A Reproducible Evaluation of Confidential Computing for AI Inference in Public Clouds

Leonardo S. Foschine[1], **Bruno M. P. Takazono**[1], **Victor Pachano**[1], **Alexandre Braga**[1]

[1]CPQD, Rua Dr. Ricardo Benetton Martins, 1.000, Campinas - SP - Brazil

{lfoschine, ambraga}@cpqd.com.br, {brunotakazono, pachanovictor94}@gmail.com

Abstract. *Confidential Computing (CC) protects data in use through hardware isolation, memory encryption, and attestation. This paper presents a reproducible case study of CC in public clouds using a CPU-only document-embedding benchmark for AI inference, representative of services that process sensitive or personal text data. We compare baseline VMs and Confidential VMs on Google Cloud with AMD SEV-SNP and Intel TDX. The experimental infrastructure records metadata, hashes, logs, and benchmark outputs to support reproducibility and auditability. Results show that enabling CC did not introduce systematic performance degradation in the evaluated benchmark. Observed gains are interpreted cautiously due to public-cloud confounders such as I/O paths, host placement, scheduling, and storage behavior.*

1. Introduction

Modern cloud applications routinely rely on cryptography to protect data at rest and in transit. However, data must still be processed in usable form by the CPU, leaving a critical exposure window known as data in use. In public cloud environments, this exposure is particularly relevant because privileged infrastructure layers, such as the hypervisor, firmware, and cloud operators, are outside the tenant's direct control [Russinovich et al. 2021, Kaplan 2023].

This issue is especially relevant for AI inference services that process proprietary corpora, enterprise knowledge bases, or personal data. The same concern appears in public-sector AI initiatives that aim to personalize citizen-facing services from integrated data sources while preserving compliance with data-protection requirements. In these scenarios, sensitive inputs and intermediate representations remain exposed while the model is executing, even if storage and network channels are encrypted.

Confidential Computing (CC) addresses this gap by combining hardware-based isolation, memory encryption, and attestation mechanisms. Instead of only protecting storage or communication channels, CC protects the execution environment itself and can strengthen data protection during AI inference. Confidential Virtual Machines (CVMs), such as those based on AMD SEV-SNP and Intel TDX, are especially attractive for practical adoption because they can run existing software stacks with limited or no application refactoring [Misono et al. 2024, Confidential Computing Consortium 2022].

Despite these benefits, adopting CC in production still depends on understanding its operational cost. The performance impact of hardware-assisted memory encryption and VM isolation is workload-dependent and may be affected by cloud-specific factors such as host placement, scheduling, I/O paths, storage configuration, and platform heterogeneity [Liu et al. 2025, Misono et al. 2024]. Therefore, reproducible and auditable benchmarking is necessary to distinguish the cost of protection from ordinary public-cloud variability.

This paper presents a proof of concept for evaluating CC in public clouds. We compare baseline VMs and CVMs on Google Cloud using AMD SEV-SNP and Intel TDX. As a case study, we

evaluate a CPU-only document-embedding benchmark, varying input length and batch size. This benchmark represents a simple AI inference operation that appears in retrieval-oriented text services. The experimental infrastructure records execution metadata, environment characteristics, hashes, logs, and benchmark outputs to support reproducibility and auditability.

The contributions of this work are fourfold: (i) a practical evaluation of AMD SEV-SNP and Intel TDX for a document-embedding inference benchmark representative of sensitive text-processing services; (ii) a reproducible and auditable benchmarking infrastructure based on manifests, hashes, logs, and environment metadata; (iii) an empirical analysis of throughput, median latency, tail latency, and stability for a sensitive text-processing workload; and (iv) a discussion of public-cloud confounders and the limits of attributing observed performance gains directly to CC.

2. Background, Threat Model, and Related Work

Interpreting the evaluation requires distinguishing the protection offered by Confidential Computing from the assumptions and limitations of the benchmark. The following subsections introduce the relevant CVM mechanisms, define the threat model, and relate this work to prior studies on confidential virtual machines.

