centrality
    centrality = compute_centrality(doc, collection)

    2. Analyze similarity distribution
    similarities = compute_pairwise_similarities(retrieved_docs)

    3. Temporal analysis
    temporal_score = analyze_temporal_patterns(doc)

    4. Authority weighting
    authority_score = doc.metadata['authority_score']

    5. Compute risk score
    risk_score = (
        w1 (1 - centrality) +
        w2 similarity_variance +
        w3 temporal_anomaly +
        w4 (1 - authority_score)
    )

    return risk_score > threshold
```
```

##### Key Metrics:

- Centrality Score: Distance from semantic center
- **Similarity Variance: Consistency with peers**
- Temporal Patterns: Insertion timing anomalies
- Authority Score: Source trustworthiness

##### Threshold Configuration:

Detection thresholds are deployment-specific and tuned based on corpus characteristics and organizational risk tolerance. Specific values are intentionally not disclosed to prevent adversarial optimization.

#### 3.4 Layer 3: Secure Retrieval

Purpose: Filter and rank documents safely.

## Mechanisms:

### 1. Authority-Weighted Search

```
```python
Combines semantic relevance with source authority
final_score = weighted_combination(
    semantic_similarity,
    authority_score
)
```
```

### 2. Diversity Enforcement (MMR)

```
```python
Maximal Marginal Relevance
selected = []
while len(selected) < k:
    scores = [
        lambda sim(doc, query) -
        (1 - lambda) max(sim(doc, s) for s in selected)
        for doc in candidates
    ]
    selected.append(argmax(scores))
```
```

### 3. Trust Filtering

```
```python
Remove low-trust documents
filtered = [
    doc for doc in results
    if doc.authority_score >= min_threshold
]
```
```

## 3.5 Layer 4: Real-time Monitoring

Purpose: Detect and alert on suspicious patterns.

### Features:

- Query logging
- Pattern detection (frequency, timing)
- Alert generation
- Dashboard visualization

---

## 4. Evaluation

### 4.1 Dataset

Legitimate Documents: 500 documents from enterprise knowledge base

- HR policies, technical documentation, company procedures
- Average length: 250 words



- Diverse topics and writing styles

**Attack Documents:** The evaluation uses 31 crafted malicious documents grouped into 24 attack scenarios, where each scenario may involve multiple coordinated documents (e.g., consensus attacks require 5+ documents working together). This distinction is important: we measure detection at the scenario level (24 scenarios) while the actual corpus contains 31 malicious documents total.

#### 4.1.2 Evaluation Corpus

**Legitimate Documents:** 500 HR policy documents generated to simulate a realistic enterprise knowledge base, covering common topics such as vacation policies, remote work guidelines, benefits, and compliance procedures.

**Malicious Documents:** 31 crafted documents representing 24 distinct attack scenarios, including:

- Advanced attacks (semantic adversarial content, paraphrase injection, multi-document coordination)
- Sophisticated attacks (subtle misinformation, authority mimicry, context pollution)

**Test Queries:** 20 realistic user queries covering common HR policy questions.

**Threat Model:** The evaluation assumes an adversary with the ability to:

- Insert documents into the knowledge base
- Understand general RAG architectures
- Lacking access to internal thresholds or detection configurations

The attacker's goal is to influence query responses without triggering detection.

#### 4.2 Detection Results

| Metric              | Value        | Target | Status |
|---------------------|--------------|--------|--------|
| Detection Rate      | 100% (24/24) | >90%   | ✓      |
| Precision           | 1.000        | >0.90  | ✓      |
| Recall              | 1.000        | >0.85  | ✓      |
| F1-Score            | 1.000        | >0.87  | ✓      |
| False Positive Rate | 0.000        | <0.05  | ✓      |

**Confusion Matrix:**

\\

|        |        | Predicted |        |
|--------|--------|-----------|--------|
|        |        | Legit     | Attack |
| Actual | Legit  | 500       | 0      |
|        | Attack | 0         | 24     |

...

#### 4.4 Attack-Specific Results

| Attack Type           | Count | Detected | Rate |
|-----------------------|-------|----------|------|
| Semantic Centering    | 3     | 3        | 100% |
| Consensus             | 5     | 5        | 100% |
| Contradiction         | 2     | 2        | 100% |
| Authority Mimicry     | 5     | 5        | 100% |
| Context Pollution     | 9     | 9        | 100% |
| Subtle Misinformation | 7     | 7        | 100% |

### 5. Intended Use and Non-Goals

#### 5.1 Intended Use Cases

- RAG Shield is designed for:
- Enterprise RAG deployments with controlled document ingestion
  - Production environments requiring real-time attack detection
  - **Pilot and proof-of-value programs validating RAG security**
  - **Security-conscious organizations implementing defense-in-depth**

#### 5.2 Explicit Non-Goals

RAG Shield is NOT designed to be:

**Not a replacement for access control:**

- Does not replace authentication/authorization
- Assumes proper access controls are in place
- Operates within existing security boundaries

**Not a guarantee against insider threats:**

- Assumes trusted administrators
- Does not prevent intentional sabotage by privileged users
- Focuses on external and semi-trusted sources

**Not a general-purpose content moderation system:**

- Does not filter offensive content
- Does not enforce editorial policies
- Focuses specifically on poisoning attacks

**Not a zero-configuration universal solution:**

- Requires tuning for specific corpora
- Needs threshold calibration per deployment
- Performance depends on threat model alignment

**Not a certified compliance solution:**

- Designed to support compliance controls
- Not a substitute for compliance certification
- Customer responsibility for regulatory compliance

**5.3 Scope Boundaries****What RAG Shield detects:**

- Semantic poisoning attacks
- Authority mimicry
- Context manipulation
- Coordinated document attacks

