CERIAS

The Center for Education and Research in Information Assurance and Security

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

Shoaib Khan
khan180@purdue.edu

Mikhail Atallah
matallah@purdue.edu

Qutaibah Malluhi
qmalluhi@qu.edu.qa

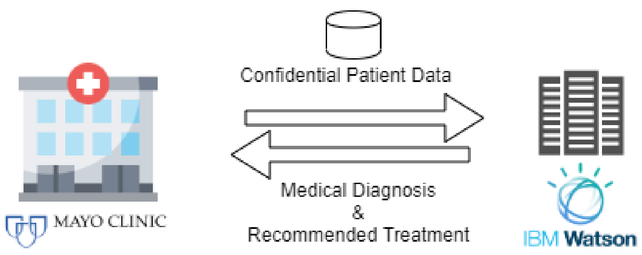
The Problem

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.

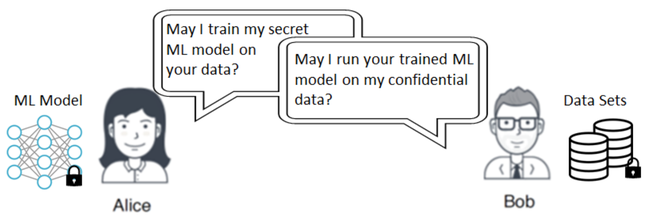
Motivation

- **Private Cooperative Financial Forecasting:**

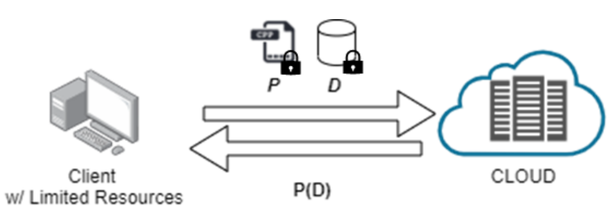
- **Privacy Preserving Medical Advice:**



- **Privacy Preserving Machine Learning/Data Science:**

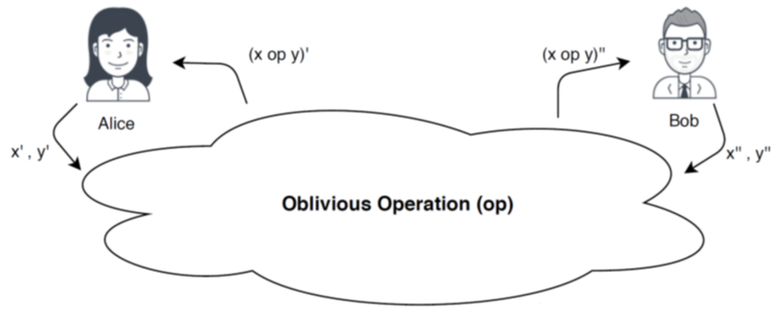


- **Secure Computational Outsourcing To The Cloud:**



Definitions

Lightweight Cryptography: *Additive Secret Sharing* is used to hide all values i.e., given some x, pick random values $x' \,\&\, x''\ s.t.\ x = x' + x''$. *Oblivious Operations* are then performed on these shared values.