2.1. Confidential Computing as Hardware-Assisted Cryptography

Confidential Computing extends cryptographic protection to data in use. While disk encryption and transport encryption protect data at rest and in transit, CC uses hardware mechanisms to protect code and data while they are being processed. In CVMs, memory encryption and hardware-enforced access control prevent privileged software outside the guest, including the hypervisor, from directly inspecting or tampering with protected memory [Russinovich et al. 2021, Confidential Computing Consortium 2022].

This approach differs from purely cryptographic computation, such as homomorphic encryption or secure multi-party computation. In CC, the processor still computes over plaintext inside a protected hardware boundary, while memory and external access remain protected. This design favors compatibility and performance, but it shifts part of the trust assumption to the processor, firmware, and attestation ecosystem.

2.2. AMD SEV-SNP and Intel TDX

AMD SEV-SNP protects full virtual machines by combining memory encryption with integrity protections for guest memory mappings. A key component is the Reverse Map Table (RMP), which helps enforce ownership and access control over memory pages, reducing the ability of an untrusted hypervisor to remap or tamper with guest memory [AMD 2020, Kaplan 2023].

Intel TDX similarly protects full virtual machines, called Trust Domains, by isolating them from the hypervisor and other privileged software layers. TDX relies on hardware and firmware mechanisms, including the Secure Arbitration Mode (SEAM), to enforce isolation and protect guest memory [Intel 2023, Misono et al. 2024].

2.3. Threat Model and Scope

The evaluated scenario is a tenant-controlled AI inference workload running inside public-cloud VMs, where the cloud provider controls the physical host, virtualization stack, firmware updates, and operational infrastructure. The tenant seeks to reduce trust in these privileged layers while processing text documents with a document-embedding model.

Given this scenario, the protected assets include input documents, model execution state and generated embeddings while they are processed inside the guest environment. The target threats

include unauthorized inspection of guest memory, privileged software compromise, accidental exposure through debugging or crash dumps, and malicious or compromised hypervisor behavior.

This work excludes protection against vulnerabilities inside the guest application, malicious output disclosure, denial of service, or all side-channel attacks [Jauernig et al. 2021, Confidential Computing Consortium 2022]. Remote attestation is also not fully implemented in this proof of concept. Instead, we record local operational evidence of confidential mode, such as CPU flags and guest devices, and treat full remote attestation with key release as future work.

2.4. Related Work

Prior work has analyzed CVMs and their performance implications across different TEE technologies. Misono et al. provide an empirical analysis of AMD SEV-SNP and Intel TDX, showing that overheads depend heavily on workload characteristics [Misono et al. 2024]. Liu et al. evaluate the performance impact of Intel TDX and AMD SEV-SNP across application workloads, showing that CC overhead varies by workload and reinforcing the need for workload-specific evaluation [Liu et al. 2025]. CONFBENCH extends this direction by providing a tool for benchmarking heterogeneous TEE-backed environments, including Intel TDX, AMD SEV-SNP, and ARM CCA. It covers ML inference, DBMS, native OS, FaaS, and attestation workloads, offering a broader framework for evaluating confidential virtual machines [De Murtas et al. 2025].

In contrast to these broad benchmarking efforts, our work focuses on a narrower public-cloud proof of concept with artifact traceability, environment metadata collection, and a CPU-only document embedding workload. This narrower scope allows us to emphasize reproducibility, auditability, and cautious interpretation of performance results in a commercially available public-cloud setting.

3. Experimental Methodology

The evaluation compares baseline VM and CVM configurations under the same document-embedding benchmark. The methodology defines the cloud environments, workload parameters, artifact collection process, and performance metrics used to interpret the results.

3.1. Experimental Environments

The experiment evaluates the scenario described in Section 2.3 through a tenant-controlled document-embedding workload executed on public-cloud VMs. The experimental design compares each CVM with a non-confidential baseline VM from the same processor family under the same workload parameters. Experiments were conducted on Google Cloud using four environments with 4 vCPUs and 16 GB of RAM, as summarized in Table 1.

Table 1. Experimental environments.

