Secure Intersection with MapReduce

R. Ciucanu\textsuperscript{1}  M. Giraud\textsuperscript{2}  P. Lafourcade\textsuperscript{2}  L. Ye\textsuperscript{3}

\textsuperscript{1}LIFO, INSA Centre Val de Loire
Université d’Orléans

\textsuperscript{2}School of Computer Science and Technology
Harbin Institute of Technology, China

26 July 2019 @ SECRYPT, Prague
Big Data

Cloud Service Provider (CSP)
## Model 1

### Application

Avoid double submissions in conferences

### Mutual Private Set Intersection (PSI)

<table>
<thead>
<tr>
<th>Participants List</th>
<th>$A$</th>
<th>$B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result</td>
<td>$A \cap B$</td>
<td>$A \cap B$</td>
</tr>
</tbody>
</table>
Model 2

Application

FBI wants to detect suspicious passengers of an airline company

One-way PSI

<table>
<thead>
<tr>
<th>Passengers List</th>
<th>$A$</th>
<th>$B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result</td>
<td>$A \cap B$</td>
<td>$\emptyset$</td>
</tr>
</tbody>
</table>
**Model 3**

### Application

Interpol wants the most dangerous persons from FBI and MI6

### Our PSI Model

<table>
<thead>
<tr>
<th>Suspects Lists</th>
<th>$A$</th>
<th>$B$</th>
<th>$A \cap B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
<td>$A \cap B$</td>
</tr>
</tbody>
</table>
Example

Suspects Lists

- Alice
- Cesar
- Mallory

- Oscar
- Bob
- Mallory

Intersection List

- Mallory
Outline

Motivations

MapReduce

Intersection with MapReduce

Security Model and Cryptographic Tools

Secure Intersection with MapReduce

Performance Evaluation

Conclusion
Outline

Motivations

MapReduce

Intersection with MapReduce

Security Model and Cryptographic Tools

Secure Intersection with MapReduce

Performance Evaluation

Conclusion
MapReduce

MapReduce Environment

Take care of
- Partitioning input data
- Scheduling program execution on a set of machines
- Handling machine failures

Programmer

Specify
- Map and Reduce functions
- Input files

MapReduce Example
MapReduce in 3 Steps

1. Map tasks

   Input: ID of chunk
   Output: *key-value* pairs

2. Master Controller

   ▶ Key-value pairs aggregated and sorted by key
   ▶ Pairs with same key sent to the same Reduce task

3. Reduce tasks

   Input: One key
   Output: Combine values associated to the key
Outline

Motivations

MapReduce

Intersection with MapReduce

Security Model and Cryptographic Tools

Secure Intersection with MapReduce

Performance Evaluation

Conclusion
Intersection with MapReduce

<table>
<thead>
<tr>
<th>3 participants</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NSA</strong></td>
</tr>
<tr>
<td>F654</td>
</tr>
<tr>
<td>U840</td>
</tr>
<tr>
<td>X098</td>
</tr>
<tr>
<td><strong>GCHQ</strong></td>
</tr>
<tr>
<td>F654</td>
</tr>
<tr>
<td>M349</td>
</tr>
<tr>
<td>P027</td>
</tr>
<tr>
<td><strong>Mossad</strong></td>
</tr>
<tr>
<td>F654</td>
</tr>
<tr>
<td>M349</td>
</tr>
<tr>
<td>U840</td>
</tr>
</tbody>
</table>

Intersection with MapReduce

Map

NSA

GCHQ

Mossad

Data owners

Map

NSA

GCHQ

Mossad

Master Controller

Public cloud

Reduce

Key

F654

Values

F654

M349

Values

M349

P027

Value

P027

U840

Value

U840

X098

Value

X098

User

Interpol

F654

Reduce function

It returns value only if: \#values = \#participants
Outline

Motivations
MapReduce
Intersection with MapReduce
Security Model and Cryptographic Tools
Secure Intersection with MapReduce
Performance Evaluation
Conclusion
Security Model

Cloud is **honest-but-curious**

Without security, Cloud learns:

- Content of relations
- Intersection result
Cryptographic Tools

Pseudorandom function

\[ f : \mathcal{K} \times \mathcal{D} \rightarrow \mathcal{R} \]

