GC-Lite: Hiding Software, Data & Computed Values Using Lightweight Primitives

The Problem

Alice has confidential software program \( \mathcal{P} \), Bob has confidential data \( \mathcal{D} \) and they want to compute \( \mathcal{P}(\mathcal{D}) \) i.e., the output of \( \mathcal{P} \) on input \( \mathcal{D} \). But Alice does not want to reveal her \( \mathcal{P} \) to anyone, and Bob does not want to reveal his \( \mathcal{D} \) to anyone.

Motivation

• Private Cooperative Financial Forecasting:

• Privacy Preserving Medical Advice:

• Privacy Preserving Machine Learning/Data Science:

• Secure Computational Outsourcing To The Cloud:

Our Solution: GC-Lite

Computational Model: One Instruction Set Computer (OISC) where \( \mathcal{P} \) is a sequence of \( n_p \) Turing-Complete (SUBBLE) instructions with associated data array \( \mathcal{D} \) of size \( n_D \).

The Protocol:

Related Work

• Universal Turing Machine - Encrypt both function and data and run simulation on UTM. Practically infeasible.

• A combination of Homomorphic Encryption, Garbled Circuit Evaluation & ORAM. Involves modular exponentiation. Does not scale well.

• Universal Circuits - Encrypt both input circuit and data and simulate using UC. Our approach outperforms state of the art implementation.

Experimental Results

<table>
<thead>
<tr>
<th></th>
<th>GC-Lite</th>
<th>UC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Offline</td>
<td>Running</td>
</tr>
<tr>
<td>Offline Time (ms)</td>
<td>60</td>
<td>35</td>
</tr>
<tr>
<td>Running Time (ms)</td>
<td>1490</td>
<td>603</td>
</tr>
<tr>
<td>Cache Size (KB)</td>
<td>72085</td>
<td>13452</td>
</tr>
</tbody>
</table>

Table 1: A comparison of the time and communication costs of GC-Lite versus UC. All time measurements are given in milliseconds (ms) and communication costs are given in kilobytes (KB).

Figure. GC-Lite's runtime dependence on (a) Data Size, (b) Program Size, and (c) Number of Rounds. Vertical axes are in seconds.