Chap3#3: Privacy Preservation in Machine Learning #3

February 20, 2023

भारतीय प्रौद्योगिकी
संस्थान जम्मू
INDIAN INSTITUTE OF
TECHNOLOGY JAMMU

Devesh C Jinwala,
Professor, SVNIT and Adjunct Prof., CSE, IIT Jammu
Department of Computer Science and Engineering,
Sardar Vallabhhai National Institute of Technology, SURAT

Chap 2: ML Applications in Security: Topics to study

- Privacy Preservation, What is Privacy? Data Privacy. Machine Learning in Privacy Preservation: Four Main stakes to Privacy preservation in ML. Two principle approaches: (a) Augmenting the ML techniques with the conventional approaches in the domain of privacy preservation to achieve privacy viz. Homomorphic Encryption, Secret Multiparty Computations, Zero Knowledge Proofs, Perturbation techniques (e.g. differential privacy) Anonymization techniques (e.g.)k-Anonymity, l-Diversity) (b) ML-specific approaches like Federated Learning OR Ensemble Learning. Homomorphic Encryption Algorithms and the associated mathematics. Ethical issues and Law for data / process privacy : GDPR, Alexa, other relevant applications [6 hours]

Reviewing the theme of ML Paradigms for Privacy Preservation

Four Main stakes to Privacy preservation in ML

There are four main stakes to privacy preservation in machine learning in general:

- Privacy of the input data
  - the assurance that other parties, including the model developer, will not be able to see a user's input data
- Privacy of the output data
  - the assurance that the output of a model is only accessible to the client whose data is being inferred upon.
- Privacy of the model
  - rhe assurance that a hostile party will not be able to steal the model
- Data privacy in training
  - the assurance that a malicious party will not reverse-engineer the training data - although gathering information about training data and model is more difficult than that for the data.

Privacy Preserving Machine Learning: How to achieve?

The goal of privacy-preserving machine learning is

- to bridge the gap between privacy while receiving the benefits of machine learning.

Privacy Preserving Machine Learning: How to achieve?

The goal of privacy-preserving machine learning is

- to bridge the gap between privacy while receiving the benefits of machine learning.
- is a critical facilitator for the protection of acquired data and adhering to data privacy laws.

Privacy Preserving Machine Learning: How to achieve?

The goal of privacy-preserving machine learning is

- to bridge the gap between privacy while receiving the benefits of machine learning.
- is a critical facilitator for the protection of acquired data and adhering to data privacy laws.

Privacy Preserving Machine Learning: How to achieve?

The goal of privacy-preserving machine learning is

- to bridge the gap between privacy while receiving the benefits of machine learning.
- is a critical facilitator for the protection of acquired data and adhering to data privacy laws.

Privacy-preservation in ML

- is achieved by augmenting conventional ML with different strategies that protect data privacy, that include....

Privacy Preserving Machine Learning: How to achieve?

The goal of privacy-preserving machine learning is

- to bridge the gap between privacy while receiving the benefits of machine learning.
- is a critical facilitator for the protection of acquired data and adhering to data privacy laws.

Privacy-preservation in ML

- is achieved by augmenting conventional ML with different strategies that protect data privacy, that include....
  - cryptographic approaches like

Privacy Preserving Machine Learning: How to achieve?

The goal of privacy-preserving machine learning is

- to bridge the gap between privacy while receiving the benefits of machine learning.
- is a critical facilitator for the protection of acquired data and adhering to data privacy laws.

Privacy-preservation in ML

- is achieved by augmenting conventional ML with different strategies that protect data privacy, that include....
  - cryptographic approaches like
    - homomorphic encryption

Privacy Preserving Machine Learning: How to achieve?

The goal of privacy-preserving machine learning is

- to bridge the gap between privacy while receiving the benefits of machine learning.
- is a critical facilitator for the protection of acquired data and adhering to data privacy laws.

Privacy-preservation in ML

- is achieved by augmenting conventional ML with different strategies that protect data privacy, that include....
  - cryptographic approaches like
    - homomorphic encryption
    - secure multi-party computing,

Privacy Preserving Machine Learning: How to achieve?

The goal of privacy-preserving machine learning is

- to bridge the gap between privacy while receiving the benefits of machine learning.
- is a critical facilitator for the protection of acquired data and adhering to data privacy laws.

Privacy-preservation in ML

- is achieved by augmenting conventional ML with different strategies that protect data privacy, that include....
  - cryptographic approaches like
    - homomorphic encryption
    - secure multi-party computing,
    - Zero knowledge proofs

Privacy Preserving Machine Learning: How to achieve?

The goal of privacy-preserving machine learning is

- to bridge the gap between privacy while receiving the benefits of machine learning.
- is a critical facilitator for the protection of acquired data and adhering to data privacy laws.

Privacy-preservation in ML

- is achieved by augmenting conventional ML with different strategies that protect data privacy, that include....
  - cryptographic approaches like
    - homomorphic encryption
    - secure multi-party computing,
    - Zero knowledge proofs
  - perturbation techniques like differential privacy

Privacy Preserving Machine Learning: How to achieve?

The goal of privacy-preserving machine learning is

- to bridge the gap between privacy while receiving the benefits of machine learning.
- is a critical facilitator for the protection of acquired data and adhering to data privacy laws.

Privacy-preservation in ML

- is achieved by augmenting conventional ML with different strategies that protect data privacy, that include....
  - cryptographic approaches like
    - homomorphic encryption
    - secure multi-party computing,
    - Zero knowledge proofs
  - perturbation techniques like differential privacy
  - anonymization techniques like k-Anonymity and l-Diversity

Privacy Preserving Machine Learning: How to achieve?

The goal of privacy-preserving machine learning is

- to bridge the gap between privacy while receiving the benefits of machine learning.
- is a critical facilitator for the protection of acquired data and adhering to data privacy laws.

Privacy-preservation in ML

- is achieved by augmenting conventional ML with different strategies that protect data privacy, that include....
  - cryptographic approaches like
    - homomorphic encryption
    - secure multi-party computing,
    - Zero knowledge proofs
  - perturbation techniques like differential privacy
  - anonymization techniques like k-Anonymity and l-Diversity
  - ML-specific approaches like Federated Learning OR Ensemble Learning - the Privacy-Preserving Techniques - modifying the conventional ML training methods to keep user data private.

Augmenting ML for Privacy Preservation: Secure Multiparty Computations

Secure Multiparty Computations: Motivation

Motivation for the SMC

- Need ways of controlling leakage of confidential data while these data are being used for a purpose - i.e. are stored, communicated, or computed on

Secure Multiparty Computations: Motivation

Motivation for the SMC

- Need ways of controlling leakage of confidential data while these data are being used for a purpose - i.e. are stored, communicated, or computed on
  - this should be so even in cases where the owner of the data does not trust the parties he or she communicates with.

Secure Multiparty Computations: Motivation

Motivation for the SMC

- Need ways of controlling leakage of confidential data while these data are being used for a purpose - i.e. are stored, communicated, or computed on
  - this should be so even in cases where the owner of the data does not trust the parties he or she communicates with.
- Need ways wherein a large amount of added value can be obtained by combining confidential information from several sources and computing some result that holds an interest for all parties.

Secure Multiparty Computations: Motivation

Motivation for the SMC

- Need ways of controlling leakage of confidential data while these data are being used for a purpose - i.e. are stored, communicated, or computed on
  - this should be so even in cases where the owner of the data does not trust the parties he or she communicates with.
- Need ways wherein a large amount of added value can be obtained by combining confidential information from several sources and computing some result that holds an interest for all parties.
- We look at four such scenarios/usecases viz.

Secure Multiparty Computations: Motivation

Motivation for the SMC

- Need ways of controlling leakage of confidential data while these data are being used for a purpose - i.e. are stored, communicated, or computed on
  - this should be so even in cases where the owner of the data does not trust the parties he or she communicates with.
- Need ways wherein a large amount of added value can be obtained by combining confidential information from several sources and computing some result that holds an interest for all parties.
- We look at four such scenarios/usecases viz.
  - Auctions

Secure Multiparty Computations: Motivation

Motivation for the SMC

- Need ways of controlling leakage of confidential data while these data are being used for a purpose - i.e. are stored, communicated, or computed on
  - this should be so even in cases where the owner of the data does not trust the parties he or she communicates with.
- Need ways wherein a large amount of added value can be obtained by combining confidential information from several sources and computing some result that holds an interest for all parties.
- We look at four such scenarios/usecases viz.
  - Auctions
  - Procurement systems

Secure Multiparty Computations: Motivation

Motivation for the SMC

- Need ways of controlling leakage of confidential data while these data are being used for a purpose - i.e. are stored, communicated, or computed on
    - this should be so even in cases where the owner of the data does not trust the parties he or she communicates with.
- Need ways wherein a large amount of added value can be obtained by combining confidential information from several sources and computing some result that holds an interest for all parties.
- We look at four such scenarios/usecases viz.
    - Auctions
    - Procurement systems
    - Benchmarking systems

Secure Multiparty Computations: Motivation

Motivation for the SMC

- Need ways of controlling leakage of confidential data while these data are being used for a purpose - i.e. are stored, communicated, or computed on
  - this should be so even in cases where the owner of the data does not trust the parties he or she communicates with.
- Need ways wherein a large amount of added value can be obtained by combining confidential information from several sources and computing some result that holds an interest for all parties.
- We look at four such scenarios/usecases viz.
  - Auctions
  - Procurement systems
  - Benchmarking systems
  - Tax authorities or the health care data mining systems

Secure Multiparty Computations: Formal definition

- In multiparty computation (MPC), we consider a number of players $P_1, \ldots, P_n$, who initially each hold inputs $x_1, \ldots, x_n$,

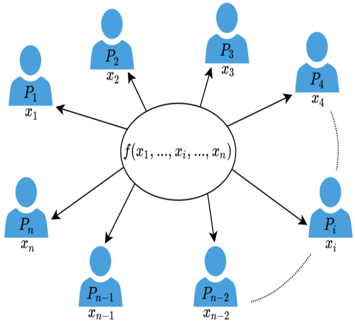

Figure 3.1 – Secure multiparty computation

Figure: Secure multiparty computation

Secure Multiparty Computations: Formal definition

- In multiparty computation (MPC), we consider a number of players $P_1, \ldots, P_n$, who initially each hold inputs $x_1, \ldots, x_n$,
- We then want to securely compute some function f on these inputs, where $f(x_1, \ldots, x_n) = (y_1, \ldots, y_n)$, such that P_i learns y_i but no other information.

