To Encrypt or Not to Encrypt? Evaluating the Trade-offs of Distance-Preserving Encryption in RAG Vector Databases

Nathaniel Jonathan Rusli - 13523013[1,2]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13523013@std.stei.itb.ac.id, [2]omgitsnathaniels@gmail.com

***Abstract*—Retrieval-Augmented Generation (RAG) often relies on cloud vector databases, exposing embeddings to severe privacy risks. This paper evaluates the utility-security trade-offs of securing these databases using Distance-Preserving Encryption (DPE). We conducted an in-memory micro-benchmark comparing raw ASPE, ASPE with dummy dimensions, Distance-Comparison-Preserving Encryption (DCPE), and Fully Homomorphic Encryption (FHE). Using the FiQA dataset, we measured Recall@10, query latency, DRAM overhead, and Known-Plaintext Attack (KPA) resistance. Results indicate that while FHE provides absolute security, its massive memory bloat and 1.8 x 10^6-fold latency increase render it impractical for real-time search. Conversely, exact linear schemes (Raw ASPE and ASPE with dummy dimensions) are highly efficient but fail catastrophically against KPA, allowing full plaintext reconstruction. Ultimately, DCPE is the only viable compromise, trading a negligible 1.08% recall loss for a twelve-order-of-magnitude improvement in reconstruction resistance. To securely balance retrieval fidelity and confidentiality, architects should adopt perturbation-based protocols.**

***Keywords*—Distance-preserving encryption (DPE), vector databases, vector embeddings, performance benchmark, privacy-preserving machine learning (PPML), retrieval-augmented generation (RAG)**

I. Introduction

Retrieval-Augmented Generation (RAG), as originated in [1], has emerged as a core element for modern Agentic AI architectures, effectively solving inherent limitations of Large Language Models (LLMs). Relying exclusively on parametric memory often leaves LLMs prone to factual hallucinations and catastrophic forgetting when adapting to new domains. RAG resolves these flaws by dynamically grounding generative models in external, non-parametric knowledge bases. This hybrid approach enables continuous, real-time data updates without the hefty costs of model retraining, while significantly enhancing the specificity, accuracy, and provenance of the generated responses.

However, the computational shift towards utilizing vector databases for similarity search introduces a critical attack surface, particularly as cloud-based RAG emerges as the predominant deployment paradigm [2]. Major cloud providers, including Amazon, Google, and Alibaba Cloud, now offer native capabilities for seamlessly integrating enterprise document repositories, causing sensitive documents and retrieval indices to reside persistently on public cloud servers [3]. The vulnerability of such cloud-hosted data at scale has been starkly demonstrated by a series of massive data breaches since 2024, most notably the Mother of All Breaches (MOAB) which exposed over 26 billion credential records, alongside incidents targeting major technology vendors and institutions [4]. Storing these dense vector embeddings in plaintext without strong cryptographic protection poses a systemic and ongoing privacy risk.

To mitigate these escalating vulnerabilities, recent literature has actively explored the paradigm of Privacy-Preserving RAG (ppRAG). Various end-to-end frameworks have been proposed, utilizing cryptographic protocols such as Private Information Retrieval (PIR), Trusted Execution Environments (TEE), efficient arbitrary Top-k retrieval mechanisms, and strict multi-tenant access controls [2], [5]-[7]. Collectively, these studies demonstrate that robust data isolation and query privacy can be achieved while maintaining generation quality, typically at the cost of manageable trade-offs in communication bandwidth and computational latency. Although there have been significant efforts to secure the broader RAG pipeline through these architectural and hardware-based solutions, securing the high-dimensional embeddings directly at rest remains the fundamental line of defense, positioning Distance-Preserving Encryption (DPE) as the main cryptographic guardrail for vector databases [8].

