CMP_SC 8001 - Introduction to Secure Multiparty Computation

Introduction

Wei Jiang

Department of Electrical Engineering and Computer Science

University of Missouri

Outline

Outline

Types of Secure and Verifiable Computation

- There are two main types of secure and verifiable computation:
  1. outsourced computation
  2. multi-party computation
- We focus on multi-party computation
- First we briefly describe outsourced computation to distinguish it from multi-party computation

Outsourced Computation

- One party owns the data and wants to be able to obtain the result of computation on that data
- Another party receives and stores the **encrypted** data:
  - performs computation on the encrypted data, and
  - provides the encrypted results to the data owner

  without learning anything about the **input data**, **intermediate values**, or **final result**
- The data owner can then decrypt the returned results to obtain the output

Common Approaches - Homomorphic Encryption

- Homomorphic encryption allows operations on encrypted data
- Partially-homomorphic encryption (PHE) schemes allow certain operations (e.g., addition or multiplication) be performed
- Examples of efficient PHE schemes
  - Paillier, 1999
  - Naccache and Stern, 1998
  - Boneh et al., 2005
- Systems built on them are limited to specialized problems that can be framed in terms of the supported operations

Common Approaches - Fully Homomorphic Encryption

- Fully homomorphic encryption (FHE) supports both addition and multiplication; thus, any function can be computed
  - FHE was first envisioned by Rivest et al. in 1978
  - The first FHE scheme was proposed by Gentry in 2009, building on lattice-based cryptography
- There has been much recent interest in implementing FHE schemes, such as
  - Gentry and Halevi (2011)
  - Halevi and Shoup (2015)
  - Chillotti et al. (2016)
- Building secure, deployable and scalable systems using FHE remains an open problem

FHE and MPC Comparison

- In their basic forms, FHE and MPC address different aspects of secure computation, but do provide similar functionalities
- There are ways to adapt FHE to use multiple keys that enables multi-party computation
  - Asharov et al., 2012
  - López- Alt et al., 2012
  - Mukherjee and Wichs, 2016
- FHE offers an asymptotic communication improvement comparing to MPC, but is computational more expensive
  - State-of-the-art FHE (Chillotti et al., 2017) are thousands of times slower than two-party and multi-party secure computation in typical applications

FHE and MPC Comparison

- The performance of FHE and MPC depends on the relative costs of computation and bandwidth
- For high-bandwidth settings, such as where devices connected within a data center, MPC vastly outperforms FHE
- As FHE techniques improve, and the relative cost of bandwidth over computation increases, FHE-based techniques may eventually become competitive with MPC

Outline

The Goal of Multi-Party Computation

- Secure multi-party computation (MPC) enables a group of independent data owners who do not trust each other or any common third party to jointly compute a function that depends on all of their private inputs
- MPC differs from outsourced computation in that all of the protocol participants are data owners who participate in executing a protocol

History of MPC

- The idea of secure computation was introduced by Andrew Yao in the early 1980s (Yao, 1982)
- The paper introduced a general notion of secure computation
  - m parties want to jointly compute a function $f(x_1, x_2, \ldots, x_m)$ where x_i is the i^{th} party's private input
- In a series of talks over the next few years (but not included in any formal publication), Yao introduced Garbled Circuits
  - the basis for many MPC implementations

History of MPC

- Secure computation was primarily of only theoretical interest for the next twenty years
- In the early 2000s, algorithmic improvements and computing costs make it more realistic to build practical systems
- Fairplay (Malkhi et al., 2004) was the first notable implementation of a general-purpose MPC
    - A privacy-preserving program could be expressed in a high level language, and
    - compiled to executables that could be run by the data-owning participants as a multi-party protocol

History of MPC

- Fairplay is scalable and limited to toy programs
- Since then, the speed of MPC protocols has improved by more than five orders of magnitude
  - due to a combination of cryptographic, protocol, network and hardware improvements
- This enabled MPC applications to scale to a wide range of interesting and important applications

Generic and Specialized MPC

- Yao's garbled circuits protocol is a generic protocol:
  - Compute any discrete function that can be represented as a fixed-size circuit
- For specific functionalities, there may be custom protocols that are much more efficient than the best generic protocols

Outline

Yao's Millionaires Problem

- It was used to introduce secure computation and not meant to be a useful application
- Yao (1982) introduces it simply:
  - "Two millionaires wish to know who is richer; however, they do not want to find out inadvertently any additional information about each other's wealth."
- More formally, the goal is to compute the Boolean result of $x_1 \leq x_2$
  - where x_1 is the first party's private input and x_2 is the second party's private input
- Although it is a toy problem, it is be useful for illustrating issues in MPC applications