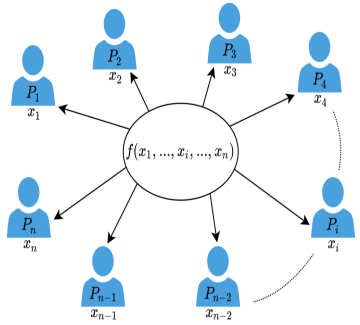

Figure 3.1 – Secure multiparty computation

Figure: Secure multiparty computation

Secure Multiparty Computations: Formal definition

- In multiparty computation (MPC), we consider a number of players $P_1, \ldots, P_n$, who initially each hold inputs $x_1, \ldots, x_n$,
- We then want to securely compute some function f on these inputs, where $f(x_1, \ldots, x_n) = (y_1, \ldots, y_n)$, such that P_i learns y_i but no other information.
    - should hold, even if players exhibit some amount of adversarial behavior.

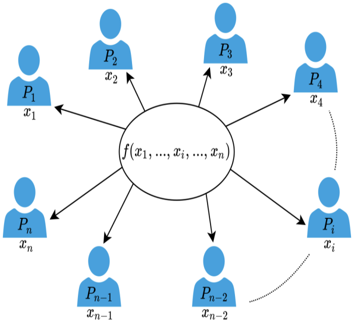

Figure 3.1 – Secure multiparty computation

Figure: Secure multiparty computation

Secure Multiparty Computations: Formal definition

- In multiparty computation (MPC), we consider a number of players $P_1, \ldots, P_n$, who initially each hold inputs $x_1, \ldots, x_n$,
- We then want to securely compute some function f on these inputs, where $f(x_1, \ldots, x_n) = (y_1, \ldots, y_n)$, such that P_i learns y_i but no other information.
    - should hold, even if players exhibit some amount of adversarial behavior.
- How to accomplish the goal ?

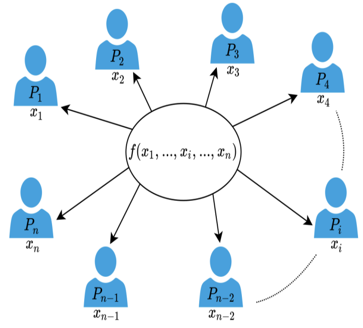

Figure 3.1 – Secure multiparty computation

Figure: Secure multiparty computation

Secure Multiparty Computations: Formal definition...

- How to accomplish the goal ?

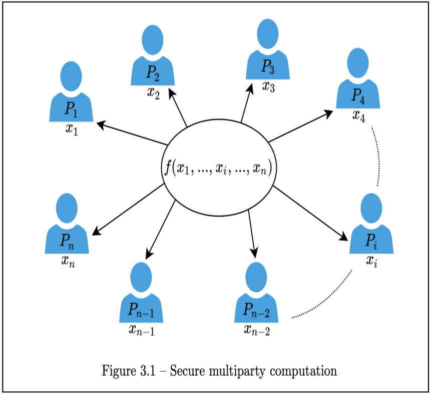

Figure: Secure multiparty computation

Secure Multiparty Computations: Formal definition...

- How to accomplish the goal ?
  - by using an interactive protocol π that the players execute.

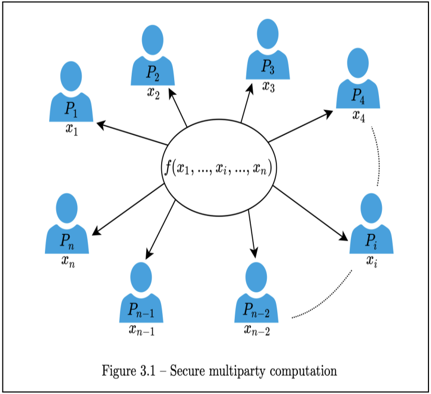

Figure: Secure multiparty computation

Secure Multiparty Computations: Formal definition...

- How to accomplish the goal ?
  - by using an interactive protocol π that the players execute.
- Intuitively, executing π must be equivalent to having a trusted party T that

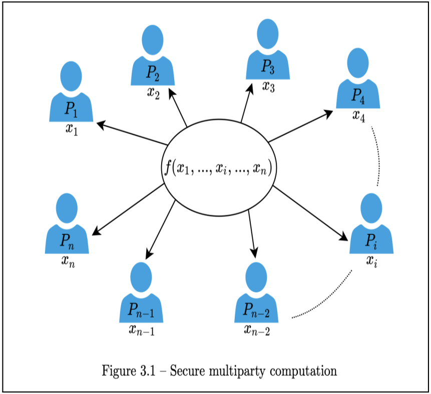

Figure: Secure multiparty computation

Secure Multiparty Computations: Formal definition...

- How to accomplish the goal ?
  - by using an interactive protocol π that the players execute.
- Intuitively, executing π must be equivalent to having a trusted party T that
  - receives privately x_i from P_i,

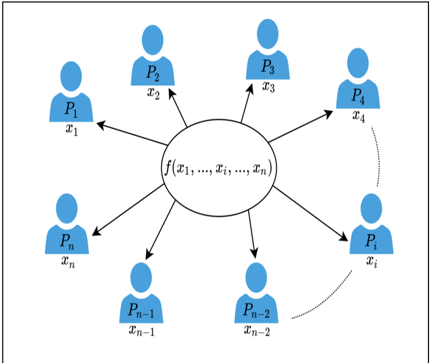

Figure 3.1 – Secure multiparty computation

Figure: Secure multiparty computation

Secure Multiparty Computations: Formal definition...

- How to accomplish the goal ?
  - by using an interactive protocol π that the players execute.
- Intuitively, executing π must be equivalent to having a trusted party T that
  - receives privately x_i from P_i,
  - computes the function, and

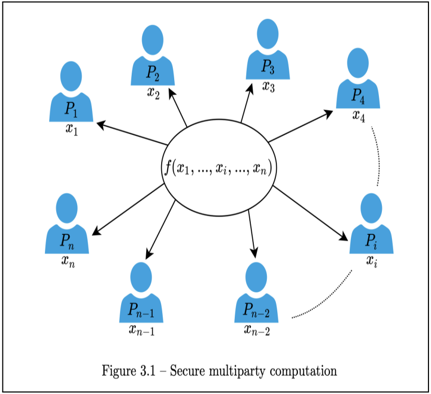

Figure: Secure multiparty computation

Secure Multiparty Computations: Formal definition...

- How to accomplish the goal ?
  - by using an interactive protocol π that the players execute.
- Intuitively, executing π must be equivalent to having a trusted party T that
  - receives privately x_i from P_i,
  - computes the function, and
  - returns y_i to each P_i

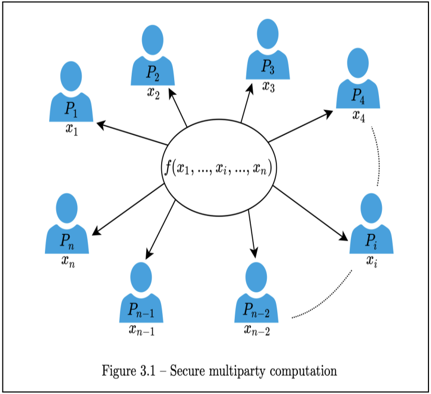

Figure: Secure multiparty computation

Secure Multiparty Computations: Formal definition...

- How to accomplish the goal ?
  - by using an interactive protocol π that the players execute.
- Intuitively, executing π must be equivalent to having a trusted party T that
  - receives privately x_i from P_i,
  - computes the function, and
  - returns y_i to each P_i
- With such a protocol we can - in principle - solve virtually any cryptographic protocol problem.

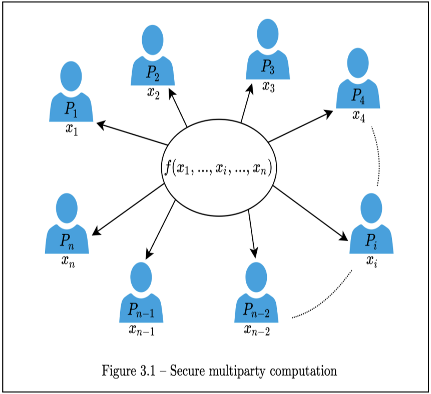

Figure: Secure multiparty computation

Secure Multiparty Computations: Auction

- Auctions exist in many variants and are used for all kinds of purposes

Secure Multiparty Computations: Auction

- Auctions exist in many variants and are used for all kinds of purposes
- a simple variant: where some item is for sale and the highest bid wins.

Secure Multiparty Computations: Auction

- Auctions exist in many variants and are used for all kinds of purposes
- a simple variant: where some item is for sale and the highest bid wins.
- conduction of the auction is in the usual way.... the price starts at some preset amount and people place increasing bids until no one wants to bid more than the currently highest bid.

Secure Multiparty Computations: Auction

- Auctions exist in many variants and are used for all kinds of purposes
- a simple variant: where some item is for sale and the highest bid wins.
- conduction of the auction is in the usual way.... the price starts at some preset amount and people place increasing bids until no one wants to bid more than the currently highest bid.
- Requirements

Secure Multiparty Computations: Auction

- Auctions exist in many variants and are used for all kinds of purposes
- a simple variant: where some item is for sale and the highest bid wins.
- conduction of the auction is in the usual way.... the price starts at some preset amount and people place increasing bids until no one wants to bid more than the currently highest bid.
- Requirements
  - every bidder, of wants to pay as small a price as possible for the item.

Secure Multiparty Computations: Auction

- Auctions exist in many variants and are used for all kinds of purposes
- a simple variant: where some item is for sale and the highest bid wins.
- conduction of the auction is in the usual way.... the price starts at some preset amount and people place increasing bids until no one wants to bid more than the currently highest bid.
- Requirements
  - every bidder, of wants to pay as small a price as possible for the item.
  - the winner of the auction always hopes to pay less than his or her maximum amount (limit he/she decided).

Secure Multiparty Computations: Auction

- Auctions exist in many variants and are used for all kinds of purposes
- a simple variant: where some item is for sale and the highest bid wins.
- conduction of the auction is in the usual way.... the price starts at some preset amount and people place increasing bids until no one wants to bid more than the currently highest bid.
- Requirements
  - every bidder, of wants to pay as small a price as possible for the item.
  - the winner of the auction always hopes to pay less than his or her maximum amount (limit he/she decided).
  - hence, the maximum amount one is willing to pay should be kept private.