Motivated by this necessity, this study aims to address the critical dilemma faced by system architects: “To encrypt or not to encrypt?”. As encrypted vector databases increasingly become the default prerequisite for enterprise security, navigating the complex landscape of cryptographic implementations requires solid empirical evidence. This paper provides a comprehensive comparative benchmark of representative DPE algorithms to guide optimal implementation choices. By systematically evaluating these methods across four vital dimensions: (i) retrieval utility (Recall@K), (ii)

computational efficiency (Query Latency), (iii) spatial footprint (DRAM Overhead), and (iv) empirical security resilience against Known Plaintext Attacks (MSE KPA). This study offers an objective framework for balancing absolute privacy with real-time semantic search performance.

II. Preliminaries

A. Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) is an architecture designed to enhance the generative capabilities of Large Language Models (LLMs) and mitigate common issues such as factual hallucinations. It achieves this by combining a pre-trained parametric memory (the LLM's internal parameters) with an explicit, non-parametric memory acting as an external knowledge base. By utilizing RAG, systems can dynamically integrate new, up-to-date information without necessitating the computationally expensive retraining or fine-tuning of the entire model [1].

A typical RAG system relies on four principal components:

1) Embedder Function: A mapping function that converts textual information into a high dimensional embedding space.

2) Knowledge Base: Commonly referred to as a vector store or database, which stores texts and their embedded representations.

3) Similarity Function: A mathematical metric (e.g. cosine similarity) to evaluate distance between pairs of embedded text vectors.

4) Generative Model: Typically an LLM that's responsible for producing the final output text based on input prompts and retrieved contexts.

The retrieval components consist of two primary phases: retrieval and generation. The retrieval phase aims to extract relevant information from various external knowledge sources comprising the indexing and searching phases. The generation components utilizes the retrieved documents' information and query to formulate a contextual response with prompting and inference phase.

B. Vector Embedding

Vector embeddings are dense, high-dimensional numerical representations of data designed to capture deep semantic meaning and context. The evolution of text representation in Natural Language Processing (NLP) has advanced significantly to overcome the limitations of early methodologies. Initial techniques, such as One-Hot Encoding (OHE), represented words as sparse, isolated vectors that failed to capture linguistic relationships and suffered from the curse of dimensionality. This primitive approach gradually evolved into dense word embeddings and eventually into sophisticated sentence embeddings powered by pre-trained transformer models. These modern encoders are capable of mapping entire sentences or document chunks into a unified high-dimensional embedding space, capturing the contextual nuance of the entire sequence rather than just isolated words [9].

Sentence embeddings are fundamentally crucial for the success of RAG architectures. Instead of relying on rigid exact-keyword matching, a RAG system utilizes an embedder function to convert both the user's query and the knowledge base documents into embedding vectors. Because texts with similar meanings are projected closely together within this continuous vector space, the system can efficiently utilize mathematical similarity functions (e.g. cosine similarity or inner product) to identify and retrieve the most relevant context. This intrinsic reliance on spatial distance highlights exactly why securing the embedding space poses a unique challenge. Any applied cryptographic method, such as Distance-Preserving Encryption (DPE), must secure the underlying text without destroying the geometric proximity that makes semantic search possible.

C. Vector Database

Vector Databases (VDBs) manage high-dimensional vectors using semantic similarity, serving as the essential non-parametric knowledge base in RAG architectures to mitigate LLM hallucinations. To retrieve the top-K relevant contexts, VDBs compute distance metrics. Since Exact Nearest Neighbor Search (NNS) scales poorly with a time complexity of $O(n)$, modern VDBs employ Approximate Nearest Neighbor (ANN) algorithms (including quantization, hashing, and tree-based methods) to achieve sub-linear query complexity at the cost of marginal accuracy [10].

Currently, graph-based algorithms, particularly Hierarchical Navigable Small World (HNSW), represent the state-of-the-art. HNSW utilizes a multi-layered proximity graph for coarse-to-fine routing, achieving an $O(log\ N)$ search complexity ideal for low-latency RAG deployments. Crucially, because ANN routing relies heavily on evaluating precise geometric distances, introducing cryptographic noise during database encryption inevitably distorts the index topology. This spatial distortion directly drives the complex utility-security trade-offs evaluated in this study [10].

