FHE Benchmarking Suite

FHE.org Software Day

September 24th 2026



Shruthi Gorantala
gshruthi@google.com



Andreea Alexandru
aalexandru@dualitytech.com

Agenda



Introduction



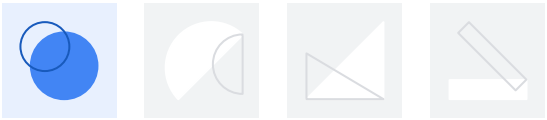
Design



Workloads



Call to Action

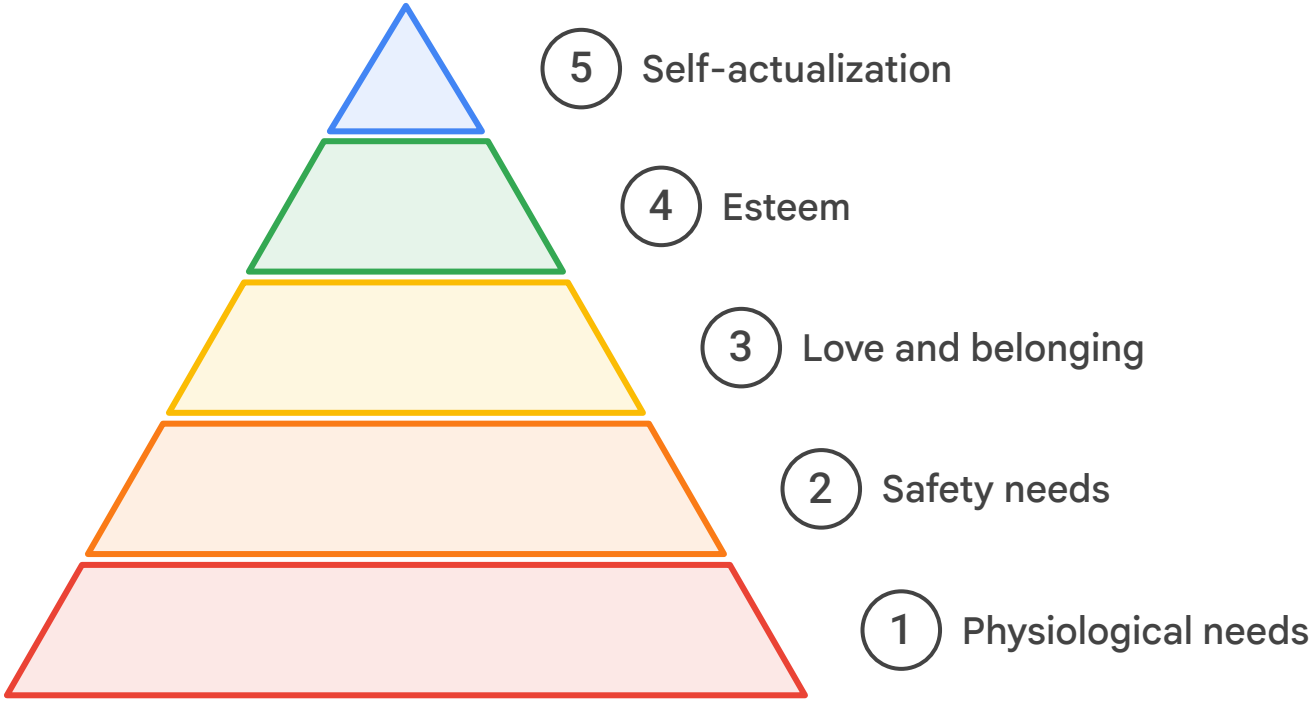


FHE Benchmarking

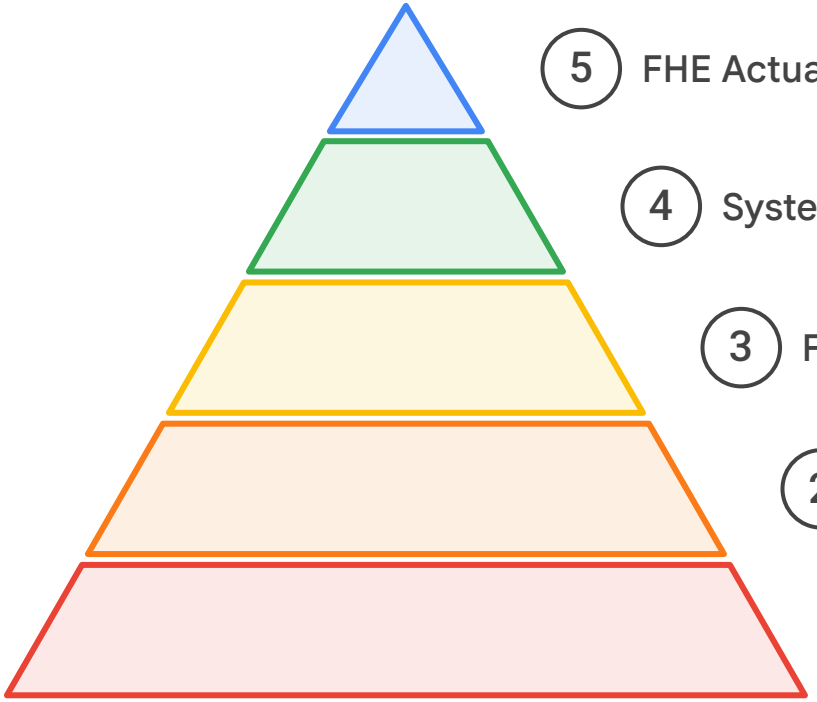
Introduction



Maslow's Hierarchy of Needs



FHE Hierarchy of Needs



5 FHE Actualization in privacy

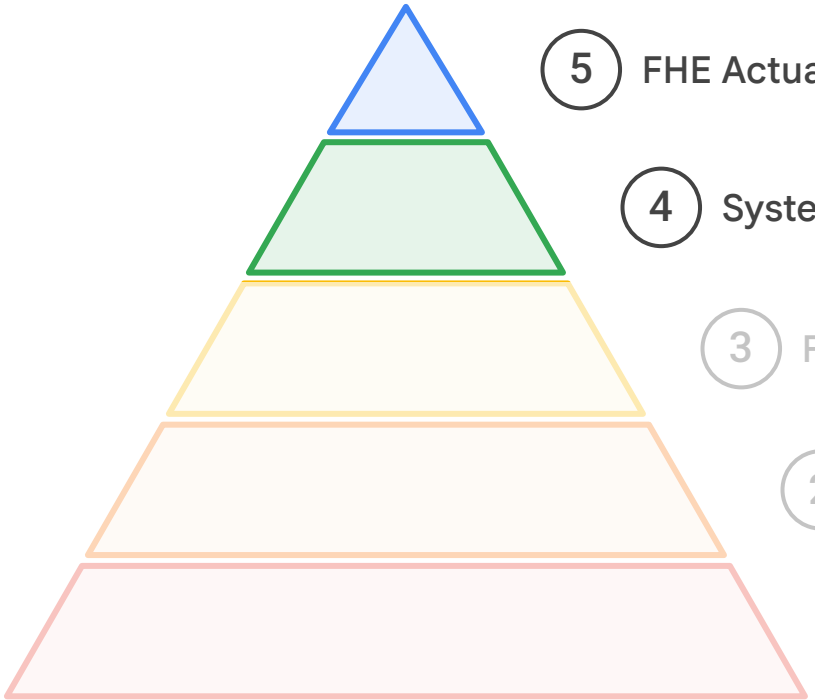
4 Systems integration / Privacy engineering

3 FHE application development

2 Compilers / Transpilers

1 FHE Instruction Set Architecture

FHE Hierarchy of Needs