Environment	Instance	Processor	CC Mode
AMD Baseline	n2d-standard-4	AMD EPYC Milan	SEV-SNP Off
AMD SEV-SNP	n2d-standard-4	AMD EPYC Milan	SEV-SNP On
Intel Baseline	c3-standard-4	Intel Sapphire Rapids	TDX Off
Intel TDX	c3-standard-4	Intel Sapphire Rapids	TDX On

Comparisons are made only within the same processor family: AMD Baseline versus AMD SEV-SNP, and Intel Baseline versus Intel TDX. We do not assume direct equivalence between AMD and Intel instances because they differ in microarchitecture, platform design, and potentially in I/O behavior.

3.2. AI Workload

The evaluated workload is a CPU-only AI inference pipeline based on document embeddings. The benchmark uses the BAAI/bge-m3 embedding model [Chen et al. 2024] and a fixed document corpus, processed under the same parameters across all environments. The repository provides a deterministic reference corpus for future runs. Concretely, the implemented benchmark measures the throughput and latency of embedding generation for text documents under different input lengths and batch sizes. In real deployments, this type of operation is representative of AI services such as semantic search, retrieval pipelines, and other embedding-based processing over private, proprietary, or personal documents in public cloud environments.

The benchmark varies two parameters: `max_length`, with values 256, 512, 1024, and 2048, and `batch_size`, with values 4, 8, and 16. These settings capture different benchmark conditions, from shorter text inputs to longer document contexts and from smaller to larger batch execution. The primary throughput metric is documents per second (docs/s), following current CVM benchmarking practice [Misono et al. 2024]. Latency is reported using P50 and P95 percentiles [Dean and Barroso 2013]. Since inference is performed in batches, per-document latency is interpreted as an estimate derived from batch execution time.

3.3. Reproducibility and Auditability

Each run produces structured artifacts, including raw logs, consolidated CSV files, execution metadata, timestamps, software version identifiers, and SHA-256 hashes of relevant inputs and outputs. The goal is to make each result traceable to a specific workload configuration and execution environment. Before benchmark execution, the infrastructure records environment characteristics such as kernel version, CPU information, memory, storage, and evidence of confidential mode. For AMD SEV-SNP, this includes checking guest-visible SEV-SNP device paths. For Intel TDX, this includes checking TDX-related CPU flags.

3.4. Metrics

We report throughput, P50 latency, P95 latency, and tail amplification, defined as the P95/P50 ratio. Throughput captures steady-state processing capacity, while P50 represents typical latency and P95 captures tail behavior. Tail amplification is used as a stability indicator. For each metric, we compare the confidential environment with its baseline. Positive variation in throughput indicates improvement, while negative variation in latency indicates reduction. However, observed improvements are not automatically attributed to CC, since public-cloud infrastructure may introduce confounding factors.

4. Results

4.1. AMD SEV-SNP

Table 2 compares the best observed configurations for AMD Baseline and AMD SEV-SNP. In all evaluated input lengths, the SEV-SNP environment presented higher throughput and lower P50 and P95 latency than the corresponding baseline for this document-embedding benchmark.

The observed throughput gain ranges from 11.5% to 13.1%, while P50 latency decreases by up to 15.6%. These results indicate that enabling SEV-SNP did not introduce systematic degradation for the evaluated benchmark. Nevertheless, we interpret these gains cautiously. Public-cloud factors such as I/O path differences, storage behavior, cache effects, and host placement may influence the observed performance. Therefore, we do not claim that SEV-SNP intrinsically improves performance.

Table 2. AMD Baseline versus AMD SEV-SNP.

Max length	Baseline	SEV-SNP	Δ docs/s	Δ P50	Δ P95
256	0.9835	1.0967	+11.5%	-10.0%	-12.0%
512	0.5859	0.6566	+12.1%	-15.3%	-13.0%
1024	0.3891	0.4356	+11.9%	-15.6%	-9.5%
2048	0.2964	0.3353	+13.1%	-12.4%	-13.4%

4.2. Intel TDX

As summarized in Table 3, Intel TDX consistently outperformed the Intel Baseline in terms of throughput across all evaluated input lengths. However, the performance gains observed with TDX were more modest than those observed with AMD SEV-SNP for the same benchmark.

Table 3. Intel Baseline versus Intel TDX.

