Towards Efficiency-Preserving Round Compression in MPC

Do fewer rounds mean more computation?

Prabhanjan Ananth
University of California Santa Barbara

Arka Rai Choudhuri
Johns Hopkins University

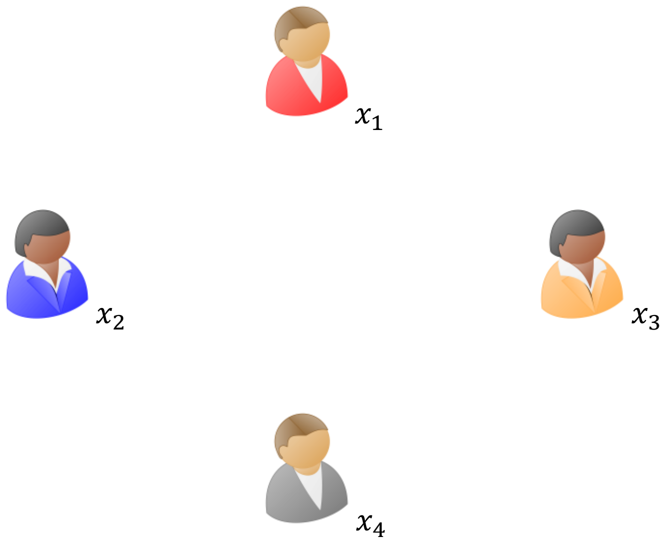
Aarushi Goel
Johns Hopkins University

Abhishek Jain
Johns Hopkins University

ASIACRYPT 2020

Multiparty Computation (MPC)

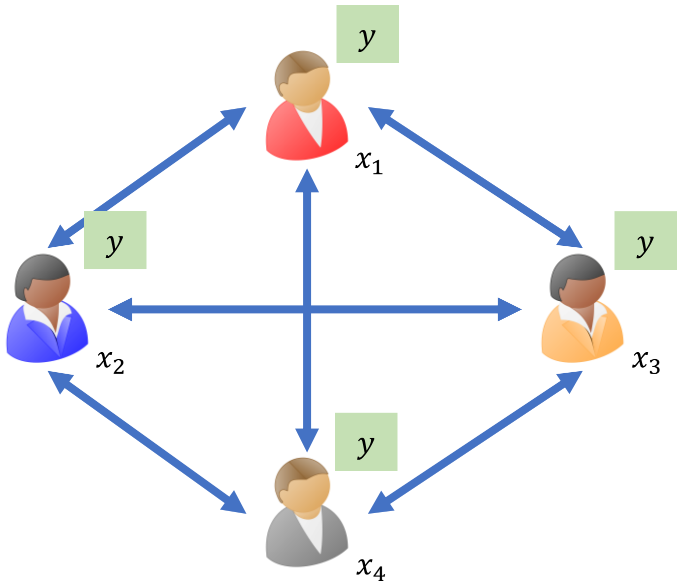
[Yao’86, Goldreich-Micali-Wigderson’87]



$$y = \mathcal{F}(x_1, x_2, x_3, x_4)$$

Multiparty Computation (MPC)

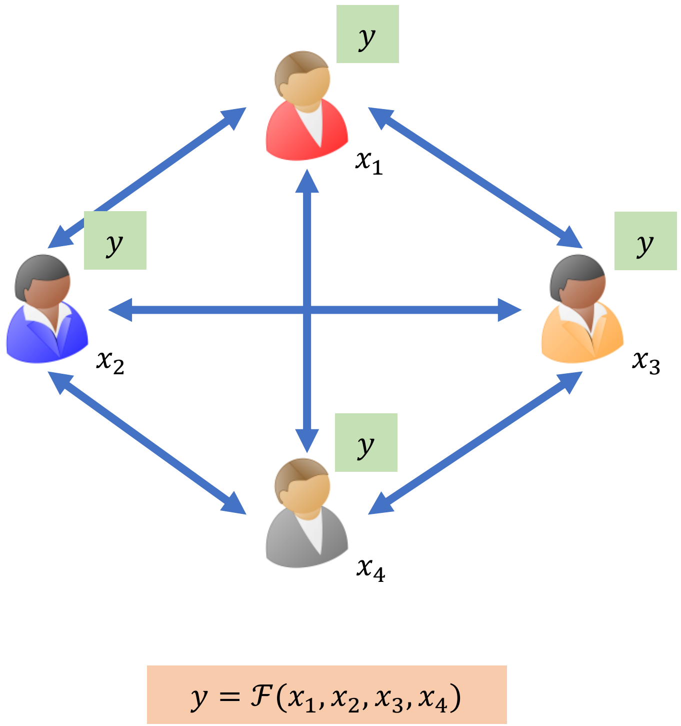
[Yao'86, Goldreich-Micali-Wigderson'87]