5 FHE Actualization in privacy

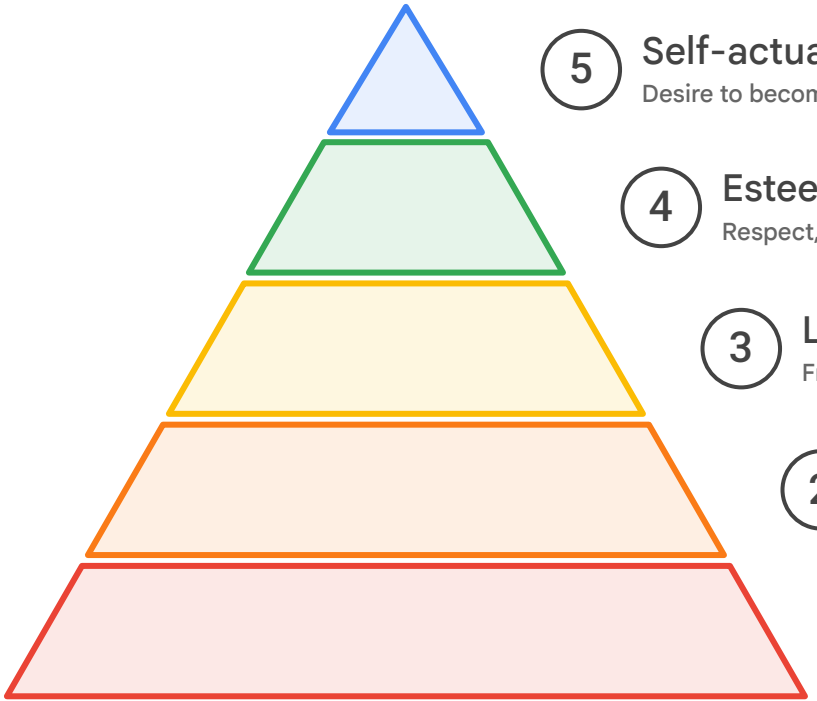
4 Systems integration / Privacy engineering

3 FHE application development

2 Compilers / Transpilers

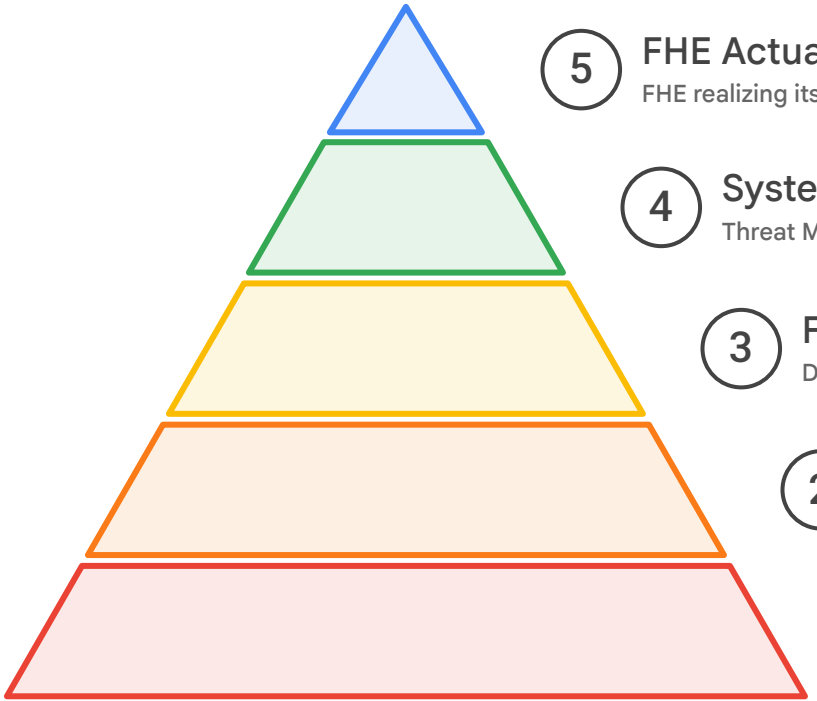
1 FHE Instruction Set Architecture

Maslow’s Hierarchy of Needs



5. **Self-actualization**
   Desire to become the most that one can be
4. **Esteem**
   Respect, self-esteem, status, recognition, strength, freedom
3. **Love and belonging**
   Friendship, intimacy, family, sense of connection
2. **Safety needs**
   Personal security, employment, resources, health, property
1. **Physiological needs**
   Air, water, food, shelter, sleep, clothing, reproduction