D. Distance-Preserving Encryption (DPE)

Distance-Preserving Encryption (DPE) is a class of cryptographic techniques that enables an untrusted server to evaluate the geometric distance or semantic similarity between vectors directly in the ciphertext domain [11]. In the context of RAG, DPE is fundamentally crucial for executing Secure Similarity Retrieval (SSR) without exposing the underlying plaintext embeddings to the cloud provider. However, standard SSR schemes face significant security challenges, as their inherent preservation of exact mathematical distances makes them susceptible to topological analysis and practical attack models [12].

1) Asymmetric Scalar-Product-Preserving Encryption (ASPE): This algorithm is designed to preserve the inner product (scalar product) between two vectors, which is the foundational metric for cosine similarity and dot product search in vector databases. In vanilla ASPE, the data owner generates a secret key composed of an invertible matrix *M*. A document vector *p* and a query vector *q* are encrypted asymmetrically:

$$p' = M^T p \tag{1}$$

$$q' = M^{-1} p \tag{2}$$

When the server computes the inner product of the ciphertexts during retrieval, the secret matrix mathematically cancels out, returning the exact plaintext similarity without requiring decryption:

$$(p')^T q' = (M^T p)^T (M^{-1} q) = p^T M M^{-1} q = p^T q \tag{3}$$

Because vanilla ASPE preserves the exact inner product, the spatial topology of the dataset is fully exposed, making it highly vulnerable to Known Plaintext Attacks (KPA) and Ciphertext-Only Attacks (COA). To obscure this topology and mitigate inversion attacks, the vectors are deliberately expanded with *k* artificial "dummy" dimensions before encryption. The original *d*-dimensional document and query vectors are extended into $\hat{p}$ and $\hat{q}$ respectively:

$$\hat{p} = [p_1, \ldots, p_d, r_1, \ldots, r_k]^T \tag{4}$$

$$\hat{q} = [q_1, \ldots, q_d, s_1, \ldots, s_k]^T \tag{5}$$

where *r* and *s* are artificially generated random variables. The vectors are then encrypted using an expanded (*d* + *k*) x (*d* + *k*) secret matrix. The new encrypted inner product computed by the server becomes:

$$\hat{p}'^T \hat{q}' = p^T q + \sum_{i=1}^{k} r_i s_i \tag{6}$$

The sigma term acts as a controlled cryptographic noise. This noise severely distorts the exact distance metrics available to an attacker, masking the true data distribution. However, this artificial expansion inevitably introduces the core trade-off evaluated in this study i.e. expanding the vector size.

2) Distance-Comparison-Preserving Encryption (DCPE): This algorithm represents an evolutionary step in approximate DPE. Utilizing mechanisms such as Scale-and-Perturb (SAP), DCPE diverges from preserving the absolute distance and instead preserves only the relative distance comparisons between vectors. Mathematically, the evaluation distance between a query *q* and a document *p* in the ciphertext domain (D_{enc}) is modeled as:

$$D_{enc}(p', q') = \lambda . ||p - q||^2 + \delta \tag{7}$$

where $\lambda > 0$ is a secret scaling factor that distorts the original magnitude, and δ is a bounded random perturbation injected to mask the exact distance. By normalizing the plaintext distribution and introducing these mathematically bounded perturbations, DCPE secures the index against approximate frequency-finding and membership inference attacks, making it highly optimized for Approximate Nearest Neighbor (ANN) search.

