Secure MPC as a Tool for Sensitive Data Analysis

Samuel Havron ⟨havron@virginia.edu⟩

February 21, 2016

1 Introduction

A *Secure multi-party computation* (MPC) is a protocol which allows for two or more parties to compute a function on sensitive input data provided by each party, without revealing anything about the inputs (other than what can be inferred from the revealed output result).

Most current implementations of MPC work by executing instructions in a *data-oblivious* manner, where the control flow of the program is independent of the inputs provided by each party and the program executes without any knowledge of the cleartext data it is operating on.

MPCs have practical uses for many organizations that have sensitive data sets which they cannot share, but could otherwise learn a lot by computing statistical functions on their joint data. For instance, two competitor companies may want to compute statistical functions on joint sales data and publish the results for internal company research and marketing, without allowing either competitor to uncover sensitive sales data about the other. In such situations it is important that neither company know the sales data inputs of their competitor, as they may learn secrets or be able to manipulate the data in their favor. By using an MPC protocol, the organizations can jointly compute functions of their combined sensitive data in such a manner that only the output of the program is revealed to them, with all sensitive input and intermediate results hidden.

- Capabilities of MPC (including overview of previous work, Samee's implementation, efficiency/speed comparison with non-secure MPC programs)
- Value of MPC to social scientists, researchers (with motivating example(s)) as a tool for sensitive data analysis

2 Methods

- Description of OblivCs efficiency/speed in comparison to other existing MPC implementations

- Overview of OblivC as a language, relationship to C (appeal to researchers with some C/basic programming experience)

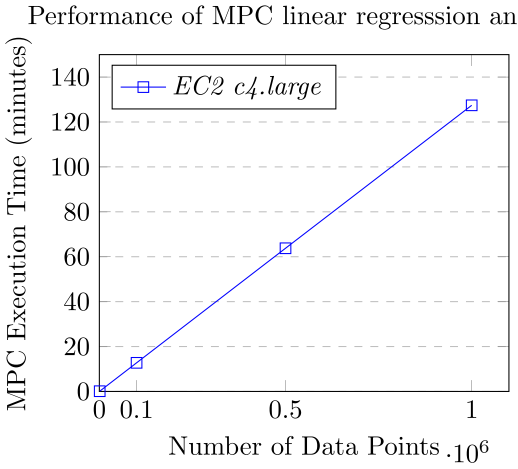
- Introduce implementation of linear regression analysis program, use code snippets and reference repo/site tutorial

3 Results

- Discuss efficiency/speed/scalability of aforementioned linear regression analysis program, provide EC2 testing data (collect more formally prior to including in final paper). Mention single-threaded implementation and limits on performance results.

- Discuss results of other motivating OblivC programs, compare to alternative MPC results

The scalability and speed of OblivC as a tool for implementing MPC programs was tested using c4.large *Elastic Compute Cloud* (EC2) nodes from *Amazon Web Services* (AWS) of Amazon.com®. Two c4.large instances were launched and connected through OblivC's API for TCP/IP connections. One node instance provided independent (x) data points, while the other provided dependent (y) data points. Data points used were 32-bit integers. Data for computation was mined from New York public health data (include link), as well as synthetic data points generated through a Python program (footnote for file location) for performance results.



- Include table with times from NY public health dataset computation.

- Include table with times from non secure MPC computation for comparison.

4 Conclusion

- Overview of secure MPC, utility/extensibility of OblivC programming, real-world usage for scientists/researchers