FHE Hierarchy of Needs



5 **FHE Actualization**
FHE realizing its true potential as a PET

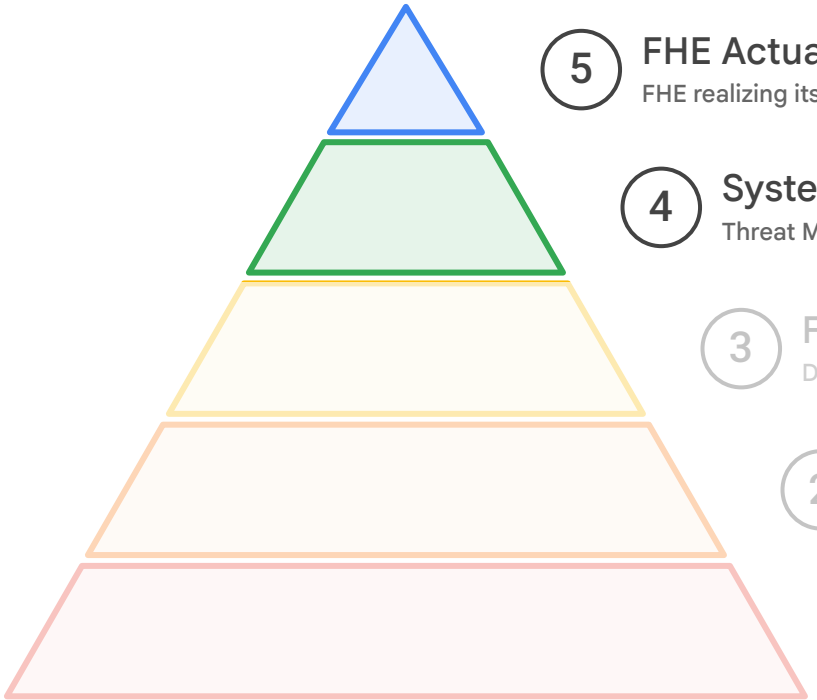
4 **Systems integration / Privacy engineering**
Threat Modeling, Privacy leaks, Network bandwidth, Observability

3 **FHE application development**
Development speed, Scheme Interoperability, Debugging

2 **Compilers / Transpilers**
Param Selection, Optimizers, Schedulers, Intermediate Representations

1 **FHE Instruction Set Architecture**
Data Manipulation, Ciphertext Maintenance, Hardware Cost

FHE Hierarchy of Needs



5 **FHE Actualization**
FHE realizing its true potential as a PET

4 **Systems integration / Privacy engineering**
Threat Modeling, Privacy leaks, Network bandwidth, Observability

3 **FHE application development**
Development speed, Scheme Interoperability, Debugging

2 **Compilers / Transpilers**
Param Selection, Optimizers, Schedulers, Intermediate Representations

1 **FHE Instruction Set Architecture**
Data Manipulation, Ciphertext Maintenance, Hardware Cost

Why we need benchmarking

- FHE is reaching practicality
  - Many applications can be feasibly deployed
- FHE Deployment stack requires multiple performance considerations on platforms
  - Not always feasible to measure in real time
- Benchmarks on standardized platforms allow for comprehensive and conclusive analysis of performance overhead

Goal: Standard Benchmarking Suite

- From a technology consumer point of view
- Helps product managers/developers make decisions
  - Which of my products could feasibly use FHE?
  - Which implementation(s) of what variant(s) can I use?
- Provide useful input for FHE deployment decisions
  - End-to-end metrics
- Well-specified workloads driven by real world use cases
- Baseline reference benchmarks

Challenge: Benchmark across Software & Hardware

- Standard platforms on cloud Hardware
  - CPU, GPU, TPU
- Specialized Hardware
  - FPGAs, ASICs, Photonics
- Open source and proprietary FHE libraries and compilers
- FHE libraries in all languages (C++, Python, Rust, Go, etc)
- Submission from remote services hosted on cloud

Who can use the benchmarks?



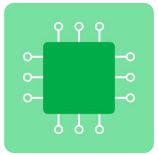
Application Developers

Interested in Application level performance for FHE Deployment



FHE Developers

Interested in FHE library and compiler based performance



Hardware Developers

Interested in Hardware Performance

What to measure?



Time



Space



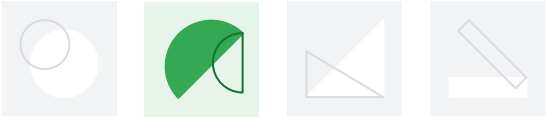
Network bandwidth



Precision



Cost



Design

Developed with HomomorphicEncryption.org



fhe-benchmarking.org

fhe-benchmarking.github.io

FHE Benchmarking Suite

Measuring Performance of Computing on Encrypted Data

About the FHE Benchmarking Suite

This Benchmarking Suite provides standardized workloads and metrics to he
feasible for their use cases. Our goal is to enable informed decisions about
comprehensive performance data across different workloads, implementatic

Each benchmark provides comprehensive metrics including:

- **Latency and Throughput** - Wall-clock time for single workload instanc
- **Memory Consumption** - Maximum RAM required during execution
- **Storage Requirements** - Keys, ciphertexts, and intermediate values
- **Communication Complexity** - Data exchanged between client and ser
- **Quality Metrics** - Accuracy loss compared to plaintext (where applicabl

Current Benchmark Workloads

Each workload represents a real-world use case with clear specifications, reference implementations, and correctness requirements.

