Secure Computation in Big Data

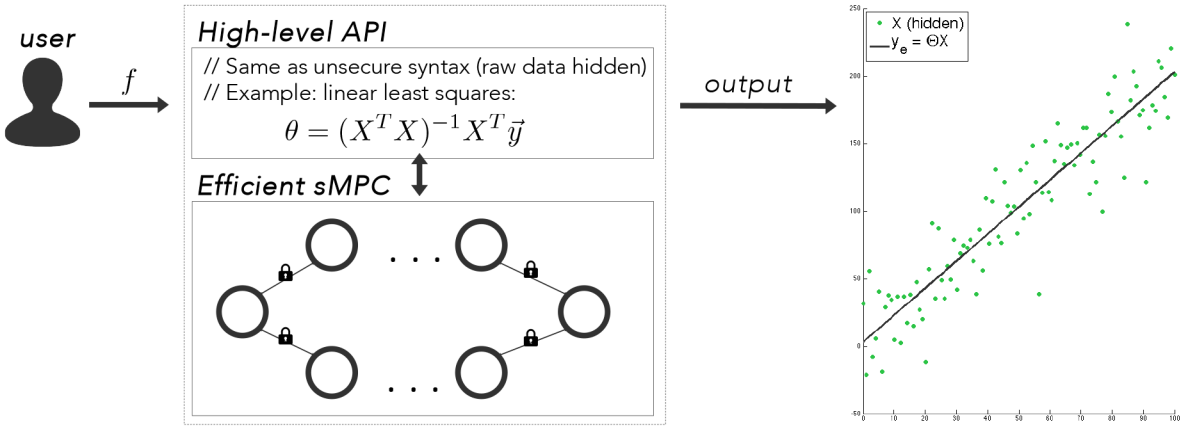
Guy Zyskind, Samuel Madden, Alex 'Sandy' Pentland

Motivation

Securely computing the result of any function without revealing the raw data is considered one of the holy grails of privacy research. And yet, while the subject has been exhaustively researched in the past 30 years and *secure multi-party computation* (MPC) was proven to be theoretically plausible, there are still no practical solutions that support running generic computations without significantly compromising utility, performance or ease of use. Specifically, there are currently no suitable solutions for doing data analysis and machine learning at scale.

Solution

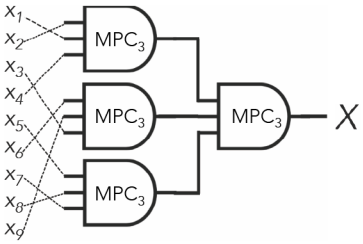
Our goal is to provide an **easy-to-use** framework for **secure data analysis** that can live up to the challenges of **Big Data**, regardless of the number of parties, the dataset, or the problem.



The Framework

Simplicity: The higher level python framework closely resembles known data-science toolkits. *virtual primitives* enable working with raw data without accessing it directly, while maintaining standard arithmetic operators.

Efficiency: The underlying secure-MPC framework simulates n-party computation using a log-depth hierarchy of 3 (or 4) party MPC gates based on secret sharing.



Security: cryptographic security is guaranteed as long as the majority is honest (using MPC_3 gates in the *passive* case). Corollary, the system is secure against at most *n/3* actively corrupted parties (using MPC_4 gates). **More importantly,** this makes the security model easy to understand and use – since the only configurable parameter is *n*.

Deployment

As a first step, we plan to deploy our framework as part of the ***MIT Living Lab*** project and demonstrate using real data that scientific research can be done efficiently in a privacy-preserving manner. This would pave the way for making the framework an open standard for privacy-preserving computation and data analysis.

MITbigdata LIVINGLAB

mit media lab