- Deterministic
- Indistinguishable from a random function

Notation

\[ [m]_k = f(k, m) \]
Cryptographic Tools

Asymmetric encryption scheme

- $(pk, sk) \leftarrow \mathcal{G}(\lambda)$
- $c \leftarrow \mathcal{E}(pk, m)$
- $m \leftarrow \mathcal{D}(sk, c)$

$$\mathcal{D}(sk, \mathcal{E}(pk, m)) = m$$

Notation

$$\{m\} = \mathcal{E}(pk, m)$$
Outline

Motivations

MapReduce

Intersection with MapReduce

Security Model and Cryptographic Tools

Secure Intersection with MapReduce

Performance Evaluation

Conclusion
Secure Intersection with MapReduce

Setting

- $n$ relations: $R_1, \ldots, R_n$
- $R_1$ has: $k_1, \ldots, k_n$ PRF secret keys, and $pk$
- $R_i$ (for $2 \leq i \leq n$) has: $k_1$ and $k_i$

Preprocessing

- One main relation using the public key of the final user
  For each element $x$, compute the key-value pair:
  $$\left( [x]_{k_1}, \{x\} \oplus (\oplus_{i=2}^{n} [x]_{k_i}) \right)$$
- Other relation compute the key-value pair:
  $$([x]_{k_1}, [x]_{k_i})$$
## Secure Intersection with MapReduce

### Processed relations

**NSA**

- $([F654]_{k_1}, \{F654\} \oplus [F654]_{k_2} \oplus [F654]_{k_3})$
- $([U840]_{k_1}, \{U840\} \oplus [U840]_{k_2} \oplus [U840]_{k_3})$
- $([X098]_{k_1}, \{X098\} \oplus [X098]_{k_2} \oplus [X098]_{k_3})$

**GCHQ**

- $([F654]_{k_1}, [F654]_{k_2})$
- $([M349]_{k_1}, [M349]_{k_2})$
- $([P027]_{k_1}, [P027]_{k_2})$

**Mossad**

- $([F654]_{k_1}, [F654]_{k_3})$
- $([M349]_{k_1}, [M349]_{k_3})$
- $([U840]_{k_1}, [U840]_{k_3})$
Secure Intersection with MapReduce

| Data owners | | Public cloud | | User (sk, pk) |
|-------------|-------------|-------------|-------------|
| NSA*        | Map         | Key: [F654] | Reduce: Interpol {F654} |
| GCHQ*       | Master Controller | Values: [F654]_{k_1} ⊕ [F654]_{k_2} ⊕ [F654]_{k_3} |
| Mossad*     | | Key: [M349]_{k_1} |
|             | | Values: [M349]_{k_2} [M349]_{k_3} |
|             | | Key: [P027]_{k_1} |
|             | | Value: [P027]_{k_2} |
|             | | Key: [U840]_{k_1} |
|             | | Values: [U840]_{k_2} ⊕ [U840]_{k_3} |
|             | | Key: [X098]_{k_1} |
|             | | Values: [X098]_{k_2} ⊕ [X098]_{k_3} |
Outline

Motivations

MapReduce

Intersection with MapReduce

Security Model and Cryptographic Tools

Secure Intersection with MapReduce

Performance Evaluation

Conclusion
# Experimental Results

## Settings

- Hadoop 3.2.0 / Standalone mode / Streaming
- Ubuntu 16.04 LTS
- Map and Reduce functions in Go

## Hardware

- 4 CPU @ 2.4 GHz
- 80 Gb of disk
- 8 Gb of RAM

## Experiments

1. Varying the number of tuples
2. Varying the number of intersected relations
Results: Varying the Number of Tuples

![Graph showing CPU time vs. number of tuples for Standard protocol and Secure Intersection.]

- **Standard protocol**: Red dots
- **Secure Intersection**: Green squares

---

Results: Varying the Number of Intersected Relations

- Standard protocol
- Secure Intersection

Number of intersected relations (500,000 tuples / relation)

CPU time (s)

Outline

Motivations

MapReduce

Intersection with MapReduce

Security Model and Cryptographic Tools

Secure Intersection with MapReduce

Performance Evaluation

Conclusion
Conclusion and Future Works

Conclusion

- Design of secure intersection with MapReduce
- Collision resistance
- Practical scalability

Future Works

- Apache Spark environment
- Malicious model
Thank you for your attention.

Any questions?

pascal.lafourcade@uca.fr