- **Fetch-by-Similarity**. Private database queries using cosine similarity search over encrypted data.
- **ML Inference - Image Classification**. Privacy-preserving machine learning inference on encrypted images. We currently support the following datasets with more to come.
  - MNIST digit classification
  - CIFAR-10 image classification
- **Zn Multiplication**. Multiplication of two encrypted 64-bit encrypted integers. Other bit-sizes for the multiplicands will be added.
- **BERT**. Privacy-preserving BERT inference on encrypted inputs. Currently supports BERT-Base (110M) inference on the MRPC task in the GLUE benchmark.
- **Transciphering**. Homomorphic AES decryption, converting AES-encrypted data into FHE ciphertexts of the same plaintexts. The quality of the resulting ciphertexts is assessed by evaluating a mini-workload (maximum or inner product) over them.
- **Face Recognition**. End-to-end privacy-preserving face verification with encrypted feature extraction and matching, including single-pair and batched benchmark results.

Additional workloads are under development. Check back for updates.

The suite in numbers

- Officially launched in March 2026
- 6 published workloads
  - 34 benchmark variants (task x instance size)
  - 22 of them already have submissions!
- 49 published result entries in the tables: 18 from the 6 reference implementations + 31 from external submitters
- 10+ external submission packages, of which 3 are improved resubmissi

FHE Benchmarking Results - (Fetch - Small)

Column visibility

Search:

Submitter				Bandwidth			Timing (harness)					Timing (server)	
Name	Date	Env	R/L	Keys	DB	Query	Total	Keygen	DB Enc	Q Enc	Compute	Total	Compute
CryptoLab Inc.	2026-06-26 00:30:11	GPU	L	3G	128M	64K	7.7625s	3.5881s	494.8ms	21.23ms	42.1ms	19.42ms	14.84ms
Lattica-ai	2026-05-25 11:20:43	GPU	R	672M	2.3G	9M	2.7004m	12.53s	53.7s	2.074s	2.2116s		284ms
Lattica-ai	2026-02-23 21:49:44	GPU	R	1.1G	2.8G	12M	3.3609m	20.8s	54.44s	2.2528s	2.7835s		817ms
Reference	2026-02-13 22:21:16	CPU	L	2.4G	5.6G	24M	1.8759m	4.2228s	33.73s	362.6ms	1.223m	1.2167m	1.1833m

FHE Benchmarking Results - BERT (Single)

Search:

Column visibility

Timing (server) Bandwidth Timing (harness) Timing (server)

Submitter Keys Input Result Total Keygen Model Prep Input Gen Input Enc Compute Decrypt Total I/O Compute

DESILO 2026-07-14 02:04:17 GPU L 21.4G 46M 30M 14.16m 57.78s 7.0413m 13.9s 12.3s 5.3337m 1.1814s 4.9873m 2.9696m 2.0177m

Reference 2026-07-08 16:29:03 CPU L 21.4G 46M 30M 41.83m 51.62s 3.2822s 4.8858s 3.9553s 40.62m 633.3ms 40.24m 58.37s 39.27m

FHE Benchmarking Results - Zn Multiplication (Small Batch - 1000)

Column visibility

Search:

Name Submitter Bandwidth Timing (harness) Timing (server)

Date Env R/L Keys Input Result Total Keygen Input Gen Input Enc Compute Decrypt Total Compute

CryptoLab Inc. 2026-08-13 20:27:39 CPU L 928.5K 1.7M 400.1K 211.3ms 15.3ms 117.8ms 14.8ms 47.03ms 15ms 25.47ms 9.08ms

CryptoLab Inc. 2026-08-12 05:27:29 CPU L 1.5M 4M 2M 222.6ms 15.2ms 107.6ms 26.8ms 54.17ms 17.87ms 38.43ms 13.03ms

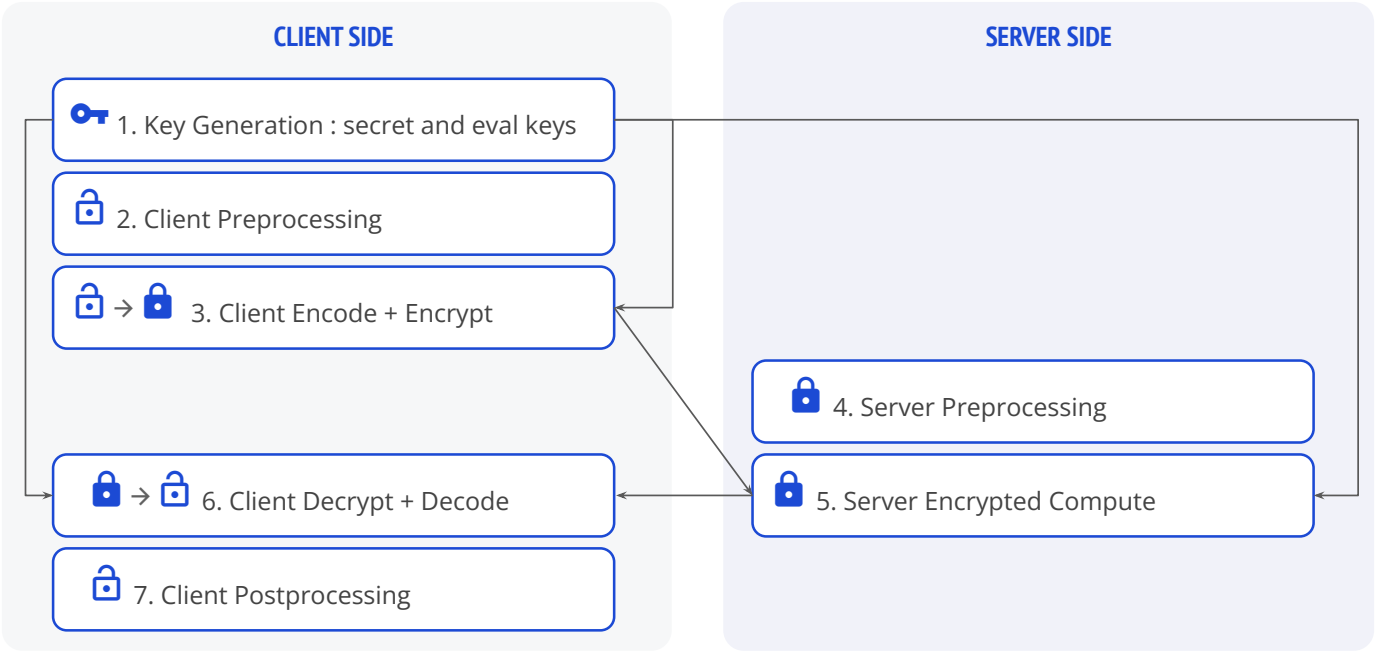
Hongren Zheng 2026-03-02 05:48:29 CPU L 2.3M 8M 2M 668.1ms 71ms 121.3ms 144.6ms 138.4ms 105.3ms