E. Fully Homomorphic Encryption (FHE)

This algorithm provides absolute semantic security (e.g., IND-CPA) without leaking any topological information, allowing for the benchmark of a security-heavy algorithm (FHE) against latency-light DPE algorithms. FHE allows the server to compute the inner product entirely under encryption using homomorphic addition ($\oplus$) and multiplication ($\otimes$) directly over the ciphertexts. For vectors *p* and *q* of *d*-dimensions encrypted as *E*(*p*) and *E*(*q*), the server computes the encrypted inner product as:

$$E(p^T q) = \sum_{i=1}^{d} (E(p_i) \otimes E(q_i)) \tag{8}$$

The server never observes the relative distance or similarity score. It only generates a new ciphertext that when decrypted by the data owner using the secret key (SK), yields the original inner product:

$$Dec_{SK}\left(\sum_{i=1}^{d} (E(p_i) \otimes E(q_i)) \right) \approx (p^T q) \tag{9}$$

While it eliminates the topological vulnerabilities inherent to ASPE and DCPE, the immense computational complexity and massive ciphertext expansion associated with FHE currently render it impractical for low-latency, real-time RAG deployments, highlighting the necessity for efficient DPE alternatives.

F. System and Retrieval Evaluation Metrics

As the development of RAG systems accelerates and grows in complexity, the necessity for comprehensive evaluation methodologies has become paramount. This evolving landscape requires comprehensive metrics capable of accurately assessing the dynamic relationship between retrieval precision and generative quality [12]. While numerous RAG evaluation frameworks exist, this study focuses on three critical metrics:

1) Recall@K: Measures retrieval accuracy by calculating the proportion of relevant documents successfully retrieved within the top *K* search results.

$$Recall@K = \frac{|Relevant\ Documents \cap Top{-}K\ Results|}{|Total\ Relevant\ Documents|} \tag{10}$$

2) Query Latency: Measures the end-to-end processing time required to execute a single retrieval

$$Latency = t_{response} - t_{request} \quad (11)$$

3) DRAM Overhead: Quantifies the peak memory (RAM) consumed by the vector database to store and search the encrypted index.

$$\triangle DRAM = Memory_{encrypted} - Memory_{plaintext} \quad (12)$$

4) Empirical Security (Reconstruction Error): Evaluates the cryptographic resilience against topological leakage and Known Plaintext Attack (KPA) by simulating an adversary attempting to estimate the secret matrix using a subset of known pairs. The security is measured by the Mean Squared Error (MSE) between the original plaintext vectors (p_i) and the adversary's reconstructed vectors ($\hat{p}_i$), where a higher MSE indicates superior obfuscation and security.

$$MSE_{KPA} = \frac{1}{N}\sum_{i=1}^{N} \left|\left|p_i - \hat{p}_i\right|\right|^2 \quad (13)$$

III. Methodologies

A. System Architecture and Technology Stack

To isolate the intrinsic cryptographic overhead from confounding vector database artifacts (e.g., disk I/O, ANN index heuristics, and concurrency control), a pure in-memory micro-benchmark is implemented using Python 3.11 and NumPy. All five retrieval pipelines: (i) Plaintext (Raw), (ii) Vanilla ASPE, (iii) ASPE + Dummy, (iv) DCPE (Scale-and-Perturb), and (v) FHE (CKKS), share an identical retrieval substrate executing exact Maximum Inner Product Search (MIPS).

This ensures that any deviations in latency, memory, or retrieval fidelity are causally attributable to the cryptographic transformations alone. While the DPE-class schemes process the full workload, the computationally prohibitive FHE pipeline is evaluated on a representative sub-sample (5 queries against 50 documents). FHE cost metrics are linearly extrapolated, while its retrieval accuracy is assumed identical to the plaintext baseline due to its precision-preserving nature.

B. Dataset Preparation

The benchmark utilizes the FiQA (Financial Question Answering) dataset from the Massive Text Embedding Benchmark (MTEB). Financial corpora are deliberately selected as they present realistic confidentiality threats under embedding-inversion attacks in RAG deployments. The corpus is sub-sampled to 5,000 documents and 500 queries, strictly preserving ground-truth relevance.

Texts are encoded using the all-MiniLM-L6-v2 sentence-transformer Hugging Face model into dense vectors of dimensionality d = 384. All embeddings are L2-normalized, equating the inner product to cosine similarity and reducing MIPS to nearest-neighbor search on a unit hypersphere.

C. Cryptographic Implementation Setup

Let p, q as real number vectors with d = 384 denote plaintext document and query embeddings.