Secure Multiparty Computations: Auction

- Auctions exist in many variants and are used for all kinds of purposes
- a simple variant: where some item is for sale and the highest bid wins.
- conduction of the auction is in the usual way.... the price starts at some preset amount and people place increasing bids until no one wants to bid more than the currently highest bid.
- Requirements
  - every bidder, of wants to pay as small a price as possible for the item.
  - the winner of the auction always hopes to pay less than his or her maximum amount (limit he/she decided).
  - hence, the maximum amount one is willing to pay should be kept private.
  - the result of the auction could, in principle, be computed if one were given as input the true maximum value each bidder assigns to the item on sale.

Secure Multiparty Computations: Auction

- Auctions exist in many variants and are used for all kinds of purposes
- a simple variant: where some item is for sale and the highest bid wins.
- conduction of the auction is in the usual way.... the price starts at some preset amount and people place increasing bids until no one wants to bid more than the currently highest bid.
- Requirements
  - every bidder, of wants to pay as small a price as possible for the item.
  - the winner of the auction always hopes to pay less than his or her maximum amount (limit he/she decided).
  - hence, the maximum amount one is willing to pay should be kept private.
  - the result of the auction could, in principle, be computed if one were given as input the true maximum value each bidder assigns to the item on sale.
- A good real world case study: that of sugar beets produce, the company Danisco in Denmark and deciding market clearing price (MCP), using a Secure auction.

Ref: Secure Multiparty Computation Goes Live? By Peter Bogetoft et al at https://eprint.iacr.org/2008/068.pdf

Secure Multiparty Computations: Secure Auction for Sugar Beet Farmers in Denmark

In Denmark,

- several thousand farmers produce sugar beets, which are sold to the company Danisco,

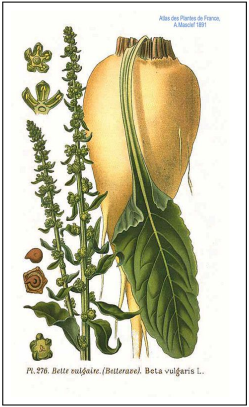
Atlas des Plantes de France, A.Masclef 1891
Pl. 276. Bette vulgaire. (Betterave). Beta vulgaris L.

Figure: Sugar Beet [Src:wiki]

Secure Multiparty Computations: Secure Auction for Sugar Beet Farmers in Denmark

In Denmark,

- several thousand farmers produce sugar beets, which are sold to the company Danisco,
- Danisco is the only sugar beets processor on the Danish market.

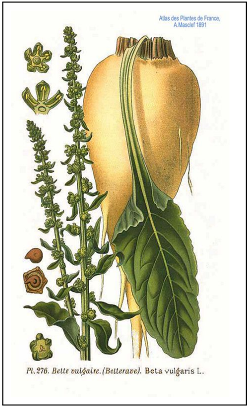
Atlas des Plantes de France, A.Masclef 1891
Pl. 276. Bette vulgaire.(Betterave). Beta vulgaris L.

Figure: Sugar Beet [Src:wiki]

Secure Multiparty Computations: Secure Auction for Sugar Beet Farmers in Denmark

In Denmark,

- several thousand farmers produce sugar beets, which are sold to the company Danisco,
- Danisco is the only sugar beets processor on the Danish market.
- Farmers used to have contracts that gave them rights and obligation to deliver a certain amount of beets to Danisco,

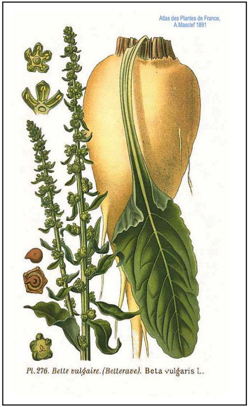
Atlas des Plantes de France,
A.Masclef 1891
Pl. 276. Bette vulgaire. (Betterave). Beta vulgaris L.

Figure: Sugar Beet [Src:wiki]

Secure Multiparty Computations: Secure Auction for Sugar Beet Farmers in Denmark

In Denmark,

- several thousand farmers produce sugar beets, which are sold to the company Danisco,
- Danisco is the only sugar beets processor on the Danish market.
- Farmers used to have contracts that gave them rights and obligation to deliver a certain amount of beets to Danisco,
- Danisco used to pay them according to a pricing scheme that is an integrated part of the contracts.

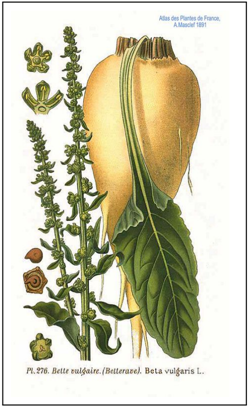
Atlas des Plantes de France,
A.Masclef 1891
Pl. 276. Bette vulgaire. (Betterave). Beta vulgaris L.

Figure: Sugar Beet [Src:wiki]

Secure Multiparty Computations: Secure Auction for Sugar Beet Farmers in Denmark

In Denmark,

- several thousand farmers produce sugar beets, which are sold to the company Danisco,
- Danisco is the only sugar beets processor on the Danish market.
- Farmers used to have contracts that gave them rights and obligation to deliver a certain amount of beets to Danisco,
- Danisco used to pay them according to a pricing scheme that is an integrated part of the contracts.
- But, subsequently it was decided to use a nation-wide exchange, and a double auction - to reallocate contracts to farmers where productions pays off best.

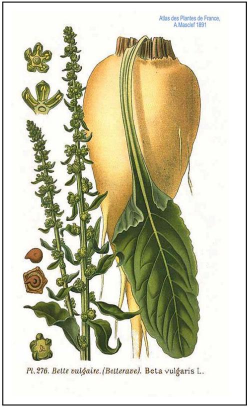
Atlas des Plantes de France, A.Masclef 1891
Pl. 276. Bette vulgaire. (Betterave). Beta vulgaris L.

Figure: Sugar Beet [Src:wiki]

Secure Multiparty Computations: Secure Auction for Sugar Beet Farmers in Denmark

In Denmark,

- several thousand farmers produce sugar beets, which are sold to the company Danisco,
- Danisco is the only sugar beets processor on the Danish market.
- Farmers used to have contracts that gave them rights and obligation to deliver a certain amount of beets to Danisco,
- Danisco used to pay them according to a pricing scheme that is an integrated part of the contracts.
- But, subsequently it was decided to use a nation-wide exchange, and a double auction - to reallocate contracts to farmers where productions pays off best.
- the double auction was to find the so called market clearing price (MCP), - a price per unit of the beets traded.

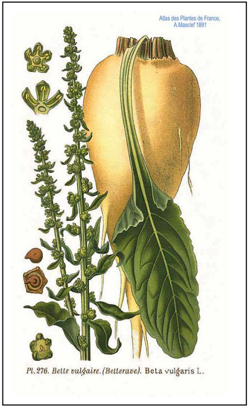
Atlas des Plantes de France,
A.Masclef 1891
Pl. 276. Bette vulgaire. (Betterave). Beta vulgaris L.

Figure: Sugar Beet [Src:wiki]

Secure Multiparty Computations: Secure Auction for ...

- In this system,

Secure Multiparty Computations: Secure Auction for ...

- In this system,
  - each buyer specifies, for each potential price, how much he is willing to buy at that price,

Secure Multiparty Computations: Secure Auction for ...

- In this system,
  - each buyer specifies, for each potential price, how much he is willing to buy at that price,
  - similarly sellers say how much they are willing to sell at each price.

Secure Multiparty Computations: Secure Auction for ...

- In this system,
  - each buyer specifies, for each potential price, how much he is willing to buy at that price,
  - similarly sellers say how much they are willing to sell at each price.
  - all bids go to an auctioneer, who computes, for each price, the total supply and demand in the market.

Secure Multiparty Computations: Secure Auction for ...

- In this system,
  - each buyer specifies, for each potential price, how much he is willing to buy at that price,
  - similarly sellers say how much they are willing to sell at each price.
  - all bids go to an auctioneer, who computes, for each price, the total supply and demand in the market.
  - the MCP is decided based on the actual demand and supply - a price where demand matches the supply

Secure Multiparty Computations: Secure Auction for ...

- In this system,
  - each buyer specifies, for each potential price, how much he is willing to buy at that price,
  - similarly sellers say how much they are willing to sell at each price.
  - all bids go to an auctioneer, who computes, for each price, the total supply and demand in the market.
  - the MCP is decided based on the actual demand and supply - a price where demand matches the supply
- this should be done with privacy - farmers would be reluctant to reveal their economic position and productivity and would be reluctant to accept Danisco acting as auctioneer, given its position in the market.

Secure Multiparty Computations: Secure Auction for ...

- In this system,
  - each buyer specifies, for each potential price, how much he is willing to buy at that price,
  - similarly sellers say how much they are willing to sell at each price.
  - all bids go to an auctioneer, who computes, for each price, the total supply and demand in the market.
  - the MCP is decided based on the actual demand and supply - a price where demand matches the supply
- this should be done with privacy - farmers would be reluctant to reveal their economic position and productivity and would be reluctant to accept Danisco acting as auctioneer, given its position in the market.
- The solution decided on was to implement an electronic double auction, where the role of the auctioneer would be played by a three-party multiparty computation done by

Secure Multiparty Computations: Secure Auction for ...

- In this system,
  - each buyer specifies, for each potential price, how much he is willing to buy at that price,
  - similarly sellers say how much they are willing to sell at each price.
  - all bids go to an auctioneer, who computes, for each price, the total supply and demand in the market.
  - the MCP is decided based on the actual demand and supply - a price where demand matches the supply
- this should be done with privacy - farmers would be reluctant to reveal their economic position and productivity and would be reluctant to accept Danisco acting as auctioneer, given its position in the market.
- The solution decided on was to implement an electronic double auction, where the role of the auctioneer would be played by a three-party multiparty computation done by
  - representatives for Danisco,

Secure Multiparty Computations: Secure Auction for ...

- In this system,
  - each buyer specifies, for each potential price, how much he is willing to buy at that price,
  - similarly sellers say how much they are willing to sell at each price.
  - all bids go to an auctioneer, who computes, for each price, the total supply and demand in the market.
  - the MCP is decided based on the actual demand and supply - a price where demand matches the supply
- this should be done with privacy - farmers would be reluctant to reveal their economic position and productivity and would be reluctant to accept Danisco acting as auctioneer, given its position in the market.
- The solution decided on was to implement an electronic double auction, where the role of the auctioneer would be played by a three-party multiparty computation done by
  - representatives for Danisco,
  - DKS (the sugar beet growers association) and

Secure Multiparty Computations: Secure Auction for ...