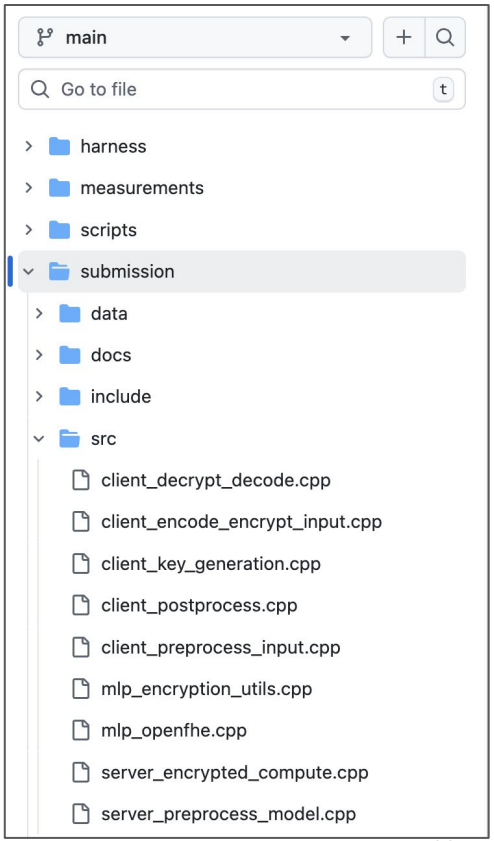
Reference 2026-02-20 03:37:04 CPU L 125M 1004.7M 502.3M 17.95m 800.8ms 129.4ms 7.4931s 17.8m 375.4ms

