SparkFHE: Distributed Dataflow Processing with Fully Homomorphic Encryption

Peizhao Hu, Asma Aloufi, Kim Laine, and Viktoria Koscinski

Introduction and Motivation

– Incorporate homomorphic encryption into a highly efficient data analytics platform to support secure data analytics and machine learning on encrypted data in the cloud.

– Explore distributed and parallel computing architecture to accelerate the evaluation of homomorphic functions.

– Provide a high-level abstraction to ease the development of secure data analytics algorithms.

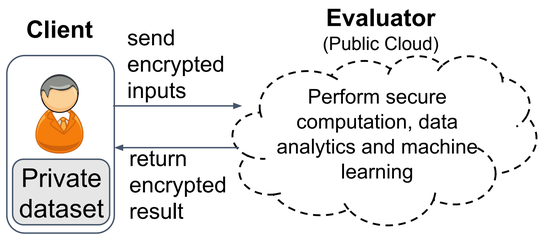

Figure 1: Blindfolded Cloud Computation Overview

Homomorphic Encryption

Homomorphic encryption (HE) supports computations on encrypted data without decryption [1]. In a nutshell, for messages m and m' we want the following properties to hold:

$$Dec(Enc(m) \oplus Enc(m')) = Dec(Enc(m + m'))$$

$$Dec(Enc(m) \otimes Enc(m')) = Dec(Enc(m \times m'))$$

where applying homomorphic addition $\oplus$ or multiplication $\otimes$ to ciphertexts has the same effect as applying similar operations to plaintexts and then encrypting the results.

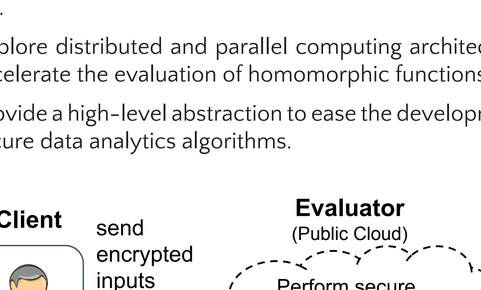

Figure 2: Additive Homomorphism Example

Distributed and Parallel Computing Frameworks

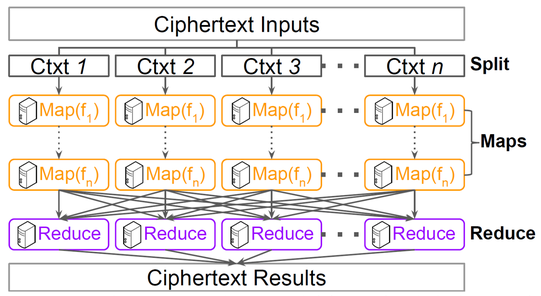

Figure 3: The Map&Reduce Model (f is a lambda function)

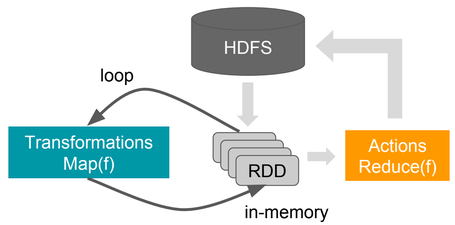

Figure 4: Dataflow Processing Model

OpenMPI (Message Passing Interface)

- Low-level library for parallelizing tasks but requires implementation of task and resource allocation algorithms.

Apache Hadoop (MapReduce model)

+ Supports distributed and parallel data processing and analytics following the Map&Reduce model.

- Lacks data abstraction for advance in-memory processing.

- Follows the "*move data to computation*" model that leads to high data movement overhead.

Apache Spark (Resilient Distributed Datasets)

+ Introduces high-level data abstraction (RDD) that captures the linage of datasets which improves in-memory processing [2].

+ Follows the "*move computation to data*" model that leads to low data movement overhead.

+ Provides various support for efficient task and resource allocation in a large cloud setting.

+ Supports low-latency data (streaming) analytics and machine learning as a service (MLaaS).