- In this system,
  - each buyer specifies, for each potential price, how much he is willing to buy at that price,
  - similarly sellers say how much they are willing to sell at each price.
  - all bids go to an auctioneer, who computes, for each price, the total supply and demand in the market.
  - the MCP is decided based on the actual demand and supply - a price where demand matches the supply
- this should be done with privacy - farmers would be reluctant to reveal their economic position and productivity and would be reluctant to accept Danisco acting as auctioneer, given its position in the market.
- The solution decided on was to implement an electronic double auction, where the role of the auctioneer would be played by a three-party multiparty computation done by
  - representatives for Danisco,
  - DKS (the sugar beet growers association) and
  - the SIMAP (Secure Information Management and Processing) project.

Motivation for using SMC: Auction for Sugar Beet Farmers in Denmark

Motivation for using an electronic auction system...besides need for a nation-wide exchange

- to have a cheap electronic solution for the secure auction - farmers do care about keeping their bids private.

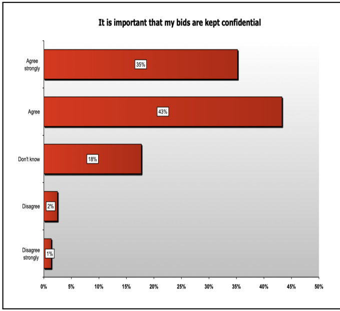

Figure: Concerned about Privacy

Motivation for using SMC: Auction for Sugar Beet Farmers in Denmark

Motivation for using an electronic auction system...besides need for a nation-wide exchange

- to have a cheap electronic solution for the secure auction - farmers do care about keeping their bids private.
- if Danisco and DKS would have tried to run the auction using conventional methods, several issues would crop up.....

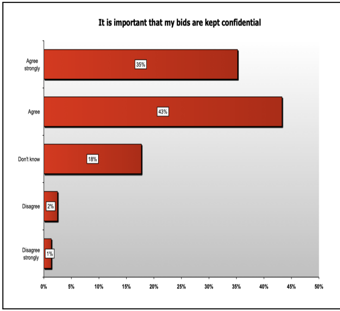

Figure: Concerned about Privacy

Motivation for using SMC: Auction for Sugar Beet Farmers in Denmark

Motivation for using an electronic auction system...besides need for a nation-wide exchange

- to have a cheap electronic solution for the secure auction - farmers do care about keeping their bids private.
- if Danisco and DKS would have tried to run the auction using conventional methods, several issues would crop up.....
  - one/more persons would need access to the bids, or

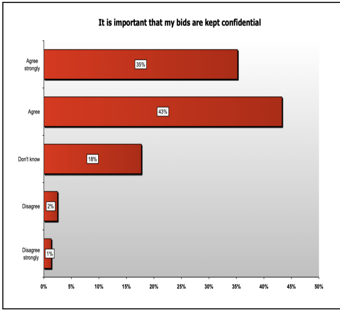

Figure: Concerned about Privacy

Motivation for using SMC: Auction for Sugar Beet Farmers in Denmark

Motivation for using an electronic auction system...besides need for a nation-wide exchange

- to have a cheap electronic solution for the secure auction - farmers do care about keeping their bids private.
- if Danisco and DKS would have tried to run the auction using conventional methods, several issues would crop up.....
  - one/more persons would need access to the bids, or
  - one or more persons would need to have control over the system holding the bids.

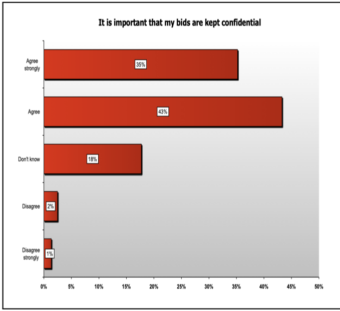

Figure: Concerned about Privacy

Motivation for using SMC: Auction for Sugar Beet Farmers in Denmark

Motivation for using an electronic auction system...besides need for a nation-wide exchange

- to have a cheap electronic solution for the secure auction - farmers do care about keeping their bids private.
- if Danisco and DKS would have tried to run the auction using conventional methods, several issues would crop up.....
  - one/more persons would need access to the bids, or
  - one or more persons would need to have control over the system holding the bids.
  - who should have access to the data and when?

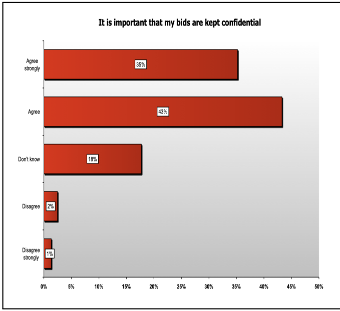

Figure: Concerned about Privacy

Motivation for using SMC: Auction for Sugar Beet Farmers in Denmark

Motivation for using an electronic auction system...besides need for a nation-wide exchange

- to have a cheap electronic solution for the secure auction - farmers do care about keeping their bids private.
- if Danisco and DKS would have tried to run the auction using conventional methods, several issues would crop up.....
  - one/more persons would need access to the bids, or
  - one or more persons would need to have control over the system holding the bids.
  - who should have access to the data and when?
  - who has responsibility if data leaks, and what are the consequences?

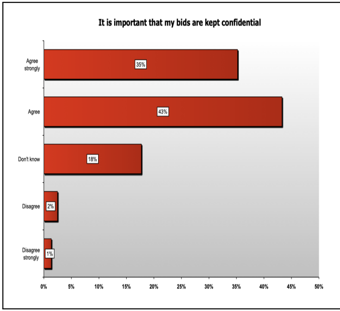

Figure: Concerned about Privacy

Motivation for using SMC: Auction for Sugar Beet Farmers in Denmark

Motivation for using an electronic auction system...besides need for a nation-wide exchange

- to have a cheap electronic solution for the secure auction - farmers do care about keeping their bids private.
- if Danisco and DKS would have tried to run the auction using conventional methods, several issues would crop up.....
  - one/more persons would need access to the bids, or
  - one or more persons would need to have control over the system holding the bids.
  - who should have access to the data and when?
  - who has responsibility if data leaks, and what are the consequences?
  - if the auctioneer knows maximum and is working with another bidder, can force the price to be always just below the maximum - force one to pay more than if the auction had been honest.

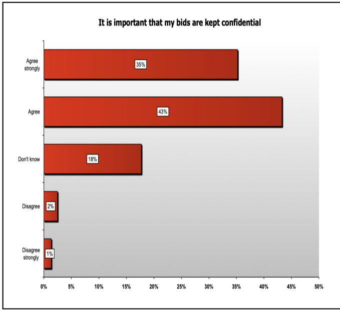

Figure: Concerned about Privacy

Motivation for using SMC: Auction for Sugar Beet Farmers in Denmark

Motivation for using an electronic auction system…besides need for a nation-wide exchange

- to have a cheap electronic solution for the secure auction - farmers do care about keeping their bids private.
- if Danisco and DKS would have tried to run the auction using conventional methods, several issues would crop up.....
  - one/more persons would need access to the bids, or
  - one or more persons would need to have control over the system holding the bids.
  - who should have access to the data and when?
  - who has responsibility if data leaks, and what are the consequences?
  - if the auctioneer knows maximum and is working with another bidder, can force the price to be always just below the maximum - force one to pay more than if the auction had been honest.

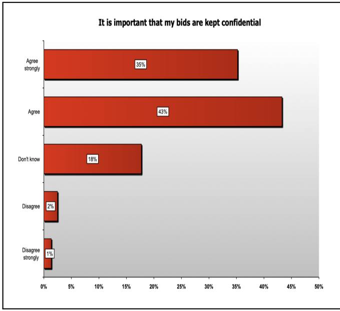

Figure: Concerned about Privacy

Motivation for using SMC: Auction for Sugar Beet Farmers in Denmark

Motivation for using an electronic auction system...besides need for a nation-wide exchange

- to have a cheap electronic solution for the secure auction - farmers do care about keeping their bids private.
- if Danisco and DKS would have tried to run the auction using conventional methods, several issues would crop up.....
  - one/more persons would need access to the bids, or
  - one or more persons would need to have control over the system holding the bids.
  - who should have access to the data and when?
  - who has responsibility if data leaks, and what are the consequences?
  - if the auctioneer knows maximum and is working with another bidder, can force the price to be always just below the maximum - force one to pay more than if the auction had been honest.

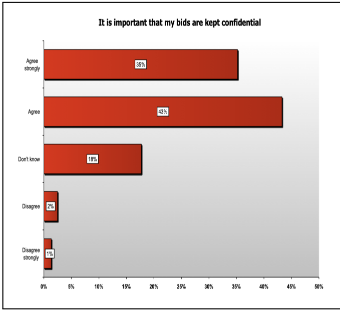

Figure: Concerned about Privacy

Src: Peter Bogetoft et al at https://eprint.iacr.org/2008/068.pdf

Secure Multiparty Computations: Procurement systems

A procurement system

- is a sort of inverted auction, where some party (typically a public institution) asks companies to make an offer on the price for doing a certain job.

Secure Multiparty Computations: Procurement systems

A procurement system

- is a sort of inverted auction, where some party (typically a public institution) asks companies to make an offer on the price for doing a certain job.
    - In such a case, the lowest bid usually wins.

Secure Multiparty Computations: Procurement systems

A procurement system

- is a sort of inverted auction, where some party (typically a public institution) asks companies to make an offer on the price for doing a certain job.
  - In such a case, the lowest bid usually wins.
- However, bidders are typically interested in getting as high a price as possible.

Secure Multiparty Computations: Procurement systems

A procurement system

- is a sort of inverted auction, where some party (typically a public institution) asks companies to make an offer on the price for doing a certain job.
    - In such a case, the lowest bid usually wins.
- However, bidders are typically interested in getting as high a price as possible.
- therefore, the bids are private information: a participating company is clearly not interested in competitors learning its bid before it has to make its own bid.

Secure Multiparty Computations: Procurement systems

A procurement system

- is a sort of inverted auction, where some party (typically a public institution) asks companies to make an offer on the price for doing a certain job.
  - In such a case, the lowest bid usually wins.
- However, bidders are typically interested in getting as high a price as possible.
- therefore, the bids are private information: a participating company is clearly not interested in competitors learning its bid before it has to make its own bid.
  - to prevent a competitor beat the company's bid by always offering a price that is slightly lower than that offered by the company.

Secure Multiparty Computations: Procurement systems

A procurement system

- is a sort of inverted auction, where some party (typically a public institution) asks companies to make an offer on the price for doing a certain job.
    - In such a case, the lowest bid usually wins.
- However, bidders are typically interested in getting as high a price as possible.
- therefore, the bids are private information: a participating company is clearly not interested in competitors learning its bid before it has to make its own bid.
    - to prevent a competitor beat the company's bid by always offering a price that is slightly lower than that offered by the company.
- The result of the process, namely, who wins the contract, can, in principle, be computed given all the true values of the bids.