Suite Structure

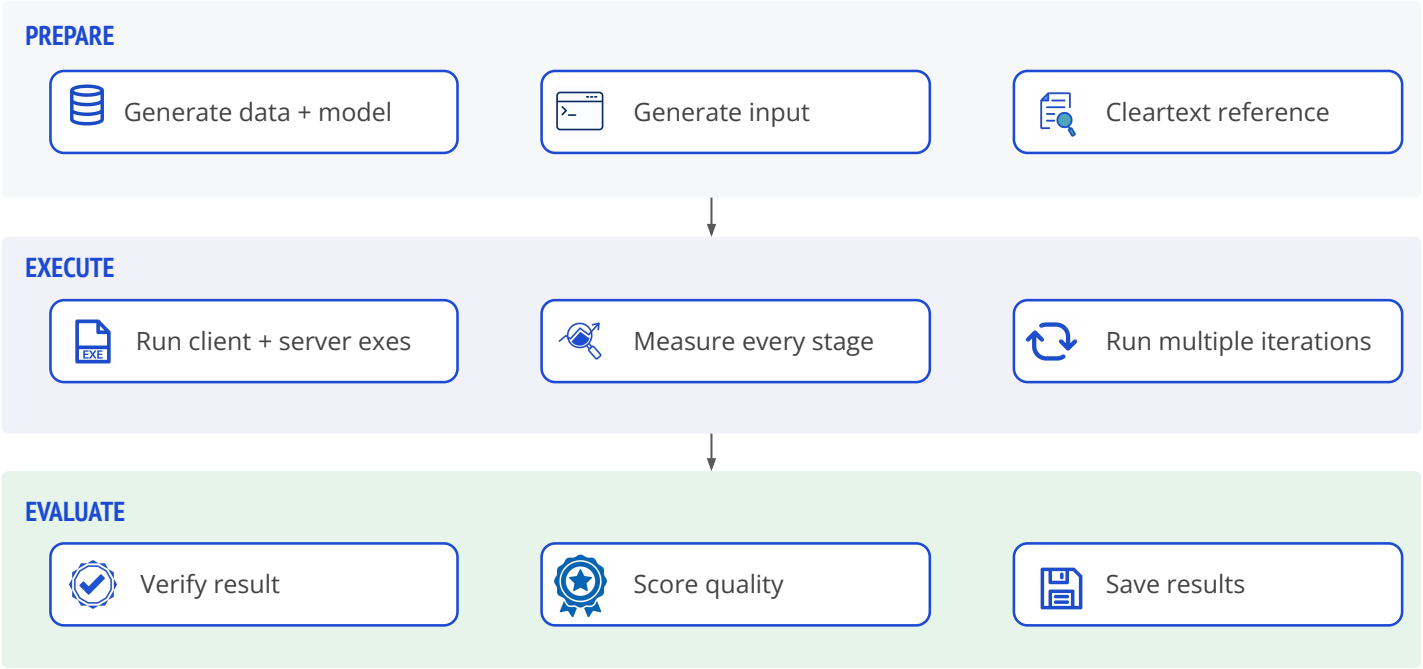


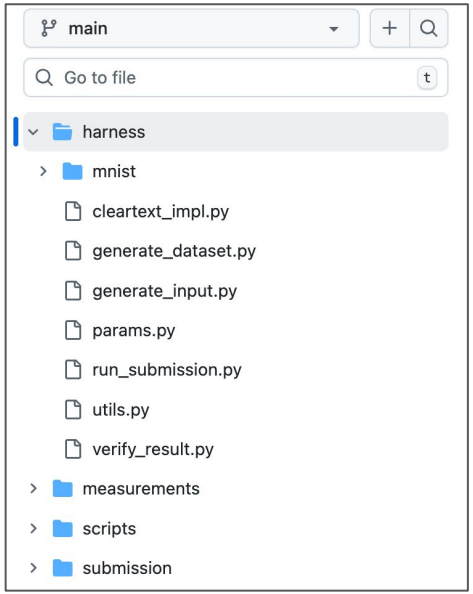
Suite structure: Two-party non-interactive protocol





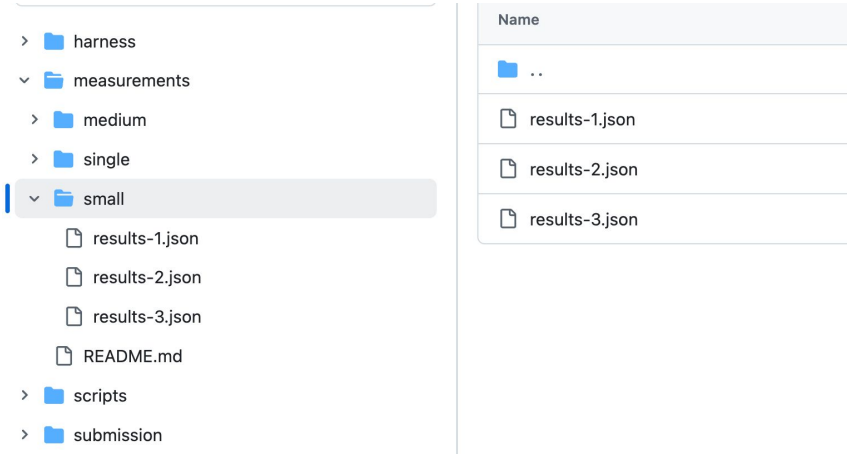
Suite Structure : Harness





Suite Structure : Measurements

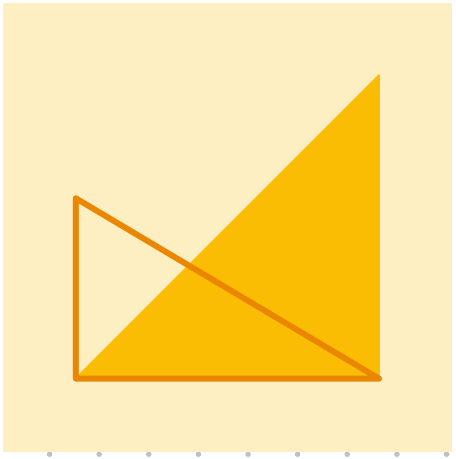

```
{
  "Timing": {
    "Test dataset generation": "4.7884s",
    "Key Generation": "0.9751s",
    "Encrypted model preprocessing": "0.0037s",
    "Input generation": "1.8093s",
    "Input preprocessing": "0.0035s",
    "Input encryption": "0.0936s",
    "Encrypted computation": "7.6682s",
    "Result decryption": "0.0842s",
    "Result postprocessing": "0.0028s",
    "Total": "15.4287s"
  },
  "Bandwidth": {
    "Public and evaluation keys": "1.0G",
    "Encrypted input": "5.0M",
    "Encrypted results": "1.0M"
  },
  "Server Reported": {
    "Encrypted computation": "6.515s",
    "Total": "6.515s"
  }
}
```



Workloads

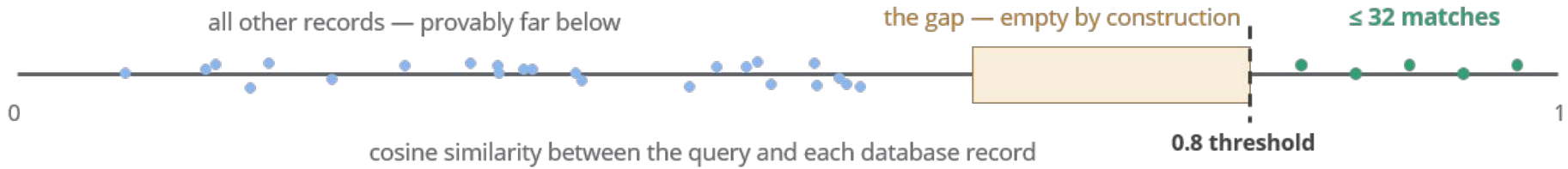
Tales from the field



Fetch by Similarity

Private k-nearest-neighbor vector-search over encrypted inputs, using cosine similarity
Essential for private biometrics and private RAG

- Both db and query are *encrypted* → *potentially large storage required*
- Deliberate simplification: *promise of gap in score and at most 32 matches above threshold*
  - The gap makes encrypted comparison cheap
  - < 32 matches make the response fixed-size



Fetch by Similarity

Workload variants

Count – how many records match per query

Fetch – return payloads of the matches (64-bit payloads)