$y = \mathcal{F}(x_1, x_2, x_3, x_4)$

Multiparty Computation (MPC)

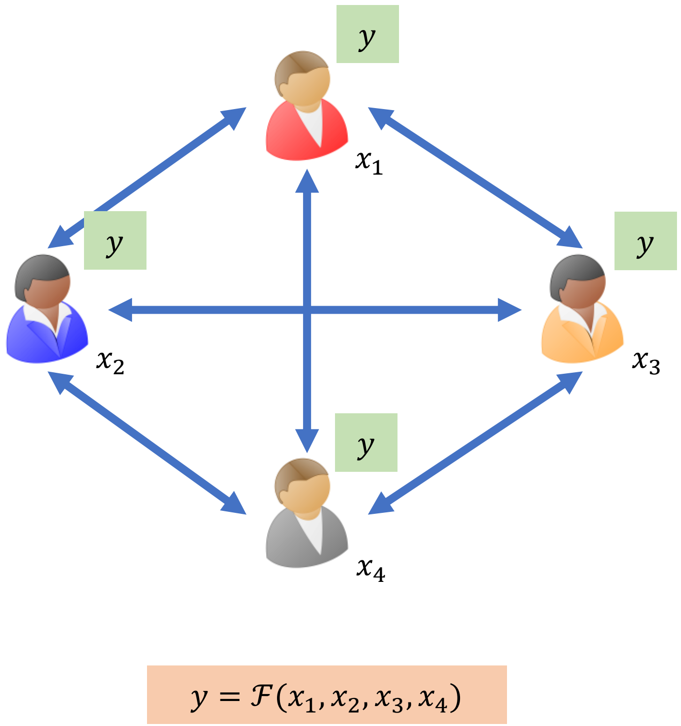
[Yao’86, Goldreich-Micali-Wigderson’87]



A **round** constitutes of every participant sending a message.

Multiparty Computation (MPC)

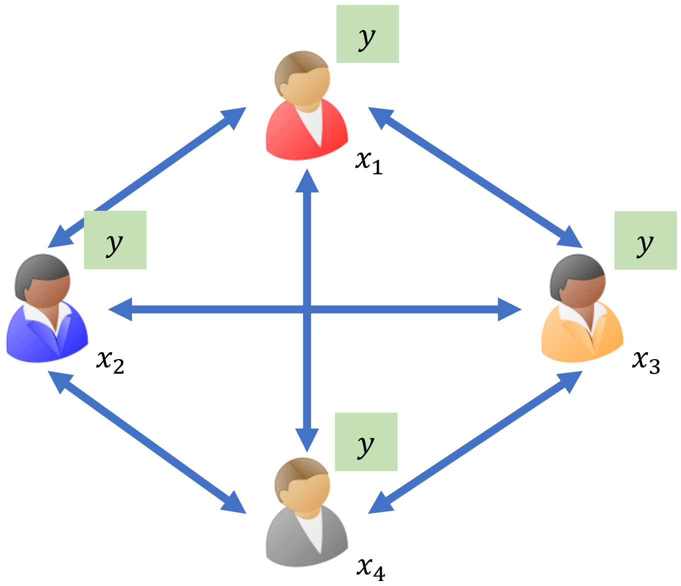
[Yao'86, Goldreich-Micali-Wigderson'87]



A **round** constitutes of every participant sending a message.