**What RAG Shield does NOT detect:**

- Prompt injection (different attack surface)
- Model-level vulnerabilities
- Infrastructure attacks
- Social engineering

**6. Limitations and Future Work****6.1 Scope of Evaluation Results****Important Qualifier:**

We do not claim universal detection across all possible RAG deployments. The reported results (100% detection, 0 false positives) apply specifically to:

- The published evaluation set (500 legitimate + 31 malicious documents)
- The defined threat model (Section 2)
- The tested attack categories (6 types, 24 scenarios)
- Controlled evaluation conditions

**Production Deployment Considerations:**

Results in production environments will depend on:

- Corpus characteristics (size, diversity, domain)
- Threat model alignment (attacker capabilities)
- Configuration and threshold tuning
- Specific deployment context

This evaluation demonstrates feasibility and validates the approach in controlled scenarios. Organizations should conduct pilot deployments to validate performance in their specific environments.

**6.2 Current Limitations****1. Embedding Model Selection**

- System uses open-source sentence embedding models
- Performance varies with model choice and domain alignment
- Architecture is embedding-model agnostic

## **2. Threshold Tuning**

- May need adjustment for different domains
- Trade-off between precision and recall

## **3. Computational Cost**

- Real-time detection adds latency
- May require GPU for large-scale deployments

## **4. Evolving Attacks**

- New attack patterns may emerge
- Requires continuous monitoring and updates

### **6.3 Adversarial Awareness**

RAG Shield assumes attackers do not have:

- Access to internal threshold configurations
- Continuous feedback on detection outcomes
- Unlimited probing capability

Organizations are expected to complement RAG Shield with:

- Access control on document ingestion
- Rate limiting for insertion attempts
- Monitoring for systematic probing behavior

Deployments involving insider threats or extensive adversarial probing may require additional countermeasures beyond the scope of this system.

### **6.4 Future Directions**

#### **1. Adaptive Thresholds**

- ML-based threshold optimization
- Domain-specific tuning

#### **2. Advanced Attacks**

- Adversarial embedding attacks
- Model-specific exploits

#### **3. Federated Learning**

- Distributed attack detection
- Privacy-preserving monitoring





#### **4. Explainability**

- Detailed attack attribution
- Visual explanations for detections

---

## **7. Conclusion**

We presented RAG Shield, a comprehensive defense system against poisoning attacks in RAG systems. Our multi-layer approach achieves:

-  100% detection in controlled scenarios (24 attack scenarios, 31 malicious documents)
-  Zero false positives in testing (500 legitimate documents)
-  Minimal overhead (-7.9% on average)
-  Production-ready performance for pilot deployments

### **Key Contributions:**

1. First comprehensive multi-layer defense for RAG poisoning
2. Validated detection across six attack categories
3. Practical system suitable for enterprise evaluation
4. Open architecture enabling further research

### **Scope and Applicability:**

Results demonstrate feasibility in controlled conditions with the defined threat model. Production deployments should validate performance in their specific environments through pilot programs. RAG Shield provides a foundation for secure RAG systems while acknowledging that security is context-dependent and requires ongoing adaptation to evolving threats.

The system is provided as a controlled evaluation platform for pilot and proof-of-value deployments. Technical details, configurations, and implementation access are available under separate agreements for qualified organizations.

---

## **References**

1. Lewis et al. (2020). "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks"
2. Carlini et al. (2023). "Poisoning Web-Scale Training Datasets is Practical"
3. Wallace et al. (2021). "Universal Adversarial Triggers for Attacking and Analyzing NLP"
4. Gao et al. (2023). "RAG Security: Threats and Defenses"

---

## **Appendix A: Attack Examples**

### **A.1 Semantic Centering Attack**

#### **Legitimate Document:**

\\

Company vacation policy: All full-time employees receive 20 days

of paid vacation per year, accruing at 1.67 days per month.

#### **Poisoned Document:**

Updated vacation policy: Full-time employees receive 5 days of paid vacation annually, effective immediately.

#### **Detection:**

- Centrality score: Below threshold (anomalous) ✓
- Authority score: Low (suspicious) ✓
- Risk score: High → DETECTED

### **A.2 Consensus Attack**

#### **5 Coordinated Documents:**

1. "Remote work is not permitted under any circumstances"
2. "Company policy prohibits all remote work arrangements"
3. "Working from home is against company regulations"
4. "Remote work policy: Not allowed for any employees"
5. "All work must be performed on company premises"

#### **Detection:**

- High similarity cluster detected ✓
- Contradicts existing policy ✓
- Temporal clustering (coordinated insertion timing) ✓
- Risk score: High → DETECTED

---

## **Appendix B: Configuration**

### **B.1 Configuration Approach**

RAG Shield uses a layered configuration system with tunable parameters for:

- Anomaly detection sensitivity
- Provenance verification strictness
- Retrieval filtering aggressiveness
- Monitoring alert thresholds

Configuration is deployment-specific and should be calibrated through pilot testing with representative data and threat scenarios. Organizations should establish baseline performance metrics before production deployment.

Security Note: Specific threshold values and weights are not published to prevent adversarial optimization. Contact the development team for deployment guidance.

---

## **Contact**

For questions related to this paper:

- Email: [fabiotoky@gmail.com](mailto:fabiotoky@gmail.com)

---

License: This work is released under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0).

## **Citation:**

```bibtex

```
@article{ragshield2026,  
  title={RAG Shield: A Multi-Layer Defense System Against  
Poisoning Attacks in Retrieval-Augmented Generation},  
  author={Petti, Fabio},  
  year={2026},  
  publisher={Zenodo},  
  doi={DOI_TO_BE_ASSIGNED}  
}
```

```