Secure Multiparty Computations: Benchmarking systems

Benchmark analysis

- applicable when one is interested in knowing how well you are doing compared with other companies in the same line of business as yours.

Secure Multiparty Computations: Benchmarking systems

Benchmark analysis

- applicable when one is interested in knowing how well you are doing compared with other companies in the same line of business as yours.
- the comparison may be concerned with a number of different parameters - typically profit relative to size, average salaries, and productivity.

Secure Multiparty Computations: Benchmarking systems

Benchmark analysis

- applicable when one is interested in knowing how well you are doing compared with other companies in the same line of business as yours.
- the comparison may be concerned with a number of different parameters - typically profit relative to size, average salaries, and productivity.
- other companies will most likely have similar interests in such a comparison.

Secure Multiparty Computations: Benchmarking systems

Benchmark analysis

- applicable when one is interested in knowing how well you are doing compared with other companies in the same line of business as yours.
- the comparison may be concerned with a number of different parameters - typically profit relative to size, average salaries, and productivity.
- other companies will most likely have similar interests in such a comparison.
- takes input from all participating companies, based on which it tries to compute information on how well a company in the given line of business should be able to perform

Secure Multiparty Computations: Benchmarking systems

Benchmark analysis

- applicable when one is interested in knowing how well you are doing compared with other companies in the same line of business as yours.
- the comparison may be concerned with a number of different parameters - typically profit relative to size, average salaries, and productivity.
- other companies will most likely have similar interests in such a comparison.
- takes input from all participating companies, based on which it tries to compute information on how well a company in the given line of business should be able to perform
- each company is subsequently told how its performance compares with this "ideal."

Secure Multiparty Computations: Benchmarking systems

Benchmark analysis

- applicable when one is interested in knowing how well you are doing compared with other companies in the same line of business as yours.
- the comparison may be concerned with a number of different parameters - typically profit relative to size, average salaries, and productivity.
- other companies will most likely have similar interests in such a comparison.
- takes input from all participating companies, based on which it tries to compute information on how well a company in the given line of business should be able to perform
- each company is subsequently told how its performance compares with this “ideal.”
- But, would anyone be willing to disclose openly the values of such parameters ?

Secure Multiparty Computations: Benchmarking systems

Benchmark analysis

- applicable when one is interested in knowing how well you are doing compared with other companies in the same line of business as yours.
- the comparison may be concerned with a number of different parameters - typically profit relative to size, average salaries, and productivity.
- other companies will most likely have similar interests in such a comparison.
- takes input from all participating companies, based on which it tries to compute information on how well a company in the given line of business should be able to perform
- each company is subsequently told how its performance compares with this "ideal."
- But, would anyone be willing to disclose openly the values of such parameters ?

Secure Multiparty Computations: Benchmarking systems

Benchmark analysis

- applicable when one is interested in knowing how well you are doing compared with other companies in the same line of business as yours.
- the comparison may be concerned with a number of different parameters - typically profit relative to size, average salaries, and productivity.
- other companies will most likely have similar interests in such a comparison.
- takes input from all participating companies, based on which it tries to compute information on how well a company in the given line of business should be able to perform
- each company is subsequently told how its performance compares with this "ideal."
- But, would anyone be willing to disclose openly the values of such parameters ?

Hence, the desired results must be computed from the private data

Secure Multiparty Computations: Tax authorities/Healthcare data mining systems

- In most countries, public institutions such as the tax authorities or the health care system keep databases containing information on citizens

Secure Multiparty Computations: Tax authorities/Healthcare data mining systems

- In most countries, public institutions such as the tax authorities or the health care system keep databases containing information on citizens
- there are advantages one can get from coordinated access to several such databases.

Secure Multiparty Computations: Tax authorities/Healthcare data mining systems

- In most countries, public institutions such as the tax authorities or the health care system keep databases containing information on citizens
- there are advantages one can get from coordinated access to several such databases.
- However, there is clearly a privacy concern here - access to many different databases by a single party opens the possibility that complete dossiers could be compiled on particular citizens,

Secure Multiparty Computations: Tax authorities/Healthcare data mining systems

- In most countries, public institutions such as the tax authorities or the health care system keep databases containing information on citizens
- there are advantages one can get from coordinated access to several such databases.
- However, there is clearly a privacy concern here - access to many different databases by a single party opens the possibility that complete dossiers could be compiled on particular citizens,
    - which would be a violation of privacy.

Secure Multiparty Computations: Tax authorities/Healthcare data mining systems

- In most countries, public institutions such as the tax authorities or the health care system keep databases containing information on citizens
- there are advantages one can get from coordinated access to several such databases.
- However, there is clearly a privacy concern here - access to many different databases by a single party opens the possibility that complete dossiers could be compiled on particular citizens,
  - which would be a violation of privacy.
- In fact, accessing data on the same person in several distinct databases is forbidden by law in some countries.

Tax authorities/Healthcare data mining systems

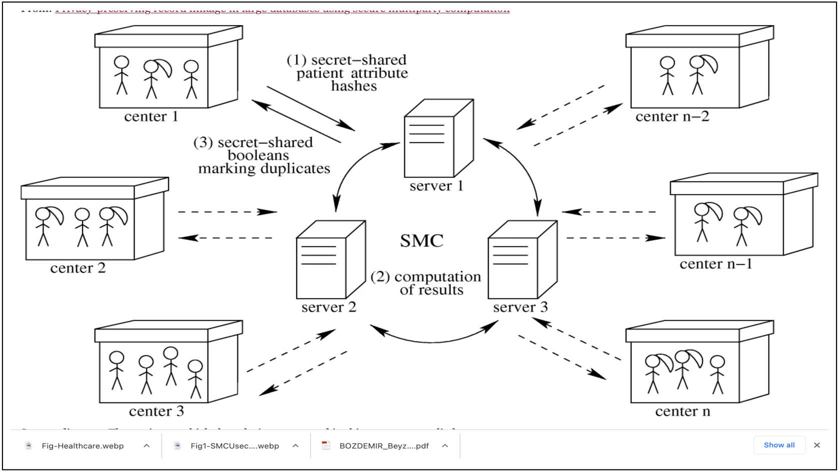
From: Privacy-preserving record linkage in large databases using secure multiparty computation
center 1
(1) secret–shared patient attribute hashes
center n–2
(3) secret–shared booleans marking duplicates
server 1
center 2
SMC
center n–1
server 2
(2) computation of results
server 3
center 3
center n
Fig-Healthcare.webp
Fig1-SMCUsec....webp
BOZDEMIR_Beyz....pdf
Show all
×

Figure: Privacy Preserving Data Mining

Src: Laud, P., Pankova, A. Privacy-preserving record linkage in large databases using secure multiparty computation. BMC Med Genomics 11 (Suppl 4), 84 (2018).

Some common features of all the scenarios looked at so far

In each application,

- there are a number of parties,

Some common features of all the scenarios looked at so far

In each application,

- there are a number of parties,
- each possesses some private data.

Some common features of all the scenarios looked at so far

In each application,

- there are a number of parties,
- each possesses some private data.
- one wants to do some computation that needs all the private data as input.

Some common features of all the scenarios looked at so far

In each application,

- there are a number of parties,
- each possesses some private data.
- one wants to do some computation that needs all the private data as input.
- all the parties are interested in learning the result, or at least some part of it, but want to keep their private data as confidential as possible.

Some common features of all the scenarios looked at so far

In each application,

- there are a number of parties,
- each possesses some private data.
- one wants to do some computation that needs all the private data as input.
- all the parties are interested in learning the result, or at least some part of it, but want to keep their private data as confidential as possible.
- How to solve this problem ?

Some common features of all the scenarios looked at so far

In each application,

- there are a number of parties,
- each possesses some private data.
- one wants to do some computation that needs all the private data as input.
- all the parties are interested in learning the result, or at least some part of it, but want to keep their private data as confidential as possible.
- How to solve this problem ?
- What could be one of the most trivial solution to the problem?

Some common features of all the scenarios looked at so far

In each application,

- there are a number of parties,
- each possesses some private data.
- one wants to do some computation that needs all the private data as input.
- all the parties are interested in learning the result, or at least some part of it, but want to keep their private data as confidential as possible.
- How to solve this problem ?
- What could be one of the most trivial solution to the problem?
- Thus, we are left with a fundamental question: can the problem be solved without relying on a trusted party?

Some common features of all the scenarios looked at so far

In each application,

- there are a number of parties,
- each possesses some private data.
- one wants to do some computation that needs all the private data as input.
- all the parties are interested in learning the result, or at least some part of it, but want to keep their private data as confidential as possible.
- How to solve this problem ?
- What could be one of the most trivial solution to the problem?
- Thus, we are left with a fundamental question: can the problem be solved without relying on a trusted party?

Some common features of all the scenarios looked at so far

In each application,

- there are a number of parties,
- each possesses some private data.
- one wants to do some computation that needs all the private data as input.
- all the parties are interested in learning the result, or at least some part of it, but want to keep their private data as confidential as possible.
- How to solve this problem ?
- What could be one of the most trivial solution to the problem?
- Thus, we are left with a fundamental question: can the problem be solved without relying on a trusted party?

How is it possible that while we want to compute a result that depends on private data from all involved parties and the data from several parties remain unknown to everyone, and hence that we do not have to trust any party?

Overview of SMC Protocol for Sugar Beet farmer's application

A typical protocol for Secure auctions for Sugar Beet Farmers in Denmark

The inputs for the protocol are as follows

- The input is a bid. The bid is an ordered list of non-negative integers $x_{ij} \mid j = 1, \ldots, P$, where index j refers to one of the P possible prices per unit.
- A bid can be a sell bid if the list is non-decreasing, or a buy bid in which case it is non-increasing.
- It must be possible to deliver these inputs non-interactively (and securely) to the servers.
- For a buy bid, x_{ij}'s denote the quantity the bidder wants to buy at the i'th price per unit, for a sell bid similarly for sell bids, y_{ij}'s denote the quantity the bidder wants to sell at the i'th price per unit

A typical protocol for Secure auctions for Sugar Beet Farmers in Denmark

The inputs for the protocol are as follows

- The input is a bid. The bid is an ordered list of non-negative integers $x_{ij} \mid j = 1, \ldots, P$, where index j refers to one of the P possible prices per unit.
- A bid can be a sell bid if the list is non-decreasing, or a buy bid in which case it is non-increasing.
- It must be possible to deliver these inputs non-interactively (and securely) to the servers.
- For a buy bid, x_{ij}'s denote the quantity the bidder wants to buy at the i'th price per unit, for a sell bid similarly for sell bids, y_{ij}'s denote the quantity the bidder wants to sell at the i'th price per unit

Then, the protocol followed is as shown further...