Overview of SparkFHE

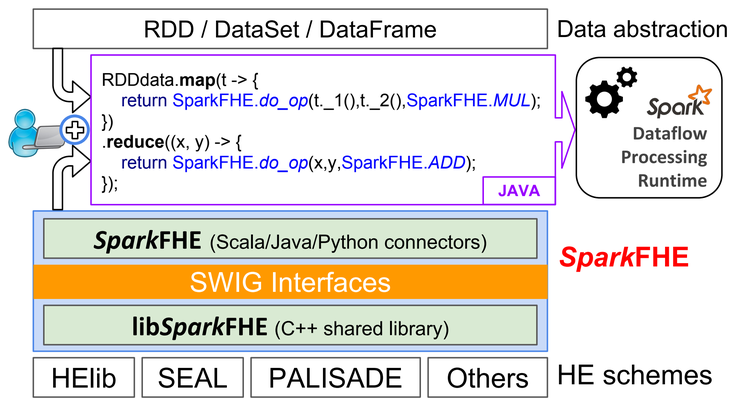

Figure 5: Secure data analytics enabled by SparkFHE

An abstraction layer that offers cryptographic primitives from different homomorophic encryption schemes without the low-level complexities.

- Supports homomorphic primitives:
  - Key Generation.
  - Encryption and Decryption.
  - $\oplus$, $\otimes$, Subtraction.
  - Comparison.
- Supports high-level homomorphic algorithms:
  - Dot Product.
  - Elementwise Product.
  - Word Count.
- Supports different data storage solutions, such as AWS S3.
- Provides native Spark semantics for writing secure data analytics algorithms.

Results

– Developed a working prototype of the SparkFHE framework which can perform evaluation of homomorphic functions in Spark.

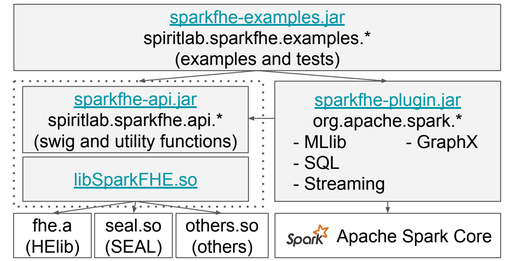

Figure 6: SparkFHE Architecture

– Developed a Spark MLlib module (e.g., `ElementwiseProduct`) to support homomorphic evaluation of machine learning functions.

```
val dataFrame = spark.createDataFrame(Seq(
    ("a", CtxtVectors.dense(one_ctxt, zero_ctxt, one_ctxt)),
    ("b", CtxtVectors.dense(zero_ctxt, one_ctxt, one_ctxt))
)).toDF("id", "vector")
val transformingVector = CtxtVectors.dense(zero_ctxt, one_ctxt, zero_ctxt)
val transformer = new ElementwiseProduct()
                    .setScalingVec(transformingVector)
                    .setInputCol("vector")
                    .setOutputCol("transformedVector")
transformer.transform(dataFrame).show()
```

Future Work

Enhancing new homomorphic primitives

- Support additional HE libraries (SEAL, PALISADE, etc).
- Support more efficient homomorphic operations (ciphertext packing, Chinese remainder theorem, etc).
- Support multi-layer parallelization techniques to accelerate homomorphic evaluation.

Advancing SparkFHE as a framework

- Support execution of homomorphic tasks in larger cluster (with resource and task allocation by YARN, Mesos, Kubernetes, etc).
- Support additional machine learning algorithms.

References

[1] V. Vaikuntanathan, "Computing blindfolded: New developments in fully homomorphic encryption," in *Proceedings of FOCS*, 2011.

[2] M. Zaharia *et al.*, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," in *Proceedings of USENIX NSDI*, 2012, pp. 2–2.

Acknowledgements: Kristin Lauter, Wei Dai, Tom Bottom, Jason Carr, Bharath Suresh Modhipalli, and Pranav Rane

RIT

Follow the **SparkFHE** project at https://github.com/SpiRITlab or http://sparkfhe.slack.com
Try it out at https://github.com/SpiRITlab/SparkFHE-Examples/wiki