1) Vanilla ASPE: Encrypts via an invertible secret matrix M (real number 384 x 384 matrix), where $p' = M^T p$ and $q' = M^{-1} q$. This exactly preserves the inner product ($\langle p', q' \rangle = p^T q$), leaving the spatial topology fully exposed with zero retrieval distortion.

2) ASPE + Dummy: Augments vectors with k = 50 zero-mean Gaussian dummy dimensions (σ = 0.05) to obfuscate score distributions, expanding dimensionality to d = 434. Encryption utilizes a 434 x 434 secret matrix. Although the additive perturbation masks exact score values, the transformation remains an invertible linear bijection.

3) DCPE (Scale-and-Perturb): Encrypts documents as $c = \lambda p + \delta$, where $\lambda > 0$ is a secret scalar and δ is bounded Gaussian noise (norm-clipped to 3σ). Queries are uniformly scaled by λ. The encrypted score $\langle c, \lambda p \rangle$ masks exact similarities while probabilistically preserving relative ordering, effectively breaking the deterministic linear map.

4) FHE: Implemented via the TenSEAL library using a Ring-Learning-With-Errors (RLWE) context. Parameterized with a polynomial modulus degree of 8,192 and coefficient-modulus bit sizes of [60, 40, 40, 60], CKKS provides absolute semantic security, computing approximate inner products homomorphically without linear vulnerabilities.

D. Environment and Evaluation Procedure

Experiments are executed on a single workstation (Apple M1 Pro, 8-core CPU, 16 GB RAM) running macOS. Hardware acceleration and external databases are explicitly excluded. The framework evaluates four metrics: (i) Recall@K (with k = 5 and k = 10), query latency (measured with Python's time.perf_counter(), (iii) peak DRAM footprint, and (iv) empirical security (with 1000 known plaintext-ciphertext pairs).

IV. Results

Table I below summarizes the four metrics across all five encryption schemes.

TABLE I
UTILITY, COST, AND SECURITY ACROSS ALL ENCRYPTION SCHEMES

Scheme	*Recall@10*	*Latency (ms/query)*	*DRAM (MB)*	*KPA MSE*
Raw	68.40%	0.0434	7.32	5.04×10^{-32}
Vanilla ASPE	68.40%	0.0431	7.32	5.17×10^{-16}
ASPE + Dummy	67.74%	0.0436	8.28	6.36×10^{-16}
DCPE	67.32%	0.0429	7.32	3.57×10^{-4}
FHE	68.40%	79,019.17	1,593.08	∞

A. Retrieval Accuracy (Recall@10)

The plaintext baseline achieves a retrieval utility of 68.40% Recall@10. Vanilla ASPE and FHE (CKKS) reproduce this accuracy exactly (68.40%), confirming that precision-preserving transformations incur zero utility cost. Conversely, noise-injecting schemes exhibit a slight, monotonic degradation proportional to the perturbation magnitude i.e. ASPE + Dummy drops to 67.74% (a 0.66% absolute reduction), while DCPE falls to 67.32% (a 1.08% reduction). Despite this loss, the utility penalty remains modest, preserving the semantic ranking sufficiently for practical RAG applications.