...continued

Requirements for a general protocol for SMC

- Assume that the parties, or players, that participate are called $P_1, \ldots, P_n$.

Requirements for a general protocol for SMC

- Assume that the parties, or players, that participate are called $P_1,\ldots,P_n$.
- Each player P_i holds a secret input x_i, and the players agree on some function f that takes n inputs.

Requirements for a general protocol for SMC

- Assume that the parties, or players, that participate are called $P_1,\ldots,P_n$.
- Each player P_i holds a secret input x_i, and the players agree on some function f that takes n inputs.
- Goal: Compute $y = f(x_1,\ldots,x_n)$ subject to the two conditions:

Requirements for a general protocol for SMC

- Assume that the parties, or players, that participate are called $P_1, \ldots, P_n$.
- Each player P_i holds a secret input x_i, and the players agree on some function f that takes n inputs.
- Goal: Compute $y = f(x_1, \ldots, x_n)$ subject to the two conditions:
  - Correctness: the correct value of y is computed; and

Requirements for a general protocol for SMC

- Assume that the parties, or players, that participate are called $P_1, \ldots, P_n$.
- Each player P_i holds a secret input x_i, and the players agree on some function f that takes n inputs.
- Goal: Compute $y = f(x_1, \ldots, x_n)$ subject to the two conditions:
  - Correctness: the correct value of y is computed; and
  - Privacy: y is the only new information that is released.

Requirements for a general protocol for SMC

- Assume that the parties, or players, that participate are called $P_1, \ldots, P_n$.
- Each player P_i holds a secret input x_i, and the players agree on some function f that takes n inputs.
- Goal: Compute $y = f(x_1, \ldots, x_n)$ subject to the two conditions:
  - Correctness: the correct value of y is computed; and
  - Privacy: y is the only new information that is released.
- To learn y, it is essential that nothing else but y is leaked.

Requirements for a general protocol for SMC

- Assume that the parties, or players, that participate are called $P_1, \ldots, P_n$.
- Each player P_i holds a secret input x_i, and the players agree on some function f that takes n inputs.
- Goal: Compute $y = f(x_1, \ldots, x_n)$ subject to the two conditions:
  - Correctness: the correct value of y is computed; and
  - Privacy: y is the only new information that is released.
- To learn y, it is essential that nothing else but y is leaked.
- This, Computing f securely $\implies$ Computing f such that privacy and correctness are achieved

Requirements for a general protocol for SMC

- Assume that the parties, or players, that participate are called $P_1,\ldots,P_n$.
- Each player P_i holds a secret input x_i, and the players agree on some function f that takes n inputs.
- Goal: Compute $y = f(x_1,\ldots,x_n)$ subject to the two conditions:
  - Correctness: the correct value of y is computed; and
  - Privacy: y is the only new information that is released.
- To learn y, it is essential that nothing else but y is leaked.
- This, Computing f securely $\implies$ Computing f such that privacy and correctness are achieved
  - e.g. if x_i is a number, representing P_i's bid in an auction, and $f(x_1,\ldots,x_n) = (z,j)$, where $x_j = z$ and $z \geq x_i$, $i = 1,\ldots,n$;

Requirements for a general protocol for SMC

- Assume that the parties, or players, that participate are called $P_1, \ldots, P_n$.
- Each player P_i holds a secret input x_i, and the players agree on some function f that takes n inputs.
- Goal: Compute $y = f(x_1, \ldots, x_n)$ subject to the two conditions:
  - Correctness: the correct value of y is computed; and
  - Privacy: y is the only new information that is released.
- To learn y, it is essential that nothing else but y is leaked.
- This, Computing f securely $\implies$ Computing f such that privacy and correctness are achieved
  - e.g. if x_i is a number, representing P_i's bid in an auction, and $f(x_1,...,x_n) = (z, j)$, where $x_j = z$ and $z \geq x_i$, $i = 1, \ldots, n$;
  - that is, f outputs the highest bid and the identity of the corresponding bidder.

Requirements for a general protocol for SMC

- Assume that the parties, or players, that participate are called $P_1, \ldots, P_n$.
- Each player P_i holds a secret input x_i, and the players agree on some function f that takes n inputs.
- Goal: Compute $y = f(x_1, \ldots, x_n)$ subject to the two conditions:
  - Correctness: the correct value of y is computed; and
  - Privacy: y is the only new information that is released.
- To learn y, it is essential that nothing else but y is leaked.
- This, Computing f securely $\implies$ Computing f such that privacy and correctness are achieved
  - e.g. if x_i is a number, representing P_i's bid in an auction, and $f(x_1, \ldots, x_n) = (z, j)$, where $x_j = z$ and $z \geq x_i$, $i = 1, \ldots, n$;
  - that is, f outputs the highest bid and the identity of the corresponding bidder.
- Similarly the bid of the second-highest bidder can also be computed - function now is second price auction

Requirements: How one might compute f securely ? SMC requires Secret Sharing

- For this purpose, the players follow a protocol i.e. a set of instructions that players are supposed to follow to obtain the desired result
- For simplicity, we will assume for now that players always follow the protocol.
- We assume that
    - any pair of players can communicate securely i.e.
    - it is possible for P_i to send a message m to P_j such that no third party sees m, and P_j knows that m came from P_i.
- There are two items of concern here
    - What could be the function f Note that, the function f could serve one of the following purposes:
        - Secure addition e.g. in electronic voting.
        - Secure multiplication e.g. in matchmaking.
        - and so on...
    - How could the second assumption mentioned above be realized in practice?

How one might compute f securely ? Secret Sharing...

Secret sharing provides a way

- for a party, say P_1, to spread information on a secret number x across all the players such that

How one might compute f securely ? Secret Sharing...

Secret sharing provides a way

- for a party, say P_1, to spread information on a secret number x across all the players such that
  - they together hold full information on x, yet no player (except, of course, P_1) has any information on x.

How one might compute f securely ? Secret Sharing...

Secret sharing provides a way

- for a party, say P_1, to spread information on a secret number x across all the players such that
  - they together hold full information on x, yet no player (except, of course, P_1) has any information on x.
- How does secret sharing work, broadly ?

How one might compute f securely ? Secret Sharing...

Secret sharing provides a way

- for a party, say P_1, to spread information on a secret number x across all the players such that
  - they together hold full information on x, yet no player (except, of course, P_1) has any information on x.
- How does secret sharing work, broadly ?
  - First, choose a prime p, and define $\mathbb{Z}_p$ as $p = \{0, 1, ..., p–1\}$ i.e. here, we will think of the secret x as a number in p.

How one might compute f securely ? Secret Sharing...

Secret sharing provides a way

- for a party, say P_1, to spread information on a secret number x across all the players such that
  - they together hold full information on x, yet no player (except, of course, P_1) has any information on x.
- How does secret sharing work, broadly ?
  - First, choose a prime p, and define $\mathbb{Z}_p$ as $p = \{0, 1, ..., p–1\}$ i.e. here, we will think of the secret x as a number in p.
  - How to share the secret s ?

How one might compute f securely ? Secret Sharing...

Secret sharing provides a way

- for a party, say P_1, to spread information on a secret number x across all the players such that
  - they together hold full information on x, yet no player (except, of course, P_1) has any information on x.
- How does secret sharing work, broadly ?
  - First, choose a prime p, and define $\mathbb{Z}_p$ as $p = \{0, 1, ..., p–1\}$ i.e. here, we will think of the secret x as a number in p.
  - How to share the secret s ?
  - P_1 chooses numbers r_1, r_2 uniformly at random in $\mathbb{Z}_p$ and sets $r_3 = x$ - r_1 - r_2 mod p.

How one might compute f securely ? Secret Sharing...

Secret sharing provides a way

- for a party, say P_1, to spread information on a secret number x across all the players such that
  - they together hold full information on x, yet no player (except, of course, P_1) has any information on x.
- How does secret sharing work, broadly ?
  - First, choose a prime p, and define $\mathbb{Z}_p$ as $p = \{0, 1, ..., p–1\}$ i.e. here, we will think of the secret x as a number in p.
  - How to share the secret s ?
  - P_1 chooses numbers r_1, r_2 uniformly at random in $\mathbb{Z}_p$ and sets $r_3 = x$ - r_1 - r_2 mod p.
    - that is, P_1 chooses r_1,r_2,r_3 randomly from p, subject to the constraint that $x = r1 + r2 + r3$ mod p.

How one might compute f securely ? Secret Sharing...

Secret sharing provides a way

- for a party, say P_1, to spread information on a secret number x across all the players such that
    - they together hold full information on x, yet no player (except, of course, P_1) has any information on x.
- How does secret sharing work, broadly ?
    - First, choose a prime p, and define $\mathbb{Z}_p$ as $p = \{0, 1, ..., p–1\}$ i.e. here, we will think of the secret x as a number in p.
    - How to share the secret s ?
    - P_1 chooses numbers r_1, r_2 uniformly at random in $\mathbb{Z}_p$ and sets $r_3 = x$ - r_1 - r_2 mod p.
        - that is, P_1 chooses r_1,r_2,r_3 randomly from p, subject to the constraint that $x = r1 + r2 + r3$ mod p.
    - this ensures that each of the r_1,r_2,r_3 numbers is uniformly chosen in p: for each of them, all values in p are possible and equally likely.

How one might compute f securely ? Secret Sharing...

Secret sharing provides a way

- for a party, say P_1, to spread information on a secret number x across all the players such that
  - they together hold full information on x, yet no player (except, of course, P_1) has any information on x.
- How does secret sharing work, broadly ?
  - First, choose a prime p, and define $\mathbb{Z}_p$ as $p = \{0, 1, ..., p–1\}$ i.e. here, we will think of the secret x as a number in p.
  - How to share the secret s ?
  - P_1 chooses numbers r_1, r_2 uniformly at random in $\mathbb{Z}_p$ and sets $r_3 = x$ - r_1 - r_2 mod p.
    - that is, P_1 chooses r_1,r_2,r_3 randomly from p, subject to the constraint that $x = r1 + r2 + r3$ mod p.
  - this ensures that each of the r_1,r_2,r_3 numbers is uniformly chosen in p: for each of them, all values in p are possible and equally likely.
  - Now, P_1 sends privately r_1,r_3 to P_2, r_1,r_2, to P_3, and keeps r_2,r_3 himself or herself.

How one might compute f securely ? Secret Sharing...

