NSF CICI #2115134

University of Colorado Colorado Springs

A Software System for FHE- and MPC-based Privacy-Preserving Computation: Design, Analysis, Implementation, and Prototype

PI: Shouhuai Xu (University of Colorado Colorado Springs; sxu@uccs.edu; https://xu-lab.org/)
Co-PI: Charles Benight (University of Colorado Colorado Springs)
Co-PI: Yanyan Zhuang (University of Colorado Colorado Springs)

Problem Statement:

- How can we enable privacy-preserving computation over sensitive data (i.e., ***sharing data utility*** without ***sharing data*** per se) in real-world environments (e.g., scientific research settings)?
- Example application: Sensitive medical data (e.g., EHRs, genomic data, and clinical research records) are highly valuable for medical analytics research but cannot be shared with researchers because of privacy concerns and regulations. Privacy-preserving computation provides a practical solution to this problem.

Challenges:

- Sensitive (e.g., medical) data often do not have standard or widely accepted data schema or structure (e.g., different attributes are used to represent the same thing).
- Practical privacy-preserving computation involves trade-offs among speed, precision, etc.
- There is no one-size-fits-all solution.

Overview of Results and Impacts:

- Design and implement FHE-based and MPC-based privacy-preserving computation systems.
- Analyze trade-offs among efficiency, data precision, and computation expressiveness.
- Can be adopted by any scientific research community to solving their data-sharing problem via privacy-preserving computation approaches.

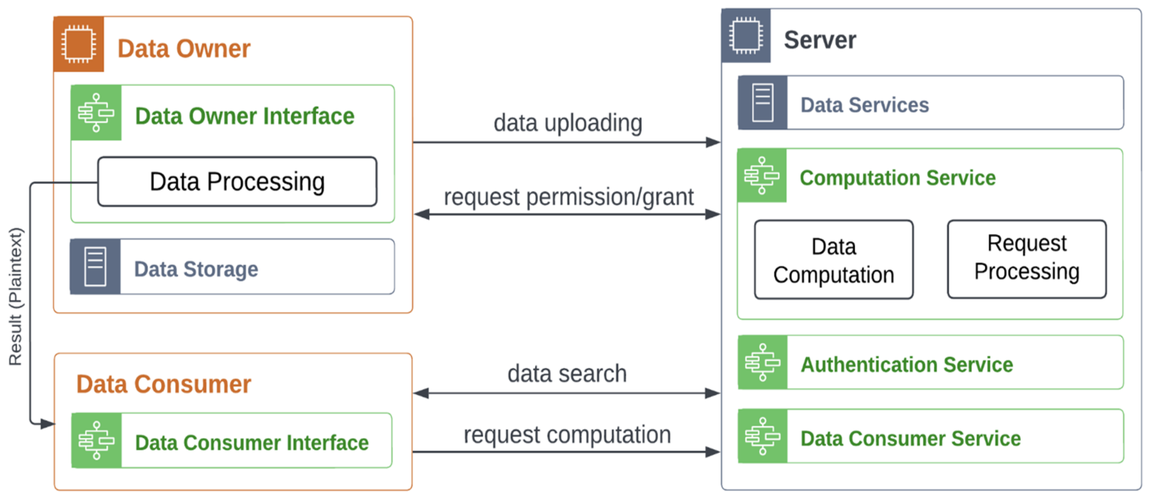

Figure 1. Architecture of Fully Homomorphic Encryption (FHE)-based solution

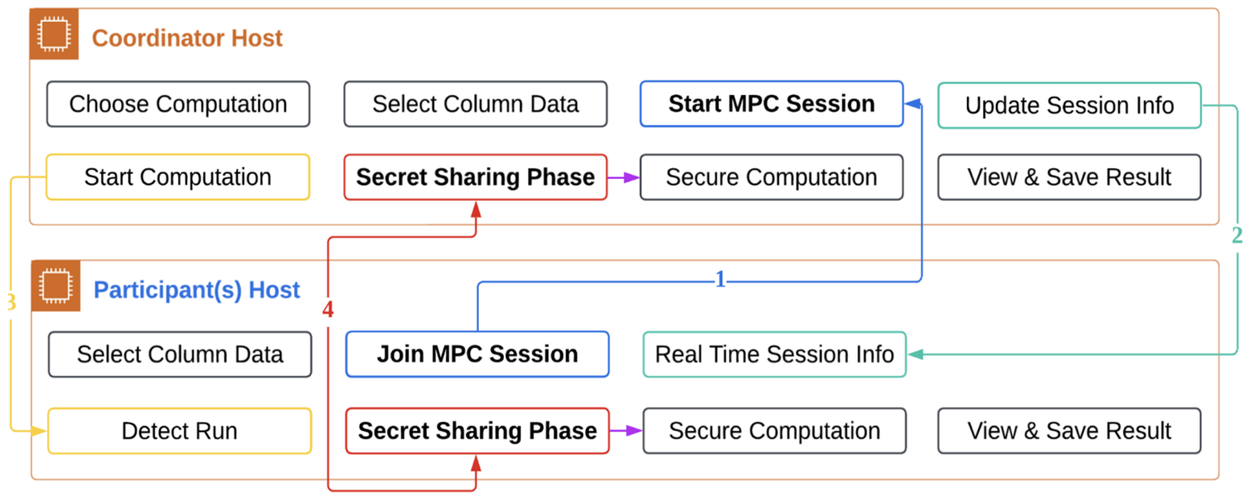

Figure 2. Architecture of Multi-Party Computation (MPC)-based solution

Use Cases of FHE-based Solution:

- One data owner wants to help data consumers (e.g., researchers) in leveraging their data for meaningful purposes, without sharing plaintext data but releasing homomorphically encrypted data to a semi-trust third party, which conducts computation on ciphertext.
- Data owner decrypts computation results for data consumers.

Use Cases of MPC-based Solution:

- Data owners (i.e., participants) want to conduct computation on their joint data (e.g., the union of their respective datasets), without exposing their private data to each other.
- No need of any semi-trusted third party because the data owners conduct the computation themselves.
- Participants need to be online at the same time

Aspect	FHE	MPC
Primary Setting	Outsourced computation over ciphertext	Decentralized joint computation
Data Location	Upload ciphertext to semi-trusted server	Each participant (i.e., no server)
Who Computes	Server computes on ciphertexts	Participants (i.e., data owners)
Use cases	Single owner, server-assisted analytics	Multi-owner collaborative analytics
Key Strength	Simple client-side participation for both data owners and data consumers	Do not need any semi-trusted server or third party
Limitation	Less natural for multiple independent data owners, but a technical twist works	High coordination and communication overhead

Table 1. Comparison between FHE-based and MPC-based solutions.

FHE-based vs. MPC-based Solutions:

- FHE-based and MPC-based solutions have their suitable scenarios, incurring differen.t computational & communication complexities.
- FHE-based solution is typically used when data owners are not data consumers.
- MPC-based solution is typically used when data owners are also data consumers.

Prototype Implementation and Source Code Availability:

- Our solutions are primarily implemented in Rust, using openFHE and TFHE libraries for FHE-based solution and using MP-SPDZ for MPC-based solution. A paper document the systems and their performance will be available soon.
- Source code will be made publicly available at https://github.com/LCD-UCCS/privacy-preserving-compute.