Goal: For efficiency, **minimize rounds of interaction**.

Security



$$y = \mathcal{F}(x_1, x_2, x_3, x_4)$$

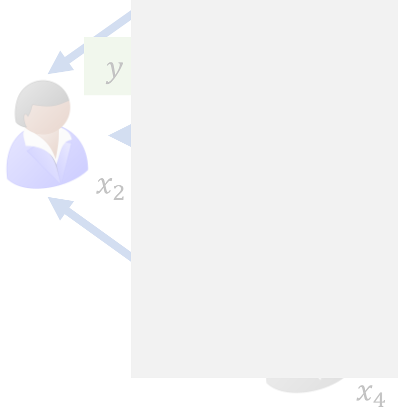
Misbehaving participants should not learn anything beyond the output of the function.

Security

Honest majority of participants.

Computational security.

hould not
output of



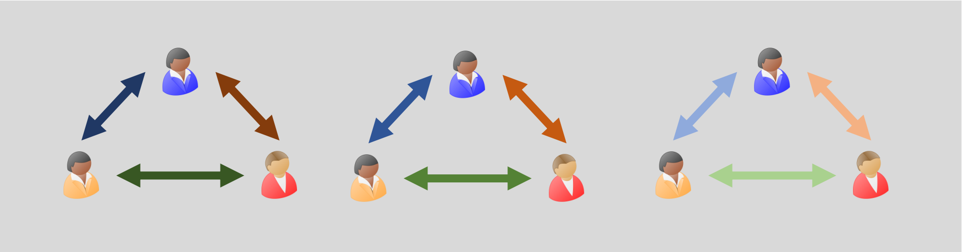
$y = \mathcal{F}(x_1, x_2, x_3, x_4)$

Round Complexity

Theorem [Ananth-C-Goel-Jain'18',19, Applebaum-Brakerski-Tsabary'18,'19, Garg-Ishai-Srinivasan'18]

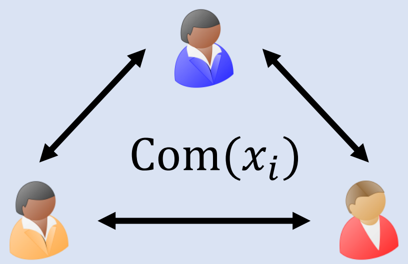
There exist two round protocols in the honest majority setting from minimal assumptions.

Round Collapse Methodology Overview



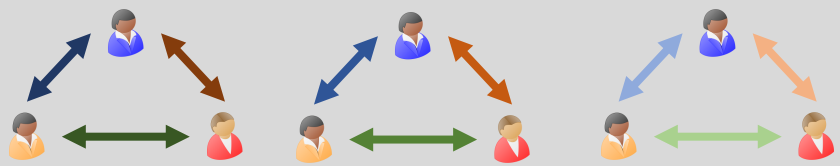
Multi-round protocol computing $\mathcal{F}$

Round Collapse Methodology Overview



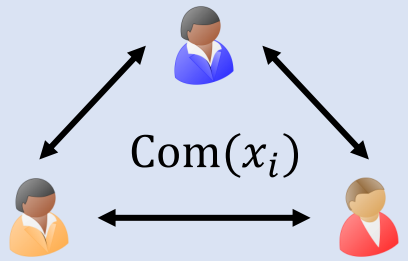
Exchange commitments of inputs.

Round 1



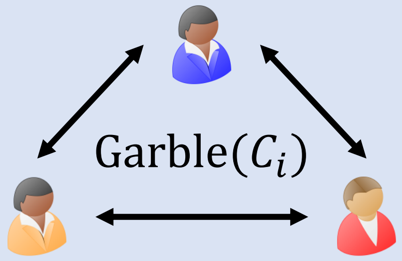
Multi-round protocol computing $\mathcal{F}$

Round Collapse Methodology Overview



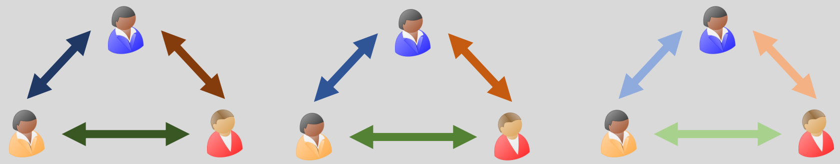
Exchange commitments of inputs.