Secret sharing provides a way

- for a party, say P_1, to spread information on a secret number x across all the players such that
  - they together hold full information on x, yet no player (except, of course, P_1) has any information on x.
- How does secret sharing work, broadly ?
  - First, choose a prime p, and define $\mathbb{Z}_p$ as $p = \{0, 1, ..., p–1\}$ i.e. here, we will think of the secret x as a number in p.
  - How to share the secret s ?
  - P_1 chooses numbers r_1, r_2 uniformly at random in $\mathbb{Z}_p$ and sets $r_3 = x$ - r_1 - r_2 mod p.
    - that is, P_1 chooses r_1,r_2,r_3 randomly from p, subject to the constraint that $x = r1 + r2 + r3$ mod p.
  - this ensures that each of the r_1,r_2,r_3 numbers is uniformly chosen in p: for each of them, all values in p are possible and equally likely.
  - Now, P_1 sends privately r_1,r_3 to P_2, r_1,r_2, to P_3, and keeps r_2,r_3 himself or herself.
  - The r_j's are called the shares of the secret x.

How one might compute f securely ? Secret Sharing...

- Two essential properties satisfied

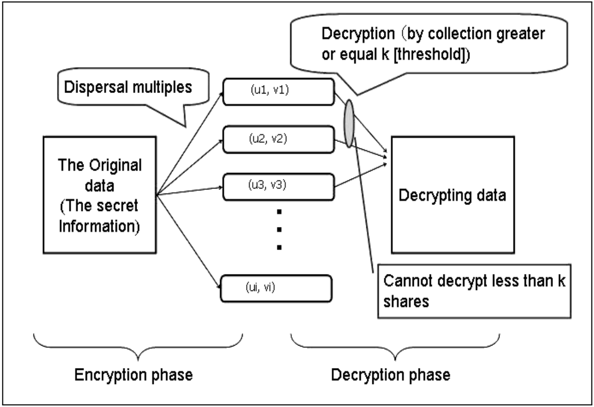
Decryption (by collection greater or equal k [threshold])
Dispersal multiples
(u1, v1)
(u2, v2)
(u3, v3)
(ui, vi)
The Original data (The secret Information)
Decrypting data
Cannot decrypt less than k shares
Encryption phase
Decryption phase

Figure: Secret Sharing conceptually

Src: Kohno, Eitaro & Ohta, Tomoyuki et al. Improvement of Dependability against Node Capture Attac
for Wireless Sensor Networks. IEICE Transactions on Information and Systems. (2011).

How one might compute f securely ? Secret Sharing...

- Two essential properties satisfied
  - the secret x is kept private in the sense that neither P_2 nor P_3 knows anything about that secret.

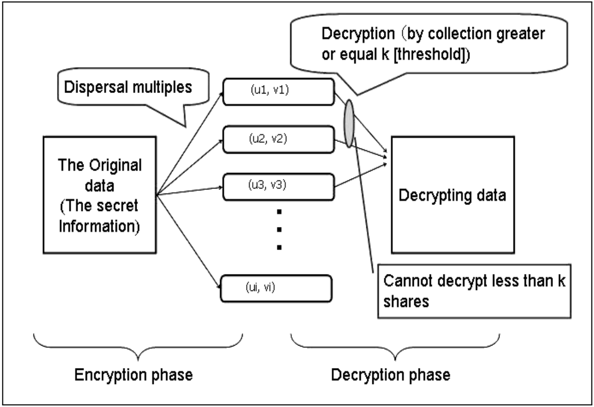
Decryption (by collection greater or equal k [threshold])
Dispersal multiples
(u1, v1)
(u2, v2)
(u3, v3)
(ui, vi)
The Original data (The secret Information)
Decrypting data
Cannot decrypt less than k shares
Encryption phase
Decryption phase

Figure: Secret Sharing conceptually

Src: Kohno, Eitaro & Ohta, Tomoyuki et al. Improvement of Dependability against Node Capture Atta
for Wireless Sensor Networks. IEICE Transactions on Information and Systems. (2011).

How one might compute f securely ? Secret Sharing...

- Two essential properties satisfied
  - the secret x is kept private in the sense that neither P_2 nor P_3 knows anything about that secret.
  - Second, x can be reconstructed if shares from at least two players are available.

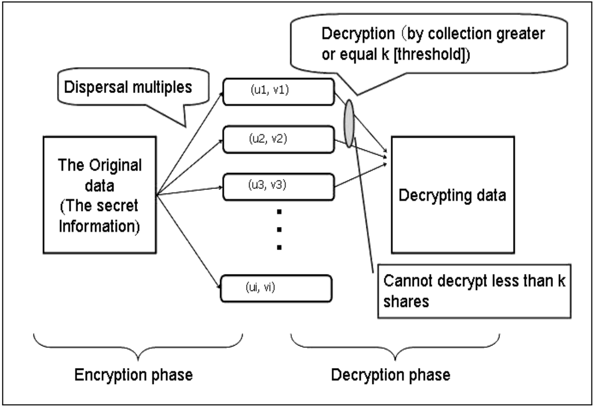
Decryption (by collection greater or equal k [threshold])
Dispersal multiples
(u1, v1)
(u2, v2)
(u3, v3)
(ui, vi)
The Original data (The secret Information)
Decrypting data
Cannot decrypt less than k shares
Encryption phase
Decryption phase

Figure: Secret Sharing conceptually

Src: Kohno, Eitaro & Ohta, Tomoyuki et al. Improvement of Dependability against Node Capture Atta
for Wireless Sensor Networks. IEICE Transactions on Information and Systems. (2011).

Broad design of a protocol for SMC with ffunction f as Secure Addition

A typical protocol for Secure Addition

Then, the protocol followed is as follows - with the assumptions as before, i.e. participants are P_1,P_2,P_3; input for P_i is $x_i \epsilon \mathbb{Z}_p$, where p is a fixed prime agreed on in advance.

1. Each P_i computes and distributes shares of his or her secret x_i. For the purpose, he or she chooses $r_{i,1}$, $r_{i,2}$ uniformly at random in p and sets $r_{i,3} = x_i - r_{i,1}$ - $r_{i,2}$ mod p.

A typical protocol for Secure Addition

Then, the protocol followed is as follows - with the assumptions as before, i.e. participants are P_1,P_2,P_3; input for P_i is $x_i \epsilon \mathbb{Z}_p$, where p is a fixed prime agreed on in advance.

1. Each P_i computes and distributes shares of his or her secret x_i. For the purpose, he or she chooses $r_{i,1}$, $r_{i,2}$ uniformly at random in p and sets $r_{i,3} = x_i - r_{i,1}$ - $r_{i,2}$ mod p.
2. Each P_i sends privately $r_{i,2}$, $r_{i,3}$ to P_1, $r_{i,3}$, $r_{i,2}$ to P_2, and $r_{i,1}$, $r_{i,2}$ to P_3.

A typical protocol for Secure Addition

Then, the protocol followed is as follows - with the assumptions as before, i.e. participants are P_1,P_2,P_3; input for P_i is $x_i \epsilon \mathbb{Z}_p$, where p is a fixed prime agreed on in advance.

1. Each P_i computes and distributes shares of his or her secret x_i. For the purpose, he or she chooses $r_{i,1}$, $r_{i,2}$ uniformly at random in p and sets $r_{i,3} = x_i - r_{i,1}$ - $r_{i,2}$ mod p.
2. Each P_i sends privately $r_{i,2}$, $r_{i,3}$ to P_1, $r_{i,3}$, $r_{i,2}$ to P_2, and $r_{i,1}$, $r_{i,2}$ to P_3.
    - Thus, P_1, for instance, now holds $r_{1,2}$,$r_{1,3}$, $r_{2,2}$,$r_{2,3}$, and $r_{3,2}$,$r_{3,3}$.

A typical protocol for Secure Addition

Then, the protocol followed is as follows - with the assumptions as before, i.e. participants are P_1,P_2,P_3; input for P_i is $x_i \epsilon \mathbb{Z}_p$, where p is a fixed prime agreed on in advance.

1. Each P_i computes and distributes shares of his or her secret x_i. For the purpose, he or she chooses $r_{i,1}$, $r_{i,2}$ uniformly at random in p and sets $r_{i,3} = x_i$ – $r_{i,1}$ - $r_{i,2}$ mod p.
2. Each P_i sends privately $r_{i,2}$, $r_{i,3}$ to P_1, $r_{i,3}$, $r_{i,2}$ to P_2, and $r_{i,1}$, $r_{i,2}$ to P_3.
   - Thus, P_1, for instance, now holds $r_{1,2}$,$r_{1,3}$, $r_{2,2}$,$r_{2,3}$, and $r_{3,2}$,$r_{3,3}$.
3. Each P_j adds corresponding shares of the three secrets – more precisely, he or she computes, for $\ell \neq j$, $s_\ell = r_{1,\ell} + r_{2,\ell} + r_{3,\ell}$ mod p and announces s_ℓ to all parties. Each party computes and announces two values.

A typical protocol for Secure Addition

Then, the protocol followed is as follows - with the assumptions as before, i.e. participants are P_1,P_2,P_3; input for P_i is $x_i \epsilon \mathbb{Z}_p$, where p is a fixed prime agreed on in advance.

1. Each P_i computes and distributes shares of his or her secret x_i. For the purpose, he or she chooses $r_{i,1}$, $r_{i,2}$ uniformly at random in p and sets $r_{i,3} = x_i - r_{i,1}$ - $r_{i,2}$ mod p.
2. Each P_i sends privately $r_{i,2}$, $r_{i,3}$ to P_1, $r_{i,3}$, $r_{i,2}$ to P_2, and $r_{i,1}$, $r_{i,2}$ to P_3.
    - Thus, P_1, for instance, now holds $r_{1,2}$,$r_{1,3}$, $r_{2,2}$,$r_{2,3}$, and $r_{3,2}$,$r_{3,3}$.
3. Each P_j adds corresponding shares of the three secrets – more precisely, he or she computes, for $\ell \neq j$, $s_\ell = r_{1,\ell} + r_{2,\ell} + r_{3,\ell}$ mod p and announces s_ℓ to all parties. Each party computes and announces two values.
4. All parties compute the result $v = s_1 + s_2 + s_3$ mod p.

Secure Addition analysis: Why the result v is indeed the correct result ?

This can be judged from the fact that

$$v = \sum_j s_j \; mod \; p = \sum_j \sum_i r_{i,j} \; mod \; p \; = \sum_i x_j \; mod \; p$$

- this expression shows that the protocol computes the sum modulo p of the inputs, no matter how the x_i are chosen.

Secure Addition analysis: Why the result v is indeed the correct result ?