Max length	Baseline	TDX	Δ docs/s	Δ P50	Δ P95
256	1.6594	1.7523	+5.6%	-4.4%	+2.4%
512	0.9757	1.0228	+4.8%	-3.7%	-3.6%
1024	0.6371	0.6713	+5.4%	-5.5%	-4.0%
2048	0.4755	0.5035	+5.9%	-5.6%	-5.8%

The observed throughput gain ranges from 4.8% to 5.9%. P50 latency decreases in all cases, while P95 increases only for the smallest input length and decreases for larger contexts. As with AMD, the main conclusion is not that TDX accelerates the workload, but that the evaluated TDX configuration did not introduce systematic performance degradation.

4.3. Tail Behavior

Tail amplification is measured as the P95/P50 ratio and is used as an indicator of latency stability. Table 4 summarizes the relative variation in tail amplification for the confidential environments with respect to their corresponding baselines, derived from the P50 and P95 values reported in Tables 2 and 3. Overall, the observed variations are small to moderate and do not indicate a consistent penalty in tail amplification for the confidential environments.

Table 4. Relative variation in tail amplification (P95/P50) of confidential environments with respect to their baselines.

Platform	Max length	Δ Tail Amplification
AMD SEV-SNP	256	-2.2%
AMD SEV-SNP	512	+2.7%
AMD SEV-SNP	1024	+7.2%
AMD SEV-SNP	2048	-1.1%
Intel TDX	256	+7.1%
Intel TDX	512	+0.1%
Intel TDX	1024	+1.6%
Intel TDX	2048	-0.2%

5. Discussion and Threats to Validity

5.1. Interpreting Performance Gains

The results show that CC did not introduce systematic degradation in the evaluated benchmark. In some cases, confidential environments outperformed their baselines. However, these gains should not be interpreted as intrinsic properties of AMD SEV-SNP or Intel TDX. Public-cloud environments may differ in host placement, storage path, I/O behavior, CPU scheduling, firmware versions, and resource contention. This distinction is central to our interpretation. The contribution of this work is not the claim that CC improves performance, but the demonstration that hardware-assisted data-in-use protection can be evaluated in a reproducible and auditable way, and that its overhead was not prohibitive in the evaluated scenario.

5.2. Validity Limitations

A public cloud is not a deterministic environment. Even when instance types and parameters are matched, the tenant does not fully control the physical host, neighboring workloads, or provider-level optimizations. Therefore, the methodology focuses on reproducibility, auditability, and careful interpretation rather than deterministic isolation of all variables. Repetition-level measurements are not available for the original experimental campaign. The reported results are therefore treated as descriptive point estimates, and no confidence intervals or claims of statistical significance are presented. The accompanying artifact supports repeated executions and the calculation of 95% confidence intervals for future evaluations, following established performance-evaluation practice [Jain 1991].

The workload is also limited to CPU-only AI inference based on document embeddings. The results should not be generalized to databases, network-intensive services, transaction processing, GPU workloads, training pipelines, or memory-intensive applications without further evaluation. Prior studies show that CC overhead varies substantially depending on workload characteristics [Liu et al. 2025, Misono et al. 2024].

Finally, this proof of concept records local operational evidence of confidential execution, but it does not implement a complete remote attestation flow. A production deployment should integrate attestation reports with a verifier and a key management service, so that secrets are released only after the platform and workload measurements satisfy an expected policy.

6. Conclusion and Future Work

This paper has evaluated Confidential Computing in public clouds through a CPU-only document-embedding benchmark on Google Cloud, comparing baseline VMs and Confidential VMs based on AMD SEV-SNP and Intel TDX. The benchmark represents AI services that process sensitive or personal text data through embedding generation and retrieval-oriented pipelines. Results indicate that enabling CC did not introduce systematic performance degradation, while observed gains remain subject to public-cloud confounders such as I/O paths, host placement, scheduling, and storage behavior.