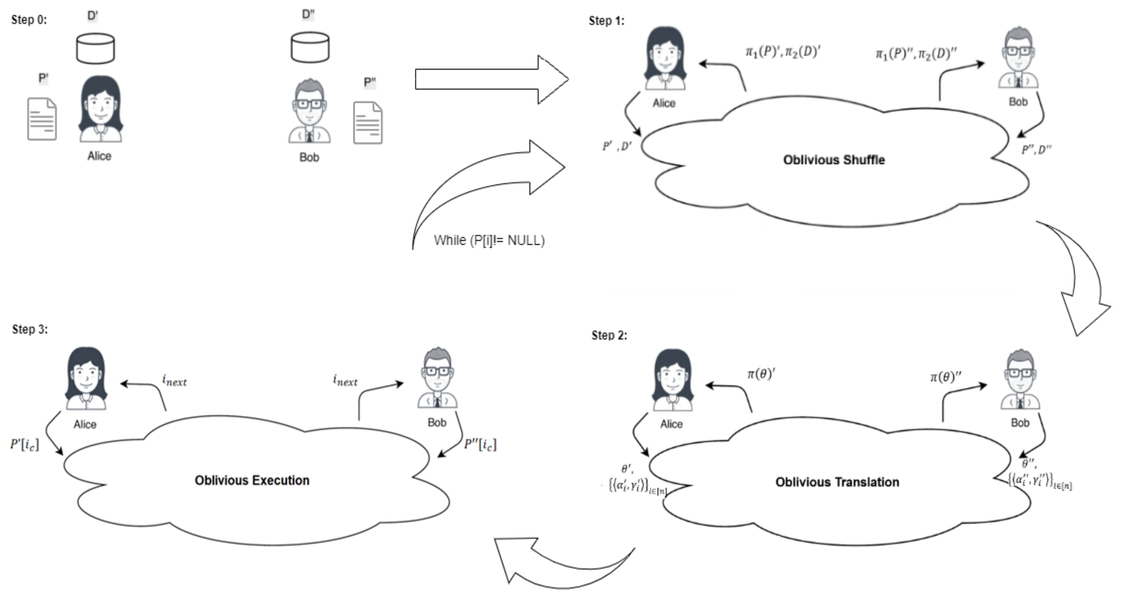
Requirement: Alice and Bob have secret shares of x and y. They compute "x op y" such that no information about x, y or "x op y" is revealed to any participant.

Note: Primitives exist where: $op \in \{+, -, *, \geq, >, ==\}$

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

Comparison of GC-Lite with UC

	GC-Lite			UC		
	Offline Time	Running Time	Comm	Offline Time	Running Time	Comm
Addition (32-bit)	81	7	2.29	660	305	667.82
Addition (64-bit)	83	7	2.29	1140	603	1401.67
Comparison (32-bit)	112	22	9.70	258	197	410.04
Multiplication (32-bit)	1517	17524	1206.99	21285	13632	35287.81

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).

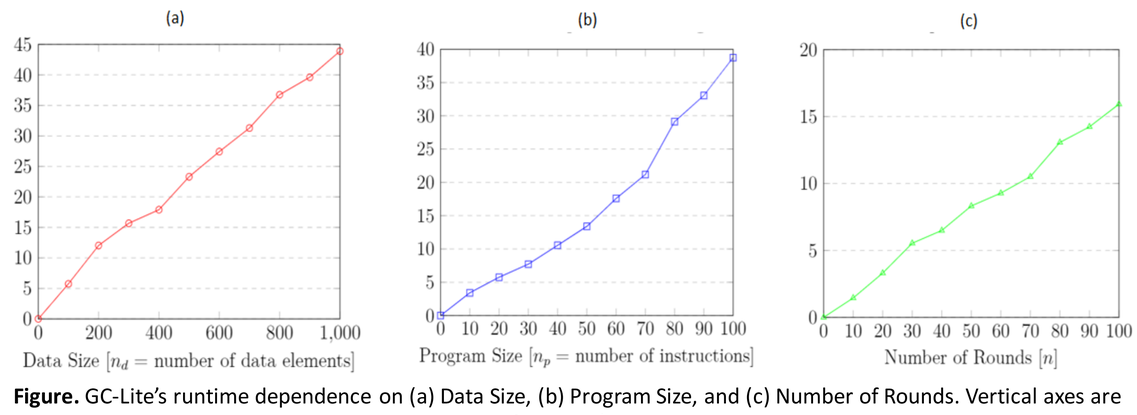


Figure. GC-Lite's runtime dependence on (a) Data Size, (b) Program Size, and (c) Number of Rounds. Vertical axes are in seconds.

PURDUE UNIVERSITY Discovery Park

CERIAS