This can be judged from the fact that

$$v = \sum_j s_j \ mod \ p = \sum_j \sum_i r_{i,j} \ mod \ p \ = \ \sum_i x_j \ mod \ p$$

- this expression shows that the protocol computes the sum modulo p of the inputs, no matter how the x_i are chosen.
- However, if we let the parties choose $x_i = 1$ for *yes* and $x_i = 0$ for *no* and make sure that $p > 3$, then $\sum i * x_i$ mod p $= \sum i * x_i$ because all x_i are 0 or 1,

Secure Addition analysis: Why the result v is indeed the correct result ?

This can be judged from the fact that

$$v = \sum_j s_j \ mod \ p = \sum_j \sum_i r_{i,j} \ mod \ p \ = \ \sum_i x_j \ mod \ p$$

- this expression shows that the protocol computes the sum modulo p of the inputs, no matter how the x_i are chosen.
- However, if we let the parties choose $x_i = 1$ for *yes* and $x_i = 0$ for *no* and make sure that $p > 3$, then $\sum i * x_i$ mod p $= \sum i * x_i$ because all x_i are 0 or 1,
- hence, their sum cannot be larger than p.

Secure Addition analysis: Why the result v is indeed the correct result ?

This can be judged from the fact that

$$v = \sum_j s_j \; mod \; p = \sum_j \sum_i r_{i,j} \; mod \; p \; = \; \sum_i x_j \; mod \; p$$

- this expression shows that the protocol computes the sum modulo p of the inputs, no matter how the x_i are chosen.
- However, if we let the parties choose $x_i = 1$ for *yes* and $x_i = 0$ for *no* and make sure that $p > 3$, then $\sum i * x_i$ mod p $= \sum i * x_i$ because all x_i are 0 or 1,
- hence, their sum cannot be larger than p.
- hence, in this case, v is indeed the number of yes votes.

Secure Addition analysis: Why the result v is indeed the correct result ?

This can be judged from the fact that

$$v = \sum_j s_j \; mod \; p = \sum_j \sum_i r_{i,j} \; mod \; p \; = \; \sum_i x_j \; mod \; p$$

- this expression shows that the protocol computes the sum modulo p of the inputs, no matter how the x_i are chosen.
- However, if we let the parties choose $x_i = 1$ for *yes* and $x_i = 0$ for *no* and make sure that $p > 3$, then
  $\sum i * x_i$ mod p $= \sum i * x_i$ because all x_i are 0 or 1,
- hence, their sum cannot be larger than p.
- hence, in this case, v is indeed the number of yes votes.

Secure Addition analysis: Why the result v is indeed the correct result ?

This can be judged from the fact that

$$v = \sum_j s_j \; mod \; p = \sum_j \sum_i r_{i,j} \; mod \; p \; = \; \sum_i x_j \; mod \; p$$

- this expression shows that the protocol computes the sum modulo p of the inputs, no matter how the x_i are chosen.
- However, if we let the parties choose $x_i = 1$ for *yes* and $x_i = 0$ for *no* and make sure that $p > 3$, then $\sum i * x_i$ mod p $= \sum i * x_i$ because all x_i are 0 or 1,
- hence, their sum cannot be larger than p.
- hence, in this case, v is indeed the number of yes votes.

How is it the case that no new information other than the result v is leaked to any player?

Secure Addition Analysis...:Why no new information except the result v is leaked ?

- Let us concentrate on P_1 for concreteness. In step 1 x_1, x_2, and x_3 are secrets shared, and this tells P_1 nothing whatsoever about x_2, x_3

Secure Addition Analysis...:Why no new information except the result v is leaked ?

- Let us concentrate on P_1 for concreteness. In step 1 x_1, x_2, and x_3 are secrets shared, and this tells P_1 nothing whatsoever about x_2, x_3
- In the final step, s_1, s_2, s_3 are announced. But,

Secure Addition Analysis...:Why no new information except the result v is leaked ?

- Let us concentrate on P_1 for concreteness. In step 1 x_1, x_2, and x_3 are secrets shared, and this tells P_1 nothing whatsoever about x_2, x_3
- In the final step, s_1, s_2, s_3 are announced. But,
    - P_1 already knows s_2,s_3, so s_1 is the only new piece of information.

Secure Addition Analysis...:Why no new information except the result v is leaked ?

- Let us concentrate on P_1 for concreteness. In step 1 x_1, x_2, and x_3 are secrets shared, and this tells P_1 nothing whatsoever about x_2, x_3
- In the final step, s_1, s_2, s_3 are announced. But,
  - P_1 already knows s_2,s_3, so s_1 is the only new piece of information.
  - However, seeing s_1 will tell P_1 what v is and nothing more. How ?

Secure Addition Analysis...:Why no new information except the result v is leaked ?

- Let us concentrate on P_1 for concreteness. In step 1 x_1, x_2, and x_3 are secrets shared, and this tells P_1 nothing whatsoever about x_2, x_3
- In the final step, s_1, s_2, s_3 are announced. But,
  - P_1 already knows s_2,s_3, so s_1 is the only new piece of information.
  - However, seeing s_1 will tell P_1 what v is and nothing more. How ?
  - Because, if one is given s_2,s_3, and v, one can compute $s_1 = v$ - s_2 - s_3 mod p.

Secure Addition Analysis...:Why no new information except the result v is leaked ?

- Let us concentrate on P_1 for concreteness. In step 1 x_1, x_2, and x_3 are secrets shared, and this tells P_1 nothing whatsoever about x_2, x_3
- In the final step, s_1, s_2, s_3 are announced. But,
  - P_1 already knows s_2,s_3, so s_1 is the only new piece of information.
  - However, seeing s_1 will tell P_1 what v is and nothing more. How ?
  - Because, if one is given s_2,s_3, and v, one can compute $s_1 = v - s_2 - s_3 \bmod p$.
  - That is, given what P_1 is supposed to know, namely, v, we can already compute what he/she sees in the protocol, namely, s_1,

Secure Addition Analysis...:Why no new information except the result v is leaked ?

- Let us concentrate on P_1 for concreteness. In step 1 x_1, x_2, and x_3 are secrets shared, and this tells P_1 nothing whatsoever about x_2, x_3
- In the final step, s_1, s_2, s_3 are announced. But,
  - P_1 already knows s_2,s_3, so s_1 is the only new piece of information.
  - However, seeing s_1 will tell P_1 what v is and nothing more. How ?
  - Because, if one is given s_2,s_3, and v, one can compute $s_1 = v - s_2 - s_3 \bmod p$.
  - That is, given what P_1 is supposed to know, namely, v, we can already compute what he/she sees in the protocol, namely, s_1,
  - Therefore, seeing the information from the protocol tells him or her nothing beyond v.

Secure Addition Analysis...:Why no new information except the result v is leaked ?

- Let us concentrate on P_1 for concreteness. In step 1 x_1, x_2, and x_3 are secrets shared, and this tells P_1 nothing whatsoever about x_2, x_3
- In the final step, s_1, s_2, s_3 are announced. But,
  - P_1 already knows s_2,s_3, so s_1 is the only new piece of information.
  - However, seeing s_1 will tell P_1 what v is and nothing more. How ?
  - Because, if one is given s_2,s_3, and v, one can compute $s_1 = v$ - s_2 - s_3 mod p.
  - That is, given what P_1 is supposed to know, namely, v, we can already compute what he/she sees in the protocol, namely, s_1,
  - Therefore, seeing the information from the protocol tells him or her nothing beyond v.
- Can't P_1 compute some information about other people's votes? That is in particular, can't P_1 compute $v - x_1 = x_2 + x_3$, i.e. the sum of the other players' votes?.

Secure Addition Analysis...:Why no new information except the result v is leaked ?

- Let us concentrate on P_1 for concreteness. In step 1 x_1, x_2, and x_3 are secrets shared, and this tells P_1 nothing whatsoever about x_2, x_3
- In the final step, s_1, s_2, s_3 are announced. But,
    - P_1 already knows s_2,s_3, so s_1 is the only new piece of information.
    - However, seeing s_1 will tell P_1 what v is and nothing more. How ?
    - Because, if one is given s_2,s_3, and v, one can compute $s_1 = v$ - s_2 - s_3 mod p.
    - That is, given what P_1 is supposed to know, namely, v, we can already compute what he/she sees in the protocol, namely, s_1,
    - Therefore, seeing the information from the protocol tells him or her nothing beyond v.
- Can't P_1 compute some information about other people's votes? That is in particular, can't P_1 compute $v - x_1 = x_2 + x_3$, i.e. the sum of the other players' votes?.
- Indeed, P_1 can compute the sum of the votes of P_2 and P_3, but this is what P_1 is *supposed to know - the result and his or her own input*.

Secure Addition Analysis...:Why no new information except the result v is leaked ?

- Let us concentrate on P_1 for concreteness. In step 1 x_1, x_2, and x_3 are secrets shared, and this tells P_1 nothing whatsoever about x_2, x_3
- In the final step, s_1, s_2, s_3 are announced. But,
  - P_1 already knows s_2,s_3, so s_1 is the only new piece of information.
  - However, seeing s_1 will tell P_1 what v is and nothing more. How ?
  - Because, if one is given s_2,s_3, and v, one can compute $s_1 = v - s_2 - s_3 \bmod p$.
  - That is, given what P_1 is supposed to know, namely, v, we can already compute what he/she sees in the protocol, namely, s_1,
  - Therefore, seeing the information from the protocol tells him or her nothing beyond v.
- Can't P_1 compute some information about other people's votes? That is in particular, can't P_1 compute $v - x_1 = x_2 + x_3$, i.e. the sum of the other players' votes?.
- Indeed, P_1 can compute the sum of the votes of P_2 and P_3, but this is what P_1 is *supposed to know - the result and his or her own input*.
- There is nothing the protocol can do to deprive P_1 of such information.....

Revisiting Sugar Beet Farmer's application: Protocol for Secure Auctions

A typical protocol for Secure auctions for Sugar Beet Farmers in Denmark

Then, the protocol followed is as follows:

1. Assume that input clients I_1, ..., I_m deliver inputs x_{ij}'s OR y_{ij}'s to a multiparty computation system to be executed by servers P_1 , . . . ,P_n
2. Then, the secure computation consists of computing the total demand and supply at each price, namely

$$d_j = \sum_i x_{ij},$$

$$s_j = \sum_i y_{ij},$$

$$j = 1, ..., P$$

3. Finally, compute the index j_0 for which d_{j_0} - s_{j_0} $= 0$, i.e. an index where the difference is as close to 0 as possible.

Augmenting ML for Privacy Preservation: Zero Knowledge Proofs