Trident: Efficient 4PC Framework for Privacy Preserving Machine Learning

Harsh Chaudhari*, Rahul Rachuri^ and Ajith Suresh*

* Indian Institute of Science (IISc), Bangalore
^ Aarhus University, Denmark

Ajith Suresh
CrIS Lab, IISc

https://www.csa.iisc.ac.in/~cris
Outline

- Privacy Preserving Machine Learning (PPML)
- Secure Multi-party Computation (MPC)
- Overview of Trident Protocol
- Benchmarking Results
Machine Learning (ML) Prediction – An Abstraction

Jasmine (Model Owner) → Model Parameters → ML Algorithm → Query → Result → Aladdin (Client)

Privacy ??

AJITH SURESH | CRYPTOGRAPHY AND INFORMATION SECURITY LAB, CSA, IISC
26-02-2020
Machine Learning (ML) Prediction – An Abstraction

Jasmine (Model Owner)

Aladdin (Client)

Model Parameters

ML Algorithm

Query

Result

Privacy ??
Machine Learning (ML) Prediction – An Abstraction

Jasmine (Model Owner)

Model Parameters

ML Algorithm

Query

Result

Privacy ??

Aladdin (Client)
Privacy Preserving Machine Learning (PPML)

MPC meets ML

Jasmine (Model Owner)

Aladdin (Client)
Secure Multi-party Computation (MPC) [Yao’82]

A set of parties with private inputs wish to compute some joint function of their inputs.

Goals of MPC:

- **Correctness** – Parties should correctly evaluate the function output.
- **Privacy** – Nothing more than the function output should be revealed.
Secure Multi-party Computation (MPC) [Yao’82]

Trusted Third Party (TTP)
Trusted Third Party (TTP)
MPC emulates TTP
TRIDENT PROTOCOL
A new 4PC protocol over ring in the pre-processing model
Trident protocol

- A new **4PC protocol** over ring in the pre-processing model
  - 4 parties
  - Honest majority
  - At most 1 corruption
Trident protocol

- A new 4PC protocol over ring in the **pre-processing** model
  
  - Data independent pre-processing
  - Fast online phase
Pre-processing
Online
Shares of
Trident protocol

- A new 4PC protocol over ring in the pre-processing model
- Malicious security with guarantee of fairness
Trident protocol

- A new 4PC protocol over ring in the pre-processing model
- **Malicious** security with guarantee of fairness

Corrupt parties arbitrarily deviate
Trident protocol

- A new 4PC protocol over ring in the pre-processing model
- Malicious security with guarantee of **fairness**

Honest parties get output whenever corrupt parties get output
## Multiplication \((x \cdot y)\)

<table>
<thead>
<tr>
<th>Ref</th>
<th>Pre-processing (#elements)</th>
<th>Online (#elements)</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Araki et al’17 (3PC)</td>
<td>12</td>
<td>9</td>
<td>Abort</td>
</tr>
</tbody>
</table>
## Multiplication ($x \cdot y$)

<table>
<thead>
<tr>
<th>Ref</th>
<th>Pre-processing (#elements)</th>
<th>Online (#elements)</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Araki et al’17 (3PC)</td>
<td>12</td>
<td>9</td>
<td>Abort</td>
</tr>
<tr>
<td>ASTRA (3PC)</td>
<td>21</td>
<td>4</td>
<td>Fair</td>
</tr>
</tbody>
</table>
## Multiplication \((x \cdot y)\)

<table>
<thead>
<tr>
<th>Ref</th>
<th>Pre-processing (#\text{elements})</th>
<th>Online (#\text{elements})</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Araki et al.’17 (3PC)</td>
<td>12</td>
<td>9</td>
<td>Abort</td>
</tr>
<tr>
<td>ASTRA (3PC)</td>
<td>21</td>
<td>4</td>
<td>Fair</td>
</tr>
<tr>
<td>Gordon et al.’18 (4PC)</td>
<td>2</td>
<td>4</td>
<td>Abort</td>
</tr>
</tbody>
</table>
## Multiplication ($x \cdot y$)

<table>
<thead>
<tr>
<th>Ref</th>
<th>Pre-processing (#elements)</th>
<th>Online (#elements)</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Araki et al.’17 (3PC)</td>
<td>12</td>
<td>9</td>
<td>Abort</td>
</tr>
<tr>
<td>ASTRA (3PC)</td>
<td>21</td>
<td>4</td>
<td>Fair</td>
</tr>
<tr>
<td>Gordon et al.’18 (4PC)</td>
<td>2</td>
<td>4</td>
<td>Abort</td>
</tr>
<tr>
<td><strong>Trident</strong></td>
<td><strong>3</strong></td>
<td><strong>3</strong></td>
<td><strong>Fair</strong></td>
</tr>
</tbody>
</table>
Trident protocol

- A new 4PC protocol over ring in the pre-processing model
- Malicious security with guarantee of fairness
- Efficient Mixed World Conversions
Mixed World Conversions

- Boolean World
  - Comparison, Bit Extraction …
Mixed World Conversions

Boolean World
- Comparison, Bit Extraction …

Arithmetic World
- Addition, Multiplication …
Mixed World Conversions

- **Boolean World**
  - Comparison, Bit Extraction …

- **Arithmetic World**
  - Addition, Multiplication …

- **Garbled World**
  - Division over rings …
Mixed World Conversions

<table>
<thead>
<tr>
<th>Online Rounds</th>
<th>Online Communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up to 7x</td>
<td>2x - 67x</td>
</tr>
</tbody>
</table>

