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.

Distributed and Parallel Computing Frameworks
- 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 MapReduce 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 lineage of datasets which improves in-memory processing.
  - 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).

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.

Overview of SparkFHE
An abstraction layer that offers cryptogaphic primitives from different homomorphic encryption schemes without the low-level complexities.
- Supports homomorphic primitives:
  - Key Generation.
  - Encryption and Decryption.
  - Addition and Subtraction.
- 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.

Future Work
- Developed a working prototype of the SparkFHE framework which can evaluate homomorphic functions in Spark.
- Developed a Spark Multi-Lib module (e.g., ElementwiseProduct) to support homomorphic evaluation of machine learning functions.

References