SparkFHE: Sharding for Batched Ciphertexts in Homomorphic Encryption

Student: Nabiha Raza
nr6024@rit.edu

Advisor: Dr. Peizhao Hu
ph@cs.rit.edu
- Reminder of the Project
- Reminder of Accomplishments
  - Milestone Summary
    - Goal
    - Progress
      - Accomplishments & Technical Details
      - Example
      - Challenges
    - Next Steps
- References
Reminder of Project

Summary

- Implement Sharding in Apache Spark.
- Trying to achieve high fault tolerance if in case of disk failure.
- End-to-End Encryption (FHE) with disk fault recovery have several applications.
  - Patient Records
  - Defense Data
  - Voting Information
Milestone 1 Summary

- Research/Implementation on Reed Solomon Codes and the mathematics behind it.
  - Understanding Galois Field Arithmetic
  - Understanding Vandermonde Matrix
  - Brushing up on Gaussian Elimination
- Reminder of the Project
- Reminder of Accomplishments
  - **Milestone Summary**
    - Goal
    - Progress
      - Accomplishments & Technical Details
      - Example
      - Challenges
    - Next Steps
- References
Goal

Original Activities Planned for Milestone 2

- Research/Implementation for Sharding Methods
  - Reed Solomon Codes (*Recovery upto One Disk*)

- *RAID-6* Algorithms Exploration
  - EVENODD, RDP
  - X-Code (*Recovery upto Two Disks*)
  - FHE-XCODE - Algorithm Implementation with *FHE* - *(Novel Idea)*

*RAID: Redundant Array of Independent Disks
*FHE: Fully Homomorphic Encryption
- Reminder of the Project
- Reminder of Accomplishments
  - Milestone Summary
    - Goal
    - **Progress**
      - Accomplishments & Technical Details
      - Example
      - Challenges
    - Next Steps
- References
Findings/Contributions of the paper

- **Erasure Coding for Storage Applications**
  - Discussed old coding theories (e.g., Reed Solomon Codes)
  - Discussed new mechanisms such as RAID-6 Codes
  - RAID-6 Codes recovers two storage disks (increased reliability)

- **Related or not?**
  - Introduced latest RAID-6 Algorithm known as X-Code

Findings/Contributions of the paper

- **X-Code with two disks recovery**
  - Enhancement in Reed Solomon (RS) Codes
  - Two replicated disks are used
  - Simple XOR operation instead of Galois Field Complex arithmetic used in RS Codes

- **Related or not?**
  - Algorithm explained

Formulated New Algorithm - “FHE-XCODE”

- **X-Code with Fully Homomorphic Encryption**
  - Combines XCode implemented so that it can be used in FHE
  - Additional Inexpensive operations on storage sets.

- **Solves What?**
  - Direct implementation is naive and computationally very expensive.
  - High reliability (storage recovery) with End-to-End Security (FHE).
  - Many applications where sensitive data is handled and manipulated (patient records, military records)
Example
Example of XCode-FHE

Disk 1

Replica Disk 1

\( p_{rc} = \text{Parity} \)
\( d_{rc} = \text{disk element} \)
\( D_i = \text{Disk v} \)
**Example XCode-FHE**

<table>
<thead>
<tr>
<th>(D_0)</th>
<th>(D_1)</th>
<th>(D_2)</th>
<th>(D_3)</th>
<th>(D_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(p_{3,0})</td>
<td>(p_{3,1})</td>
<td>(d_{0,0})</td>
<td>(d_{1,0})</td>
<td>(d_{2,0})</td>
</tr>
<tr>
<td>(p_{3,1})</td>
<td>(d_{0,1})</td>
<td>(d_{1,1})</td>
<td>(d_{2,1})</td>
<td>(p_{4,0})</td>
</tr>
<tr>
<td>(d_{0,0})</td>
<td>(p_{4,1})</td>
<td>(d_{1,2})</td>
<td>(d_{2,2})</td>
<td>(d_{3,2})</td>
</tr>
<tr>
<td>(d_{1,2})</td>
<td>(d_{0,2})</td>
<td>(d_{1,3})</td>
<td>(d_{2,3})</td>
<td>(d_{3,3})</td>
</tr>
<tr>
<td>(d_{2,3})</td>
<td>(d_{0,3})</td>
<td>(d_{1,4})</td>
<td>(d_{2,4})</td>
<td>(p_{4,3})</td>
</tr>
<tr>
<td>(p_{4,0})</td>
<td>(p_{4,1})</td>
<td>(p_{4,2})</td>
<td>(p_{4,3})</td>
<td>(p_{4,4})</td>
</tr>
</tbody>
</table>

**Apply Column Rotation**

- \(p_{3,1} = d_{0,0} \oplus d_{1,3} \oplus d_{2,2}\)
- \(d_{0,1} = d_{1,0} \oplus p_{3,3} \oplus d_{2,4}\)
- \(d_{1,1} = d_{0,2} \oplus p_{3,4} \oplus d_{2,0}\)
- \(d_{2,1} = p_{3,0} \oplus d_{1,2} \oplus d_{0,3}\)
- \(p_{4,1} = d_{2,0} \oplus d_{0,3} \oplus d_{1,4}\)
**Example XCode-FHE**