#recs x dim – **Small**: 50K x 128, **Medium**: 1M x 256, **Large**: 20M x 512 (*no runs yet*)

Search on 1M records: CPU reference vs fastest GPU submission

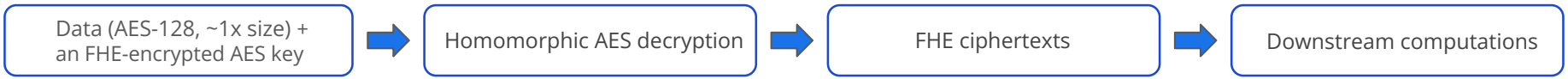
- Count: 65x reduction in end-to-end runtime (28.9 s) and >40x reduction in enc db size and enc query (4 GB, 64 KB)
- Fetch: retrieval costs 1.4x–1.7x more than count in server compute

Once the server is fast, the client dominates – one-time medium DB encryption is half of the end-to-end run: fixed-cost vs per-query cost

AES-FHE Transciphering

Bridge symmetric encryption and FHE: upload data AES-encrypted and compute under FHE

Essential for bandwidth-constrained clients and for data already encrypted at rest



- The quality of the transciphered ciphertexts (noise, remaining levels) is not straightforward

  ⟶ *Mini-workloads as downstream applications to test end-to-end performance/accuracy*

Maximum – comparison-heavy

Inner product mod 2^{16} – arithmetic-heavy

Toy: 1 block – ECB, **Small**: 16 – CTR,

Medium: 256 – CTR, **Large** 4096 – CTR

The number of blocks amortizes over the conversion + The conversion amortizes over the workload

Large-Integer Multiplication

Exact multiplication of encrypted 64-bit integers (mod 2^{64})

Essential for blockchain-style applications: many exact arithmetic operations

- Workload variants

  Single – one multiplication ⟶ latency

  Batched (Small, Medium, Large) – 1 K / 100K / 10M pairs ⟶ throughput

- The all-CPU leaderboard
  - Single: latency 2.34 s ⟶ 181 ms, key material 125 MB ⟶ 54 KB
  - 10M: latency 28.1 min ⟶ 52.7 s, encoding 76.6 GB ⟶ 16.4 GB, results 19.2 GB ⟶ 3.7 GB

*Batching in action: per-multiplication cost falls **~34,000×** from a single call to a 10 M batch.*

ML Inference

- ***Organized by dataset:*** ***Dataset and quality threshold are fixed*** *across submissions*
- ***Single & batched workloads:*** *Measure* ***single-query latency*** *and* ***batch throughput***
- ***Submitter freedom:*** *Bring* ***custom model architectures*** *and* ***pre-/post-processing***
- ***Cryptographic trade-offs:*** ***evaluation key sizes, ciphertexts, and compute***

Supported Submission Models

Open Source
Full source in submission/
(e.g., **Reference baselines**)

Closed Source
Pre-compiled library/binary
(e.g., **DESILO**)

Remote Backend
Client shims to cloud/HW backend
(e.g., **Lattica**)

ML Inference

FIXED BY THE WORKLOAD

- Dataset
- Quality threshold
- Single and batch workloads to measure throughput

FREE FOR THE SUBMITTERS

- Submitters can submit their own architecture!
- Submitters can chose to pre-process input / post process output as needed

- We support
  - Open source
  - Closed source - DESILO
  - Remote - Lattica

ML Inference - MNIST

Workload variants

Single – classify one 28×28 handwritten digit image (0–9)

Batched – classify a batch of digit images (quality bar: ≥90% accuracy)

#images (N) – **Small**: 100, **Medium**: 1K, **Large**: 10K (*all 4 variants run*)

MNIST digit classification (MLP): CPU reference vs fastest GPU submission

- Single & 1K batch: 229x end-to-end speedup on 1K batch (1.83 h ⟶ 28.8 s, 215 ms server compute) at 97.2% accuracy
- Large batch (10K, GPU): 49.9 s total (~5 ms/image, 1.05 s compute), 7.3 MB keys, 96.6% accuracy

GPU compute (~0.2–0.6 ms/batch), and encrypted accuracy (97.2%) matches plaintext (97.4%)

ML Inference - CIFAR-10

Workload variants

Single – classify one 32×32 RGB image into 10 object classes (CIFAR-10)

Batched – classify a batch of RGB images (quality bar: ≥90% accuracy)

#images (N) – **Small**: 100, **Medium**: 1K, **Large**: 10K (*batches: no runs yet*)

CIFAR-10 object classification (ResNet-20): CPU reference baseline by FHEON

- Single inference (CPU): 32.1 min encrypted compute (33.5 min total runtime), 534 ms input encryption, 370 ms decryption
- Bandwidth & keys: 50.4 GB evaluation keys (1.08 min keygen), 28 MB encrypted input, 8 MB result

Deep residual networks (ResNet-20) grow evaluation keys 50x vs MNIST (~50.4 GB vs 1 GB)

BERT Inference

Workload variants

Single – classify 1 sentence pair for semantic equivalence (MRPC)

Batched – classify a batch of pairs (quality bar: ≥75% accuracy)

#pairs (N) – **Small**: 100, **Medium**: 200, **Large**: 408 (*Med/Lrg: no runs yet*)

BERT-Base (110M) on MRPC: CPU reference vs GPU submission (**THOR - DESILO**)

- Single inference: 20x reduction in server compute (39.3 min ⟶ 2 min) with **21.4 GB** evaluation keys
- Small batch (100 pairs, GPU): 3.32 h compute (6.37 h server total) at 84.3% accuracy (vs ≥75% bar)

On GPU, encrypted BERT inference is I/O-bound: 3.0 of the 5.0 server minutes are spent moving 21.4 GB of keys