Future work will expand the benchmark suite beyond document embeddings by incorporating additional AI workloads as well as DBMS, native OS, FaaS-style, and attestation workloads, following broader frameworks such as CONFBENCH [De Murtas et al. 2025]. We also plan to move from local evidence collection to policy-based remote attestation and secret release using Confidential Containers and Trustee, allowing models, keys, or protected datasets to be released only after the workload proves that it runs in an expected hardware TEE with a measured software stack [Confidential Containers Project 2026]. Finally, we plan to evaluate confidential containers as a finer-grained deployment model for Kubernetes-based and microservice-oriented workloads.

Artifact Availability

The source code, scripts and reference corpus are publicly available at: `https://github.com/lsfoschine/PoC_Computacao_Confidencial`

Acknowledgment

This work is supported by the Ministry of Science, Technology and Innovation (MCTI), with financial resources from the National Fund for Scientific and Technological Development (FNDCT) administered by the Brazilian Funding Authority for Studies and Projects (FINEP), specifically within the scope of the INSPIRE projects - National Data Infrastructure, Core, and AI Solutions for Public Service (Contract 01.25.0728.00, Reference 2315/25), executed in partnership with the Ministry of Management and Innovation (MGI). This work exclusively reflects the views of the authors; the funding agencies are not responsible for the use of the information presented herein.

Use of Generative AI

Generative Artificial Intelligence tools, including ChatGPT (OpenAI), were used during the preparation of this work to assist with LaTeX formatting, structural organization, language revision, section refinement, table positioning, and Overleaf manuscript editing. These tools were also used as programming assistance for auxiliary benchmark code, including corpus-generation scripts and code used to execute the embedding workload.

Generative AI tools were not used to conduct the experiments, operate the cloud environments, generate experimental results, validate measurements, define scientific claims, perform the technical analysis, or determine the conclusions presented in this manuscript. All architectural decisions, experimental design choices, result analyses, interpretations, and final claims were reviewed and validated by the authors, who assume full responsibility for the originality, technical correctness, and integrity of the work.

References

AMD (2020). SEV-SNP: Strengthening VM isolation with integrity protection and more. Technical Report 56860, Advanced Micro Devices.

Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D., and Liu, Z. (2024). BGE M3-Embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Confidential Computing Consortium (2022). Confidential computing: Hardware-based trusted execution for applications and data. Technical report, Linux Foundation. Accessed: 2026-03-24.

Confidential Containers Project (2026). Trustee: Trusted components for attestation and secret management. `https://github.com/confidential-containers/trustee`. Accessed: 2026-05-11.

De Murtas, A., D'Elia, D. C., Di Luna, G. A., Felber, P., Querzoni, L., and Schiavoni, V. (2025). CONFBENCH: A tool for easy evaluation of confidential virtual machines. In *2025 55th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 279–288.

Dean, J. and Barroso, L. A. (2013). The tail at scale. *Communications of the ACM*, 56(2):74–80.

Intel (2023). *Intel Trust Domain Extensions (Intel TDX) Technical Overview*. Intel Corporation. Accessed: 2026-03-24.

Jain, R. (1991). *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. John Wiley & Sons, New York, NY, USA.

Jauernig, P., Sadeghi, A.-R., and Stapf, E. (2021). The state of confidential computing: Challenges and opportunities. *IEEE Security & Privacy*, 19(2):10–21.

Kaplan, D. (2023). Hardware VM isolation in the cloud: Enabling confidential computing with AMD SEV-SNP technology. *ACM Queue*, 21(4):49–67.

Liu, X., Sankhe, M., Rodrigues, R., Szydło, T., Ranjan, R., and Jha, D. N. (2025). Benchmarking confidential computing: Application performance comparison of TDX v/s SEV-SNP. In *2025 IEEE International Conference on High Performance Computing and Communications (HPCC)*, pages 178–185.

Misono, M., Stavrakakis, D., Santos, N., and Bhatotia, P. (2024). Confidential VMs explained: An empirical analysis of AMD SEV-SNP and Intel TDX. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 8(3):Article 36.

Russinovich, M., Costa, M., Fournet, C., Chisnall, D., Delignat-Lavaud, A., Clebsch, S., Vaswani, K., and Bhatia, V. (2021). Toward confidential cloud computing. *Communications of the ACM*, 64(6):54–61.