**Step 1**

<table>
<thead>
<tr>
<th>$p_{4,0}$</th>
<th>$p_{3,1}$</th>
<th>$d_{2,2}$</th>
<th>$d_{1,3}$</th>
<th>$d_{0,4}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_{0,0}$</td>
<td>$p_{4,1}$</td>
<td>$p_{3,2}$</td>
<td>$d_{2,3}$</td>
<td>$d_{1,4}$</td>
</tr>
<tr>
<td>$d_{1,0}$</td>
<td>$d_{0,1}$</td>
<td>$p_{4,2}$</td>
<td>$p_{3,3}$</td>
<td>$d_{2,4}$</td>
</tr>
<tr>
<td>$d_{2,0}$</td>
<td>$d_{1,1}$</td>
<td>$d_{0,2}$</td>
<td>$p_{4,3}$</td>
<td>$p_{3,4}$</td>
</tr>
<tr>
<td>$p_{3,0}$</td>
<td>$d_{2,1}$</td>
<td>$d_{1,2}$</td>
<td>$d_{0,3}$</td>
<td>$p_{4,4}$</td>
</tr>
</tbody>
</table>

$D^L_0$ $D^L_1$ $D^L_2$ $D^L_3$ $D^L_4$

$C^L_0$ $C^L_1$ $C^L_2$ $C^L_3$ $C^L_4$

$p_{r,c}$ = Parity
$d_{r,c}$ = disk element
$D_i$ = Disk
$C_i$ = Ciphertext

$$d_{2,3} = d_{1,4} + p_{3,2} + d_{3,2}$$

$$d_{1,1} = d_{0,2} + p_{3,4} + d_{2,0}$$

$$d_{2,1} = p_{4,2} + d_{1,0} + d_{0,4}$$

$$d_{1,3} = d_{0,2} + p_{4,0} + d_{2,4}$$
Example XCode-FHE

- Reminder of the Project
- Reminder of Accomplishments
- Milestone Summary
- Goal
- Progress
- Accomplishments & Technical Details
- Example
- Challenges
- Next Steps
- References B

- p_{r,c} = Parity
- d_{r,c} = disk element
- D = Disk
- C_i = Ciphertext

**Step 2**

\[
\begin{align*}
p_{3,1} &= d_{0,4} + d_{1,3} + d_{2,2} \\
p_{3,3} &= d_{1,0} + d_{2,2} + d_{3,4} \\
\end{align*}
\]

\[
\begin{align*}
d_{2,3} &= d_{1,4} + p_{3,2} + d_{3,2} \\
d_{1,1} &= d_{0,2} + p_{3,4} + d_{2,0} \\
\end{align*}
\]

\[
\begin{align*}
p_{3,0} &= d_{1,4} + d_{2,2} + d_{0,4} \\
p_{4,3} &= d_{2,2} + d_{0,1} + d_{2,4} \\
\end{align*}
\]
Example XCode-FHE

- Reminder of the Project
- Reminder of Accomplishments
- Milestone Summary
- Goal
- Progress
  - Accomplishments & Technical Details
- Example
- Challenges
- Next Steps
- References B

\[ d_{r,c} = \text{Parity} \]
\[ d_{r,c} = \text{disk element} \]
\[ D_i = \text{Disk} \]
\[ C_i = \text{Ciphertext} \]

**Step 3**

\[
\begin{align*}
p_{3,1} &= d_{0,4} + d_{1,3} + d_{2,2} \\
p_{3,2} &= d_{0,3} + d_{1,2} + d_{2,1} \\
p_{3,3} &= d_{0,2} + d_{1,1} + d_{2,0} \\
p_{3,4} &= d_{0,1} + d_{1,0} + d_{2,2} \\
p_{4,1} &= d_{4,1} + d_{3,2} + d_{2,3} + d_{1,4} \\
p_{4,2} &= d_{4,2} + d_{3,3} + d_{2,4} + d_{1,0} \\
p_{4,3} &= d_{4,3} + d_{3,4} + d_{2,0} + d_{1,1} \\
p_{4,4} &= d_{4,4} + d_{3,0} + d_{2,1} + d_{1,2} \\
p_{5,0} &= d_{5,0} + d_{4,0} + d_{3,0} + d_{2,0} + d_{1,0} + d_{0,0} \\
p_{5,1} &= d_{5,1} + d_{4,1} + d_{3,1} + d_{2,1} + d_{1,1} + d_{0,1} \\
p_{5,2} &= d_{5,2} + d_{4,2} + d_{3,2} + d_{2,2} + d_{1,2} + d_{0,2} \\
p_{5,3} &= d_{5,3} + d_{4,3} + d_{3,3} + d_{2,3} + d_{1,3} + d_{0,3} \\
p_{5,4} &= d_{5,4} + d_{4,4} + d_{3,4} + d_{2,4} + d_{1,4} + d_{0,4} \\
\end{align*}
\]
Challenges

- **Understanding the Algorithm to implement with FHE**
  - Several tries attempted to implement XCode with FHE
  - Coded to test implementation and recovery process
  - Actual implementation requires specialised hardware
Next Steps towards Milestones

- Milestone 1: Research/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

  - Final

How to achieve?

- Reminder of the Project
- Reminder of Accomplishments
- Milestone Summary
- Goal
- Progress
- Accomplishments & Technical Details
- Example
- Challenges

Next Steps

References
Next Steps towards Milestones

- Next Steps

- MileStone 3
  Integrating Sharding with SparkFHE and test with bached ciphertexts

- How to achieve?
  - Implement basic algorithm (XCode-FHE) *(completed)*
  - Test on multiple metrics and test algorithm
    - Time taken to Encode/Decode
    - CPU utilization
    - Memory utilization
    - Access rate to the shards

- Further Extension
  - Further Extension of FHE-XCODE for better memory utilization-* (Novel Idea)*
References


- Chapter 1: An Introduction to Reed-Solomon Codes –Stephen B. Wicker (https://people.cs.clemson.edu/~westall/851/rs-code.pdf)
Thank You

Any Questions?