SparkFHE: Sharding for Batched Ciphertexts in Homomorphic Encryption

Student: Nabiha Raza  
nr6024@rit.edu

Advisor: Dr. Peizhao Hu  
ph@cs.rit.edu
**Key Definitions**

### Homomorphic Encryption (HE):
HE differs from typical encryption methods in that it allows computation to be performed directly on encrypted data without requiring access to a secret key.

### Cipher Texts:
Ciphertext is the result of encryption performed on plaintext using an algorithm, called a cipher.

### Sharding:
Sharding is a type of database partitioning that separates very large databases into smaller, faster, more easily managed parts called data shards.
- Implementation of Sharding methods in Apache Spark to increase efficiency and loss of data (via Reed Solomon Codes)
- The data to be sharded consists of batched Ciphertexts created due to Homomorphic Encryption
Motivation

- Encrypting data end to end using HE with sharding batched cipher texts to protect data integrity
- Implementing Sharding in Apache Spark to increase data retrieval, faster index searches and fewer database locks.
- Electronic voting systems
- Analyzing medical data
- Enabling private queries in search engines
- Medical data (HIPPA Compliant)
Final Deliverables
- Implementing sharding in Apache Spark - SparkFHE

Results & Evaluation:
- Analysing the size of shards and evaluating the optimal number of shards.
- Checking the data retrieval time in shards vs non-shared data

Background & Related Work:
- SparkFHE which uses Apache Spark as a basic cloud computational framework and Fully Homomorphic Encryption (FHE) uses Apache spark to provide encrypted computations on the cloud without decrypting.
- Mult-key Homomorphic Encryption (MKHE)
- Database Sharding: To Provide Fault Tolerance and Scalability of Big Data on the Cloud
Milestones

- **MileStone 1**
  - Implementation of Reed Solomon Codes for Error Corrective Codes for Retrieval
  - **Week - 4**

- **MileStone 2**
  - Implementing Sharding Methods and run Evaluation Tests
  - **Week-8**

- **MileStone 3**
  - Integrating Sharding with SparkFHE and test with bached ciphertexts
  - **Week-12**

- **MileStone 4**
  - Final Testing and Bug Fixing, Report, Presentation and poster
  - **Final**

---

- Key Definitions
- Project Summary
- Motivation
  - Bigger Picture
  - Narrower Focus
  - Examples
- Proposed Work
  - Results and Evaluation
  - Background and Related Work

- **Milestones**
- References
References

- Analysing and Improving Shard Allocation Protocols for Sharded Blockchains
- Sharding by Hash Partitioning
  https://www.scitepress.org/Papers/2015/53762/53762.pdf
- Secure Computation on Encrypted data
  https://www.cs.rit.edu/~ph/PrivateComputation
- SparkFHE Source code: https://github.com/SpiRITlab/SparkFHE-Examples
Thank You

Any Questions?