Range of improvement over ABY3
Mixed World Conversions – An Example

\[
\min(x_1 + x_2, x_3) = x_4
\]
Mixed World Conversions – An Example

\[ x_1 + x_2 \]
Mixed World Conversions – An Example

Arithmetic

\[ x_1 + x_2 \]
Mixed World Conversions – An Example

Arithmetic

\[ x_1 + x_2 \]

\[ \min(x_1 + x_2, x_3) \]
Mixed World Conversions – An Example

Arithmetic

\[ x_1 + x_2 \]

Boolean

\[ \min(x_1 + x_2, x_3) \]
Mixed World Conversions – An Example

Arithmetic

\[ x_1 + x_2 \]

Boolean

\[ \min(x_1 + x_2, x_3) \]
Mixed World Conversions – An Example

Arithmetic

\[ x_1 + x_2 \]

Boolean

\[ \min(x_1 + x_2, x_3) \]

A2B

B2G

A2G

\[ x_4 \]
Mixed World Conversions – An Example

Arithmetic

\[ x_1 + x_2 \]

Boolean

\[ \min(x_1 + x_2, x_3) \]

\[ \min(x_1 + x_2, x_3) \div x_4 \]
Mixed World Conversions – An Example

Arithmetic

\[ x_1 + x_2 \]

Boolean

\[ \min(x_1 + x_2, x_3) \]

Garbled

\[ \min(x_1 + x_2, x_3) \div x_4 \]
Mixed World Conversions – An Example

Arithmetic

\[ x_1 + x_2 \]

\[ \min(x_1 + x_2, x_3) \]

\[ x_4 \]

Boolean

\[ \min(x_1 + x_2, x_3) \]

\[ \min(x_1 + x_2, x_3) \div x_4 \]

Garbled

\[ x_4 \]
Trident protocol

- A new 4PC protocol over ring in the pre-processing model
- Malicious security with guarantee of fairness
- Efficient Mixed World Conversions
- Special tools for PPML
Dot Product

\[ X \mathbin{\bigcirc} Y = \sum_{i=1}^{d} x_i \cdot y_i \]

<table>
<thead>
<tr>
<th>Ref</th>
<th>Pre-processing (#elements)</th>
<th>Online (#elements)</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABY3 (3PC)</td>
<td>12d</td>
<td>9d</td>
<td>Abort</td>
</tr>
</tbody>
</table>

\(d\) – #elements in each vector
## Dot Product

$$X \bullet Y = \sum_{i=1}^{d} x_i \cdot y_i$$

<table>
<thead>
<tr>
<th>Ref</th>
<th>Pre-processing (#elements)</th>
<th>Online (#elements)</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABY3 (3PC)</td>
<td>12d</td>
<td>9d</td>
<td>Abort</td>
</tr>
<tr>
<td>ASTRA (3PC)</td>
<td>21d</td>
<td>2d+2</td>
<td>Fair</td>
</tr>
</tbody>
</table>

$d$ – #elements in each vector
## Dot Product

\[ X \cdot Y = \sum_{i=1}^{d} x_i \cdot y_i \]

<table>
<thead>
<tr>
<th>Ref</th>
<th>Pre-processing (#elements)</th>
<th>Online (#elements)</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABY3 (3PC)</td>
<td>12d</td>
<td>9d</td>
<td>Abort</td>
</tr>
<tr>
<td>ASTRA (3PC)</td>
<td>21d</td>
<td>2d+2</td>
<td>Fair</td>
</tr>
<tr>
<td><strong>Trident</strong></td>
<td><strong>3</strong></td>
<td><strong>3</strong></td>
<td><strong>Fair</strong></td>
</tr>
</tbody>
</table>

\[ d \text{ – \#elements in each vector} \]
Tools for PPML

- Bit Injection
- Bit to Arithmetic
- Comparison
- Fixed Point Arithmetic
- Truncation
- Dot Product
- Non-linear Activation Functions
Trident protocol

- A new 4PC protocol over ring in the pre-processing model
- Malicious security with guarantee of fairness
- Efficient Mixed World Conversions
- Special tools for PPML
- Lower monetary cost in the outsourced setting
Trident protocol

- A new 4PC protocol over ring in the pre-processing model
- Malicious security with guarantee of fairness
- Efficient Mixed World Conversions
- Special tools for PPML
- Lower monetary cost in the outsourced setting

Computation is outsourced to a set of hired servers
Benchmarking

- Implemented both Trident and ABY3, using the ENCRYPTO library.

- Benchmarked the protocols over LAN (40 Mbps) and WAN (1 Gbps) with the Google Cloud Platform.

- Servers located in West Europe, East Australia, South Asia, and South East Asia.

- For benchmarking, we used batch sizes up to 512 and feature sizes up to 1000.
## Summary of Our Benchmarking Results

<table>
<thead>
<tr>
<th>ML Algorithm</th>
<th>Improvement in terms of Online Throughput over ABY3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training</td>
</tr>
<tr>
<td>Linear Regression</td>
<td>251.84x</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>34.58x</td>
</tr>
<tr>
<td>Neural Networks</td>
<td>63.71x</td>
</tr>
<tr>
<td>Convolutional Neural Networks</td>
<td>42.81x</td>
</tr>
</tbody>
</table>

*Throughput for Training - #iterations processed by servers / minute
*Throughput for Prediction - #queries processed by servers / minute
thank you!
References