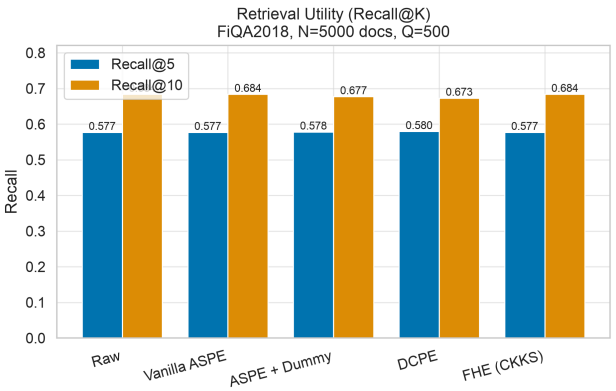

Fig. 1. Recall@K across all encryption algorithms

B. Computational Efficiency (Query Latency)

Query latency remains effectively invariant across all DPE-class configurations. Raw (0.0434 ms), Vanilla ASPE (0.0431 ms), DCPE (0.0429 ms), and ASPE + Dummy (0.0436 ms) all execute at approximately 0.043 ms per query. This demonstrates that scalar- and matrix-based DPE introduces negligible algebraic overhead. In stark contrast, FHE exhibits a catastrophic computational penalty, requiring 79,019.17 ms (≈ 79 seconds) per query. This 1.8 x 10^6-fold latency increase renders FHE strictly impractical for real-time similarity retrieval.

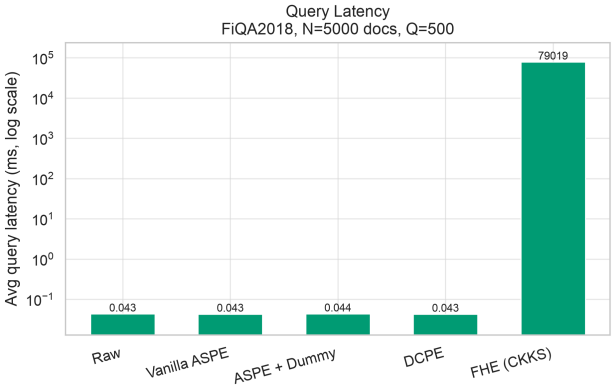

Fig. 2. Query latency across all encryption algorithms

C. Memory Overhead (DRAM)

The plaintext corpus, Vanilla ASPE, and DCPE all maintain the exact 384-dimensional storage footprint of 7.32 MB. Dimensional padding in ASPE + Dummy (k = 50) expands the footprint by 13.0% to 8.28 MB, scaling strictly linearly with the added dimensions. Meanwhile, the ciphertext expansion inherent to RLWE-based encryption causes the FHE index to explode to 1,593.08 MB (≈ 1.56 GB), imposing a severe 218x storage bloat over the baseline.

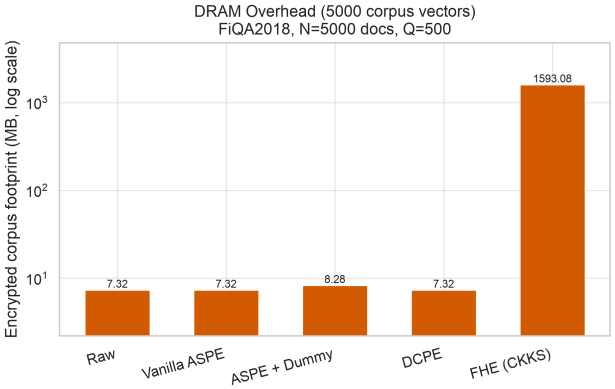

Fig. 3. DRAM overhead across all encryption algorithms

D. Practical Security (KPA Reconstruction Error)

The simulated Known-Plaintext Attack exposes the critical vulnerability of exact linear mappings. While Raw is trivially invertible (MSE ≈ 5.04×10^{-32}), Vanilla ASPE is identically compromised (MSE ≈ 5.17×10^{-16}), as the least-squares estimator effortlessly recovers the secret matrix to floating-point precision. Crucially, ASPE + Dummy fails to mitigate this threat (MSE ≈ 6.36×10^{-16}). Despite perturbing the top-k scores, its augmented encryption remains an exact linear bijection, allowing complete plaintext reconstruction. DCPE is the only DPE scheme to resist the attack, achieving an MSE of 3.57×10^{-4}, twelve orders of magnitude higher due to its irreducible, per-vector random perturbation. As expected, FHE provides absolute semantic security, remaining mathematically immune to linear inversion (MSE = ∞).