Face Recognition

Workload variants

Single – verify 1 CelebA face pair via encrypted similarity score

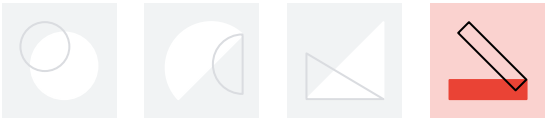
Batched – verify a batch of pairs (quality bar: EER ≤ ArcFace + 0.15)

#pairs (N) – **Small**: 128, **Medium**: 256, **Large**: 1,024 (*all 4 variants run*)

CelebA face verification: **CryptoFace** CPU reference

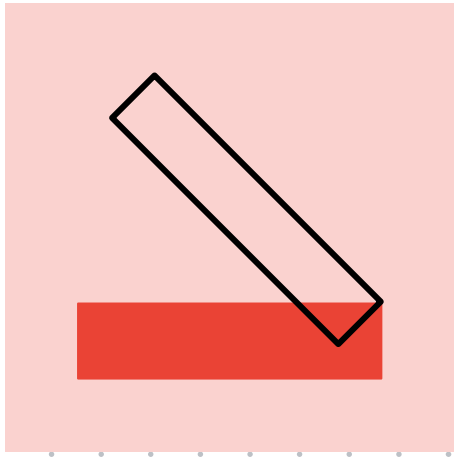
- One-time setup & payload: 114.1 GB packed encrypted model + **28.8 GB keys** (~11.6 min prep); marginal verify is 1.83 min
- Batched: 49.5 min (Small) to 6.07 h (Large); TAR@FAR=1% is 0.84–0.85 vs 0.94 ArcFace

The encrypted model weights are: 114 GB.



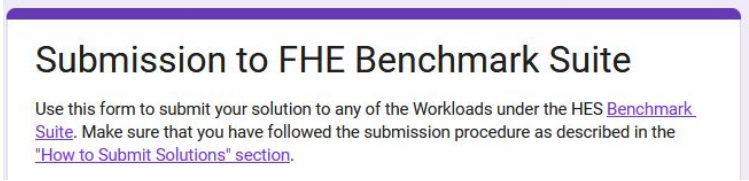
Call to Action

New workloads, submissions and maintainers



How to submit

- **Fork Repository** github.com/fhe-benchmarking/<workload>.
- **Implement Workload**: populate /submission subdirectories
- **Document Solution**: update the READMEs with all relevant information and optionally provide more documentation in the /docs subdirectory.
    - Justification of achieving 128-bit security level, < 2-40 probability failure
- **Generate Measurement Files**: run the harness with a --num_runs 3 argument
- **Submit Form**: make the fork public and notify suite organizers via the **form** on the website; sign author certification



Current Workloads

- Fetch By Similarity
- FHE Transciphering
- Zn multiplication
- ML Inference - Image classification
  - MNIST DIGIT classification
  - CIFAR 10 image classification
- BERT inference
- Face recognition

New Workloads

- LLM Inference
- ML Inference variants
  - Imagenet - resnet50?
- String search
- Database queries
- FHE based PSI
- ...

How to add new workloads

- Use case justifying the workload
- Initial design of workload: Input, Output, Correctness, Quality, Security
- Implementation of the harness
  - Data generation and cleartext implementation
- Reference submission for encrypted solution
- Add it to [fhe-benchmarking/repositories]

Thank you

fhe-benchmarking.org

Contributors are welcome!

fhe-benchmarking@homomorphicencryption.org
gshruthi@google.com
aalexandru@dualitytech.com

We are looking for submissions, new workloads, maintainers.

ML Inference

- Computation-communication trade-offs (**encrypted MNIST inference**)
  - For one inference, both GPU submissions send ~50 MB: one as 46 MB of evaluation keys + 128 KB input, the other as ~110 KB of keys + 51 MB of redundantly packed input ⟶ thin-client vs batch throughput
  - Trading rotations for bandwidth collapses compute: the key-light design runs a batch in ~0.6 ms of server compute — 27–380× below the other GPU submission
  - Symmetric (secret-key) encryption halves upload via seeded ciphertexts (valid when the client owns data and keys, unavailable in public-key ingestion settings)

Insights

- Computation-communication trade-offs (**encrypted MNIST inference**)
  - For one inference, both GPU submissions send ~50 MB: one as 46 MB of evaluation keys + 128 KB input, the other as ~110 KB of keys + 51 MB of redundantly packed input → thin-client vs batch throughput
  - Trading rotations for bandwidth collapses compute: the key-light design runs a batch in ~0.6 ms of server compute — 27–380× below the other GPU submission
  - Symmetric (secret-key) encryption halves upload via seeded ciphertexts (valid when the client owns data and keys, unavailable in public-key ingestion settings)
- *Warm vs. cold starts matter*: reported 0.57 ms compute excludes a 58 ms one-time setup, ~100× the steady-state cost

Insights (cont'd)

- **Encrypted BERT inference**: on GPU, encrypted inference is I/O-bound, not compute-bound
  - For a single inference, 3.0 of the 5.0 server-side minutes are I/O (moving ~21 GB of keys and working set) vs 2.0 min of actual encrypted compute
  - *Accelerating FHE shifts the bottleneck to memory and data movement*

Insights (cont'd)

- **Encrypted face recognition**:
  - The encrypted model is itself the payload: 114 GB packed encrypted model + 29 GB of keys ship once; after ~30 min of one-time preparation, the marginal cost of a verification is ~1.8 min of server compute
  - First workload where the quality gap is visible: encrypted TAR@FAR=1% is 0.84–0.85 vs 0.93–0.94 for the plaintext model, unlike MNIST where encrypted accuracy matched plaintext
- *Quality columns are not a formality once models get hard*