// FHE_BENCHMARKING_V1.0

A NEW FHE BENCHMARKING SUITE

Standardizing Application-Driven Performance Evaluation for Encrypted Computation

Andreea Alexandru • Flavio Bergamaschi • Shruthi Gorantala • Shai Halevi

Duality Technologies • Optalysys • Google • AWS

FHE.org 2026 Conference

THE PARADIGM SHIFT

THE GAP: Potential adopters currently lack a common yardstick. Existing numbers are library-specific, tied to isolated cryptographic micro-benchmarks, or linked to specific academic papers.

THE GOAL: The Suite provides **relevant, fully-specified, end-to-end workloads** that represent interesting and useful use cases, informing business decisions through performance data standardized across libraries and backends.

PERSONAS SERVED

- **Application Developers**
  Assess E2E feasibility and cost of FHE for their workloads.
- **FHE Library/Compiler Developers**
  Optimize scheme-level performance targets.
- **Hardware Vendors**
  Guide design and testing of accelerators for FHE workloads.

METRIC A: WALL-CLOCK

Latency & Throughput

Wall-clock time for single workload instances and batch processing performance.

METRIC B: FOOTPRINT

Memory, Storage & Communication

Maximum RAM usage and total storage for keys, ciphertexts, and intermediate values. Bandwidth of data exchange.

METRIC C: QUALITY METRICS

Correctness & Accuracy

Result validation and accuracy loss compared to plaintext reference (where applicable).

SECURITY MANDATE

128-BIT

Minimum required security level. Submitters must provide formal justification of parameter selection.

SUITE STRUCTURE

HARNESS SUBDIRECTORY

Immutable executions and measurement logic provided by organizers.

SUBMISSION SUBDIRECTORY

Submitter-filled solution with implementation code and full documentation. Core workload computation must be performed on **encrypted data**, though pre-processing before encryption and post-processing after decryption are permitted.

SUBMISSION OPTIONS

- **Open-source software**
  Complete implementation code.
- **Closed-source software**
  Pre-compiled libraries or containers.
- **Hardware and Backed**
  Shims for backend communication. Backends should remain available (for testers) for at least a few weeks following initial submission.

SUBMISSION PROTOCOL

1. Fork Repository github.com/fhe-benchmarking/<workload>.
2. Implement Workload: populate /submission or /submission-remote subdirectories.
3. Maintain Harness: do not modify the /harness subdirectory.
4. Document Solution: update the README with all relevant information and optionally provide more documentation in the /docs subdirectory.
5. Generate Measurement Files: run the harness with a --num_runs 3 argument and commit the measurements files.
6. Submit Results: make the fork public and notify suite organizers via the form provided at fhe-benchmarking.github.io.

TARGET EVALUATION PLATFORMS

- Software-only submissions are tested on normalized platforms to ensure hardware-agnostic comparisons. **Recommended**: platforms with 5th-gen Intel Xeon (Emerald Rapids), 96 vCPUs and ample memory:

  EC2 I7ie.24xl GCP c4-highmem-96 Azure Standard-E96s-v6

- Submissions that rely on accelerated computing should specify the acceleration hardware used (e.g., number and type of GPUs).

CURRENT WORKLOADS

VECTOR SEARCH `fetch-by-similarity`

Private database queries using **Cosine Similarity** search over encrypted data. Essential for private biometrics and RAG.

DB size: 50K, 100K, 20M
Record size [b]: 128, 256, 512

LARGE INTEGER MULTIPLICATION `Zn-multiplication`

Multiplication of two large encrypted integers. Essential for blockchain applications.

Batch size: 1, 1K, 100K, 10M
Size progression [b]: 64, 128, 256

MACHINE LEARNING INFERENCE `ml-inference`

Privacy-preserving **machine learning inference** on encrypted inputs. Requirement: must meet accuracy thresholds for batch inference.

Batch size: 1, 1K, 100K, 1M
Model progression: MNIST, CIFAR-10, ResNet-50, ..., BERT

DEVELOPMENT ROADMAP

FHE TRANSCIPHERING	ENCRYPTED FACE RECOGNITION	PIR/PSI
PRIVATE SQL QUERIES	FHE SIGNATURES	SPARSE MATRIX MULTIPLICATION

Active contributions from the homomorphicencryption.org community

FHE BENCHMARKING RESULTS

Example: fetch-by-similarity (Fetch – Small)

Submitter			Bandwidth			Timing (harness)					Timing (server)	
Name	Env	Date	Keys	DB	Query	Total	Keygen	DB Enc	Q Enc	Compute	Total	Compute
Reference	CPU	2026-02-13 22:21:16	2.4G	5.6G	24M	1.8759m	4.2228s	33.73s	2.3333ms	1.223m	1.2167m	1.1833m

Get in Touch & Get Involved

fhe-benchmarking@homomorphicencryption.org

github.com/fhe-benchmarking

fhe-benchmarking.github.io

Duality Optalysys Google aws