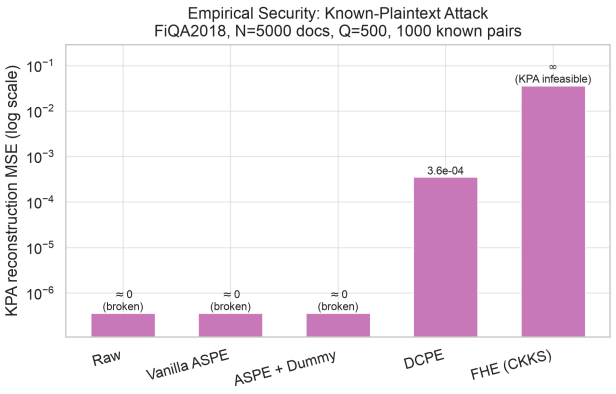

Fig. 4. KPA reconstruction MSE across all encryption algorithms

V. Discussions

A. Geometric Distortion & the Utility Cost of Noise

The accuracy degradation in noise-injecting schemes is caused by the disruption of fine-grained distance margins. In top-k MIPS, semantically similar documents form dense clusters with narrow score margins. While Vanilla ASPE preserves these precise margins, ASPE + Dummy (additive noise) and DCPE (multiplicative and additive noise) superimpose stochastic perturbations. This cryptographic noise "clouds" the relative ordering, occasionally displacing borderline relevant documents from the top-k results. Although the recall penalty is minimal ($\leq$ 1.08%) due to calibrated noise boundaries, it confirms a fundamental DPE trade-off: obfuscating score geometry inevitably compromises retrieval fidelity.

B. Architectural Viability (DPE-FHE Dilemma)

The empirical results answer the paper's core dilemma by illustrating a stark cost-security frontier. FHE provides absolute semantic security but imposes an intolerable 1.8 x 10^6-fold latency penalty (≈ 79 seconds per query) and massive memory bloat, rendering it entirely unviable for interactive RAG. This operational bottleneck justifies the necessity of DPE.

However, DPE's viability is strictly conditional on the threat model. Both Vanilla ASPE and ASPE + Dummy are catastrophically vulnerable to linear Known-Plaintext Attacks (KPA) and offer no practical privacy if the adversary obtains known pairs. DCPE emerges as the only viable compromise, sacrificing roughly 1% recall to achieve a twelve-order-of-magnitude improvement in reconstruction resistance. Therefore, perturbation-based schemes like DCPE are strongly recommended for production environments facing KPA risks, while FHE must be reserved for offline, latency-insensitive workloads.

C. Limitations and Future Gaps

This study evaluates foundational DPE archetypes using an in-memory algebraic micro-benchmark, presenting three principal limitations:

1) System Abstraction: The benchmark isolates algebraic overhead but abstracts away production vector database artifacts, such as Approximate Nearest Neighbor (ANN) indexing heuristics and disk I/O, which may alter performance at scale.

2) Threat Model Scope: The security analysis is confined to linear KPA. It does not evaluate non-linear inversion or machine learning-based attacks trained on auxiliary data.

3) Algorithmic Scope: This work benchmarks basic DPE primitives and excludes emerging state-of-the-art protocols designed to selectively bound leakage, such as CAPRISE, Distance Comparison Encryption (DCE), Salty Embeddings, and Secure Binary Embeddings.

VI. Conclusion

This study evaluated the operational and cryptographic trade-offs of Distance-Preserving Encryption (DPE) and Fully Homomorphic Encryption (FHE) for its absolute semantic security in securing cloud-based RAG vector databases. Through an isolated micro-benchmark, we demonstrated that while FHE provides absolute semantic security, its extreme latency and memory overheads render it currently impractical for real-time similarity search. Conversely, foundational DPE schemes such as Vanilla ASPE and ASPE + Dummy offer near-plaintext efficiency and high retrieval utility, but fail catastrophically against linear Known-Plaintext Attacks (KPA), exposing the underlying vectors to full reconstruction.