Secure Auctions

- The need for privacy in auctions is well understood: both bidders and sellers need to be able to rely on the privacy and non-malleability of bids
- Bid privacy requires that no player may learn any other player's bid (other than perhaps revealing the winning bid upon the completion of the auction)
- Bid non-malleability means that a player's bid may not be manipulated to generate a related bid
    - If a party generates a bid of n, then another party should not be able to use this bid to produce a bid of $n + 1$
    - Note that bid privacy does not necessarily imply bid non-malleability

Sealed Bib Auction

- Bidders submit private (sealed) bids in attempts to purchase property, selling to the highest bidder
- The first bidder's bid value must be kept secret from other bidders to prevent those from having an unfair advantage
- Bid malleability may allow a dishonest bidder Bob to present a bid just slightly over Alice's bid
- The auction itself must be conducted correctly, awarding the item to the highest bidder for the amount of their bid

Vickrey Auction

- A type of sealed-bid auction:
  - The highest bidder wins but the price paid is the value of the second-highest bid
  - This gives bidders an incentive to bid their true value
- It also requires privacy and non-malleability of each bid, and correctness in determining the winner and price

MPC for Secure Auctions

- MPC can be used to easily achieve all these features by
  - embedding the desired properties into the function used to jointly execute the auction
- All the participants can verify the function
- Then rely on the MPC protocol to provide high confidence that the auction will be conducted confidentially and fairly

Voting

- Secure electronic voting is simply computation of the addition function which tallies the vote
- Privacy and non-malleability of the vote (properties discussed above in the context of auctions) are essential for similar technical reasons
- Additionally, because voting is a fundamental civil process, these properties are often asserted by legislation

Voting

- Voting is an example of an application which may require properties not covered by the standard MPC security definitions
- In particular, the property of **coercion resistance** is not standard in MPC (but can be formally expressed and achieved (Küsters et al., 2012))
- The issue here is the ability of voters to prove to a third party how they voted
- If such a proof is possible, then voter coercion is also possible

Secure Machine Learning

- MPC can be used to enable privacy in both the **inference** and **training** phases of machine learning systems
- Oblivious model inference allows a client C to submit a request to a server S holding a pre-trained model
    - keeping the request private from S and the model private from C
- In this setting, the inputs to the MPC are the private model from S, and the private test input from C, and the output is the model's prediction only known to C
- MiniONN (Liu et al., 2017) allows any standard neural network to be converted to an oblivious model service using a combination of MPC and homomorphic encryption techniques

Secure Machine Learning

- In the training phase, MPC can be used to enable a group of parties to train a model based on their combined data without exposing that data
- For large scale data sets, it is not feasible to perform training across private data sets as a generic many-party computation
- To improve training efficiency and scalability
  - hybrid approaches that combine MPC with homomorphic encryption (Nikolaenko et al., 2013b; Gascón et al., 2017)
  - custom protocols to perform secure arithmetic operations efficiently (Mohassel and Zhang, 2017)

Other Applications

Many other interesting applications have been proposed for using MPC to enable privacy, such as

- Network security monitoring (Burkhart et al., 2010) and genomics (Wang et al., 2015a; Jagadeesh et al., 2017)
- Stable matching (Doerner et al., 2016), contact discovery (Li et al., 2013; De Cristofaro et al., 2013), ad conversion (Kreuter, 2017), and spam filtering on encrypted email (Gupta et al., 2017)

Deployment Challenges

- We are still in the early stages of deploying MPC solutions to real problems
- Challenging problems beyond MPC execution itself
  - Building confidence in the system executing the protocol
  - Understanding what sensitive information might be inferred from the revealed output of MPC
  - Enabling decision makers without technical cryptography background to understand the benefits and risks of MPC

Deployment Challenges

- Despite these challenges, there have been several successful deployments of MPC
- Companies now focus on providing MPC-based solutions
- In this early stage, organizations are typically not seeking to use MPC as an added layer of privacy
- MPC is mainly deployed to enable a feature or an entire application which would not be possible without trusting specialized hardware
  - due to the value of the shared data, protective privacy legislation, or mistrust of the participants

Danish Sugar Beets Auction

- It is considered to be the first commercial application of MPC
- Danish researchers collaborated with the Danish government and stakeholders to create an auction and bidding platform for sugar beet production contracts
- As reported in Bogetoft et al. (2009), bid privacy and auction security were seen as essential for auction participants
  - The farmers felt that their bids reflected their capabilities and costs, which they did not want to reveal to Danisco
  - Also, Danisco needed to be involved in the auction as the contracts were securities directly affecting the company

