Mystique: Efficient Conversions for Zero-Knowledge Proofs with Applications to Machine Learning

Chenkai Weng       Kang Yang
Xiang Xie          Jonathan Katz    Xiao Wang
How to Demonstrate Evasion Attacks

I can crash your model, but I won’t reveal the vulnerability unless you offer me bug bounty rewards.

Zero-Knowledge Proofs (of Knowledge)

• For a predicate $C$, a prover $P$ convinces a verifier $V$ that it knows a witness $\omega$ such that $C(\omega) = 1$, with

  • **Completeness**: if $C(\omega) = 1$, then $V$ is convinced.
  
  • **Soundness**: if $C(\omega) \neq 1$, $V$ is convinced with negligible probability.
    
    • Stronger properties required for proof of knowledge.

• **Zero-knowledge**: no information is revealed except for the output of $C(\omega)$. 
Applications for Machine Learning

ZKPs of evasion attacks

Zero Knowledge Proofs

Challenger

Private input (witness)

Public input

Public model

0.007

“Gibbon”

Model Developer

+ 0.007 X
Applications for Machine Learning

ZKPs of evasion attacks

ZKPs of correct inference

ZKPs of private benchmarks
Current Approaches are not sufficient because...  
For Large-Scale Machine Learning Tasks

- **Scalability**
  - Circuits containing billions of gates

- **Supporting ML operations**
  - Neural networks involving linear/non-linear layers

- **Efficiency**
  - Meaningful applications run in short time

- **Ease-of-use**
  - Cryptographic protocols behind simple interface
Our result

- Interactive zero-knowledge proof protocol for machine learning tasks.
- Prove private inference using ResNet-101 model (42.5 million model parameters) in 336 s.
- Optimizations for Sigmoid, Max Pooling, ReLU, SoftMax and Batch normalization.
  - Easily extend to other operations.
Key Contribution 1
Conversion Between
*Boolean* and *Arithmetic* Circuits

• **Goal:** convert between \( ([x_0]_2, \ldots, [x_{m-1}]_2) \) and \( [x]_p \)
Key Contribution 1

Conversion Between Boolean and Arithmetic Circuits

• **Goal**: convert between \( ([x_0]_2, \ldots, [x_{m-1}]_2) \) and \([x]_p\)

• **ZK-friendly extended doubly authenticated bits** (zk-edabits)
  • Adopted from dabits/edabits in MPC settings.

Key Contribution 1

Conversion Between Boolean and Arithmetic Circuits

• **Goal**: convert between \([x_0], \ldots, [x_{m-1}]_2\) and \([x]_p\)

• **ZK-friendly extended doubly authenticated bits (zk-edabits)**
  • Adopted from dabits/edabits in MPC settings.

• **Performance**

  Average time per conversion

<table>
<thead>
<tr>
<th></th>
<th>50 Mbps</th>
<th>200 Mbps</th>
<th>1 Gbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>A2B</td>
<td>107 µs</td>
<td>45 µs</td>
<td>29 µs</td>
</tr>
<tr>
<td>B2A</td>
<td>109 µs</td>
<td>49 µs</td>
<td>33 µs</td>
</tr>
</tbody>
</table>

Key Contribution 2
Conversion from Publicly Committed Values to Privately Authenticated Values

Public non-interactive commitments

\[ c = \text{Commit}(x) \]

Private authenticated values

\[ [x]_p \]

\[ DB = (x_1, ..., x_n) \]
Key Contribution 2
Conversion from Publicly Committed Values to Privately Authenticated Values

- Hybrid commitment
  - Security reduced to random oracle and PRF.

- Performance
  Average time per conversion

<table>
<thead>
<tr>
<th></th>
<th>50 Mbps</th>
<th>200 Mbps</th>
<th>1 Gbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>C2A</td>
<td>56 µs</td>
<td>55 µs</td>
<td>55 µs</td>
</tr>
</tbody>
</table>
Key Contribution 3
Integration with TensorFlow

• Simple interface
  • *Rosetta* framework based on TensorFlow.
Other Contributions

• Matrix multiplication.
  • Freivalds algorithm.

<table>
<thead>
<tr>
<th></th>
<th>50 Mbps</th>
<th>200 Mbps</th>
<th>1 Gbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>2048×2048</td>
<td>15.19 s</td>
<td>11.30 s</td>
<td>10.39 s</td>
</tr>
</tbody>
</table>

• Conversion of fixed-point and floating-point numbers.

<table>
<thead>
<tr>
<th></th>
<th>50 Mbps</th>
<th>200 Mbps</th>
<th>1 Gbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fix2Float</td>
<td>50 μs</td>
<td>46 μs</td>
<td>46 μs</td>
</tr>
<tr>
<td>Float2Fix</td>
<td>49 μs</td>
<td>46 μs</td>
<td>46 μs</td>
</tr>
</tbody>
</table>
Benchmarks Private Inference

• Prove inference of large neural network.

<table>
<thead>
<tr>
<th>Model</th>
<th>Image</th>
<th>LeNet-5</th>
<th>ResNet-50</th>
<th>ResNet-101</th>
</tr>
</thead>
<tbody>
<tr>
<td>Private</td>
<td>Private</td>
<td>Exec. time</td>
<td>5.9 s</td>
<td>333 s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Comm.</td>
<td>16.5 MB</td>
<td>1.27 GB</td>
</tr>
<tr>
<td>Private</td>
<td>Public</td>
<td>Exec. time</td>
<td>5.5 s</td>
<td>336 s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Comm.</td>
<td>16.5 MB</td>
<td>1.27 GB</td>
</tr>
<tr>
<td>Public</td>
<td>Private</td>
<td>Exec. time</td>
<td>4.9 s</td>
<td>158 s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Comm.</td>
<td>16.4 MB</td>
<td>0.53 GB</td>
</tr>
</tbody>
</table>

Network Bandwidth: 200 Mbps
Image from CIFAR-10 dataset
Microbenchmark of ResNet-101

Public model

Private model

- Batch Normalization
- Framework overhead
- Convolution2D
- Protocol Setup
- Max Pooling
- Average Pooling
- SoftMax
- Fully Connected
- ReLU
- Private Input

0 - 500 seconds
Thank You

Full version: https://eprint.iacr.org/2021/730

Implementation:

Contact information:
Chenkai Weng, Northwestern University, ckweng@u.northwestern.edu