Round 1



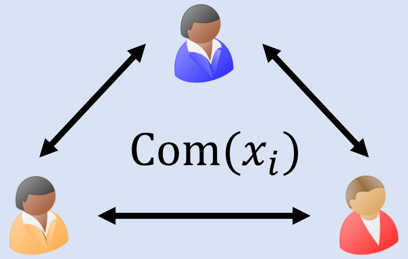
Exchange garbled circuits that will act as proxy in the multi-round protocol.

Round 2



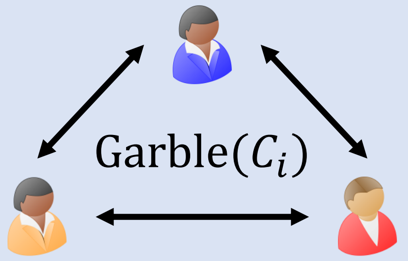
Multi-round protocol computing $\mathcal{F}$

Round Collapse Methodology Overview



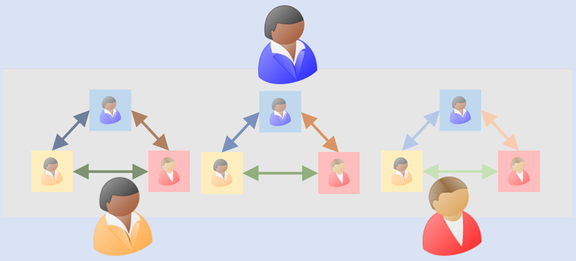
Exchange commitments of inputs.

Round 1



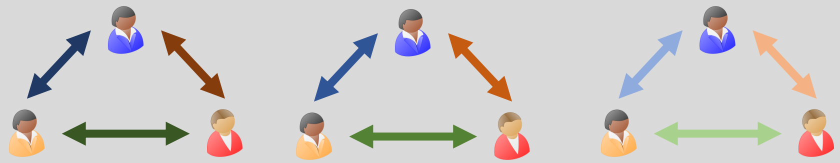
Exchange garbled circuits that will act as proxy in the multi-round protocol.

Round 2



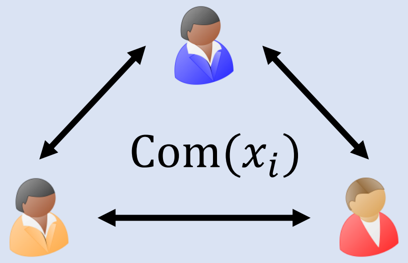
Locally execute multi-round protocol with garbled circuits as proxy for each party.

End of Round 2



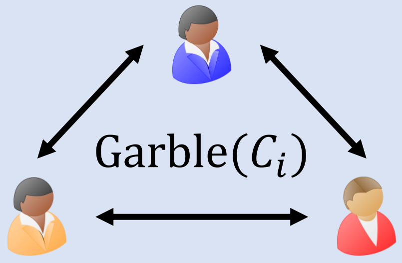
Multi-round protocol computing $\mathcal{F}$

Round Collapse Methodology Overview



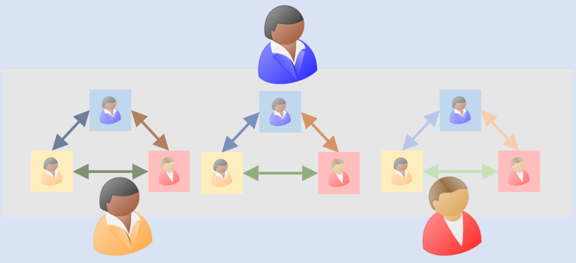
Exchange commitments of inputs.