Danish Sugar Beets Auction

- The auction was implemented as a three-party MPC among representatives for Danisco, the farmer's association (DKS) and the researchers (SIMAP project)
- Bogetoft et al. (2009) explained a three party solution was selected because
  - it was natural in the given scenario, and
  - allowed using efficient information theoretic tools such as secret sharing
- This led to the formation of Partisia, a company supporting secure auctions and related applications for industries such as spectrum and energy markets (Gallagher et al., 2017)

Estonian Students Study

- Estonia was alarmed about graduation rates of IT students
  - In 2012, nearly 43% of IT students enrolled in the previous five years had failed to graduate
- One potential explanation considered was that
  - the IT industry was hiring too aggressively, luring students away from completing their studies

Estonian Students Study

- The Estonian Association of Information and Communication Technology wanted to investigate by mining education and tax records to see if there was a correlation
- However, privacy legislation prevented data sharing across the Ministry of Education and the Tax Board
  - k-anonymity-based sharing was allowed, but it would have resulted in low-quality analysis
  - since many students would not have had sufficiently large groups of peers with similar qualities

Estonian Students Study

- MPC provided a solution, facilitated by Cybernetica using their Sharemind framework (Bogdanov et al., 2008a)
- The data analysis was done as a three-party computation, with servers representing the Estonian Information System's Authority, the Ministry of Finance, and Cybernetica
- The study, reported in Cybernetica (2015) and Bogdanov (2015), found that
  - there was no correlation between working during studies and failure to graduate on time
  - but that more education was correlated with higher income

Boston Wage Equity Study

- An initiative of the City of Boston and the Boston Women's Workforce Council (BWWC)
  - to identify salary inequities across various employee gender and ethnic demographics at different levels of employment
  - widely supported by the Boston area organizations, but privacy concerns prevented direct sharing of salary data
- In response, Boston University researchers designed and implemented a web-based MPC aggregation tool
  - which allowed employers to submit the salary data privately with full technical and legal protection

Boston Wage Equity Study

- As reported by Bestavros et al. (2017), MPC enabled the BWWC to conduct their analysis and produce a report presenting their findings
- The effort included meetings with stakeholders to convey
  - the risks and benefits of participating in the MPC
  - the importance of addressing usability and trust concerns
- One indirect result of this work is inclusion of secure multi-party computation as a requirement in a bill for student data analysis introduced in the United States Senate (Wyden, 2017)

Key Management

- One of the biggest problems faced by organizations today is safeguarding sensitive data as it is being used
- This is best illustrated using the example of authentication keys
- This use case lies at the core of the product offering of Unbound Tech (Unbound Tech, 2018)
- Unlike other uses of MPC where the goal is to protect data owned by multiple parties from exposure, here the goal is to protect from compromise the data owned by a single entity

Key Management

- To enable a secure login facility, an organization must maintain private keys
- Suppose shared-key authentication, where each user has shared a randomly chosen secret key with the organization
- Each time the user U authenticates, the organization's server S looks up the database of keys and retrieves U's public key sk_U
- The key is then used to authenticate and admit U to the network by running key exchange

Key Management

- The security community has long accepted that
  - it is nearly impossible to operate a fully secure complex system, and
  - an adversary will be able to penetrate and stealthily take control over some of the network nodes
- The advanced adversary, sometimes called Advanced Persistent Threat (APT), aims to quietly undermine the organization
- Naturally, the most prized target for APT and other types of attackers is the key server

Hardening the Key using MPC

- Splitting the key server's functionality into two (or more) hosts, S_1 and S_2, and secret-sharing key material between the two
- Now, an attacker must compromise both S_1 and S_2 to gain access to the keys
  - run S_1 and S_2 on two different software stacks to minimize the chance that they will be both vulnerable to malware, and
  - operate them using two different sub-organizations to minimize insider threats

Hardening the Key using MPC

- Routine execution does need access to the keys to provide authentication service
- At the same time, key should never be reconstructed as the reconstructing party will be the target of the APT attack
- Instead, the three players, S_1, S_2, and the authenticating user U, will run the authentication inside MPC
  - without ever reconstructing any secrets, and thus
  - removing the vulnerability and hardening the defense

Acknowledgment

The contents of these slides are based on the following book:

- A Pragmatic Introduction to Secure Multi-Party Computation
  `https://securecomputation.org/`
- Chapter 1: Introduction