Ultimately, DCPE emerged as the only viable compromise among the tested primitives, trading a negligible utility penalty (≈ 1.08% recall loss) for a twelve-order-of-magnitude increase in reconstruction resistance. Answering the main question of this paper i.e. "To encrypt or not to encrypt?", RAG system architects should encrypt their vector databases, but abandon exact linear encryptions in favor of perturbation-based or advanced selective-leakage protocols to ensure that the critical balance between retrieval fidelity and data confidentiality is strictly maintained.

VII. Acknowledgment

The author expresses heartfelt gratitude to God Almighty for granting the strength, perseverance, and opportunity to successfully complete this paper. The author also wishes to express profound gratitude to Prof. Dr. Ir. Rinaldi, M.T., the lecturer for the II4021 Cryptography course, for his unwavering guidance and inspiration throughout his time teaching the students.

VIII. Appendix

The complete source code used for the benchmark experiment is available on GitHub. Access the code repository here:
https://github.com/0xNathaniel/vector-database-encryption

References

[1] P. Lewis, et al., "Retrieval-Augmented Generation for knowledge-intensive NLP tasks," *arXiv*, arXiv:2005.11401, 2020. [Online]. Available: https://arxiv.org/abs/2005.11401.

[2] Y. Cheng, L. Zhang, J. Wang, M. Yuan, and Y. Yao, "Remoterag: A privacy-preserving llm cloud rag service," in Findings of the Association for Computational Linguistics: ACL 2025, 2025, pp. 3820–3837.

[3] Amazon Web Services, "Knowledge bases for amazon bedrock," https://docs.aws.amazon.com/bedrock/latest/userguide/knowledge-base.html, Amazon.com, Inc., 2026.

[4] C. Team, "Mother of all breaches reveals 26 billion records: what we know so far," https://cybernews.com/security/billions-passwords-credentials-leaked-mother-of-all-breaches/ s, Cybernews, 2024.

[5] Z. Li, et al., "PRAG: End-to-End Privacy-Preserving Retrieval-Augmented Generation," *arXiv*, arXiv:2604.26525, 2026. [Online]. Available: https://arxiv.org/abs/2604.26525.

[6] Y. Ming, et al., "P^2 RAG: Efficient Privacy-Preserving RAG Service Supporting Arbitrary Top-*k* Retrieval" *arXiv*, arXiv:2603.14778, 2026. [Online]. Available: https://arxiv.org/abs/2603.14778.

[7] P. Zhou, et al., "Privacy-Aware RAG: Secure and Isolated Knowledge Retrieval," *arXiv*, arXiv:2503.15548v1, 2025. [Online]. Available: https://arxiv.org/html/2503.15548v1.

[8] Y. Zheng, R. Lu, Y. Guan, J. Shao, and H. Zhu, "Achieving efficient and privacy-preserving exact set similarity search over encrypted data," IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 2, pp. 1090–1103, 2020.

[9] C. Zhang, *et al.*, "From Word Vectors to Multimodal Embeddings: Techniques, Applications, and Future Directions For Large Language Models," *arXiv*, arXiv:2411.05036v3, 2025. [Online]. Available: hhttps://arxiv.org/html/2411.05036v3.

[10] L. Ma, et al., "A Comprehensive Survey on Vector Database: Storage and Retrieval Technique, Challenge," *arXiv*, arXiv:2310.11703, 2023. [Online]. Available: https://arxiv.org/pdf/2310.11703.

[11] G. Fuchsbauer, R. Ghosal, N. Hauke, and A. O'Neill, "Approximate Distance-Comparison-Preserving Symmetric Encryption," in Security and Cryptography for Networks, C. Galdi and V. Kolesnikov, Eds., Cham, Switzerland: Springer International Publishing, 2022, pp. 111–134.

[12] S. Barnett, et al., "Seven Failure Points When Engineering a Retrieval Augmented Generation System," *arXiv*, arXiv:2401.05856, 2024. [Online]. Available: https://arxiv.org/abs/2401.05856.

Statement

I hereby declare that this paper is my own work, not a paraphrase or translation of someone else's paper, and not plagiarism.

Bandung, 18 June 2026

Nathaniel Jonathan Rusli
13523013