Round 1



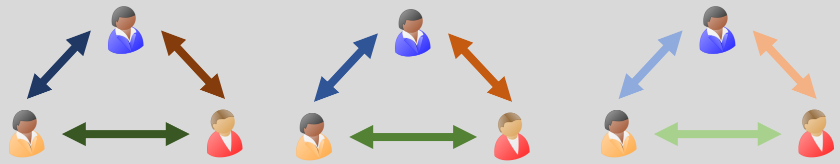
Exchange garbled circuits that will act as proxy in the multi-round protocol.

Round 2



Locally execute multi-round protocol with garbled circuits as proxy for each party.

End of Round 2



Multi-round protocol computing $\mathcal{F}$

Additional two round protocol executed in parallel to obtain appropriate keys to the garbled circuits.

Costs of the Two Round Protocols

If total computational work of the underlying protocol is $W(n, |C|)$ then existing compilers yield a two round protocol with total communication and per-party computation at least $\tilde{O}(n^2 \cdot W(n, |C|))$.

$|C|$ - size of circuit representing the function $\mathcal{F}$ to be computed.
n – number of parties

Costs of the Two Round Protocols

If total computational work of the underlying protocol is $W(n, |C|)$ then existing compilers yield a two round protocol with total communication and per-party computation at least $\tilde{O}(n^2 \cdot W(n, |C|))$.

The $\tilde{O}$ notation hides polylogarithmic factors in the number of parties n and polynomial factors in the security paramter λ.

$|C|$ - size of circuit representing the function $\mathcal{F}$ to be computed.
n – number of parties

Costs of the Two Round Protocols

If total computational work of the underlying protocol is $\tilde{O}(|C| + nd)$ then existing compilers yield a two round protocol with total communication and per-party computation at least $\tilde{O}(n^2|C| + n^3 d)$.

Plugging in most efficient semi-honest protocols where $W(n, |C|) = \tilde{O}(|C| + nd)$ [Genkin-Ishai-Polychroniadou'15, Damgård-Ishai-Krøigaard-Nielsen-Smith'08, , Damgård-Ishai-Krøigaard'10]

Can we construct efficiency-preserving round compression compilers?

Efficiency measured as the total communication or per-party computation.

Our Results

If total computational work of the underlying protocol is $W(n, |C|)$ then our compiler produces a two round semi-honest protocol with total communication and per-party computation at least $\tilde{O}(W(\log^2(n), |C|) + n^4)$.

If total computational work of the underlying protocol is $W(n, |C|)$ then existing compilers yield a three round protocol with total communication and per-party computation at least $\tilde{O}(W(\log^2(n), |C|) + n^6)$.

Our Results

If total computational work of the underlying protocol is $W(n, |C|)$ then our compiler produces a two round semi-honest protocol with total communication and per-party computation at least $\tilde{O}(W(\log^2(n), |C|) + n^4)$.

If total computational work of the underlying protocol is $W(n, |C|)$ then existing compilers yield a three round protocol with total communication and per-party computation at least $\tilde{O}(W(\log^2(n), |C|) + n^6)$.

Our Results

	Semi-honest	Malicious*
Prior work	$\tilde{O}(n^2\lvert C\rvert + n^3 d)$	$\tilde{O}(n^2\lvert C\rvert + n^3 d + n^4)$
Our work	$\tilde{O}(\lvert C\rvert + n^4)$	$\tilde{O}(\lvert C\rvert + n^6)$

Total communication and per-party computation costs of resultant protocol.

Plugging in most efficient protocols where $W(n, |C|)$ is

1. $\tilde{O}(|C| + nd)$ for semi-honest
2. $\tilde{O}(|C| + nd + n^2)$ for malicious

* Malicious protocols in prior work only require two rounds.

Our Results

	Semi-honest	Malicious*
Prior work	$\tilde{O}(n^2\lvert C\rvert + n^3 d)$	$\tilde{O}(n^2\lvert C\rvert + n^3 d + n^4)$
Our work	$\tilde{O}(\lvert C\rvert + n^4)$	$\tilde{O}(\lvert C\rvert + n^6)$

Total communication and per-party computation costs of resultant protocol.

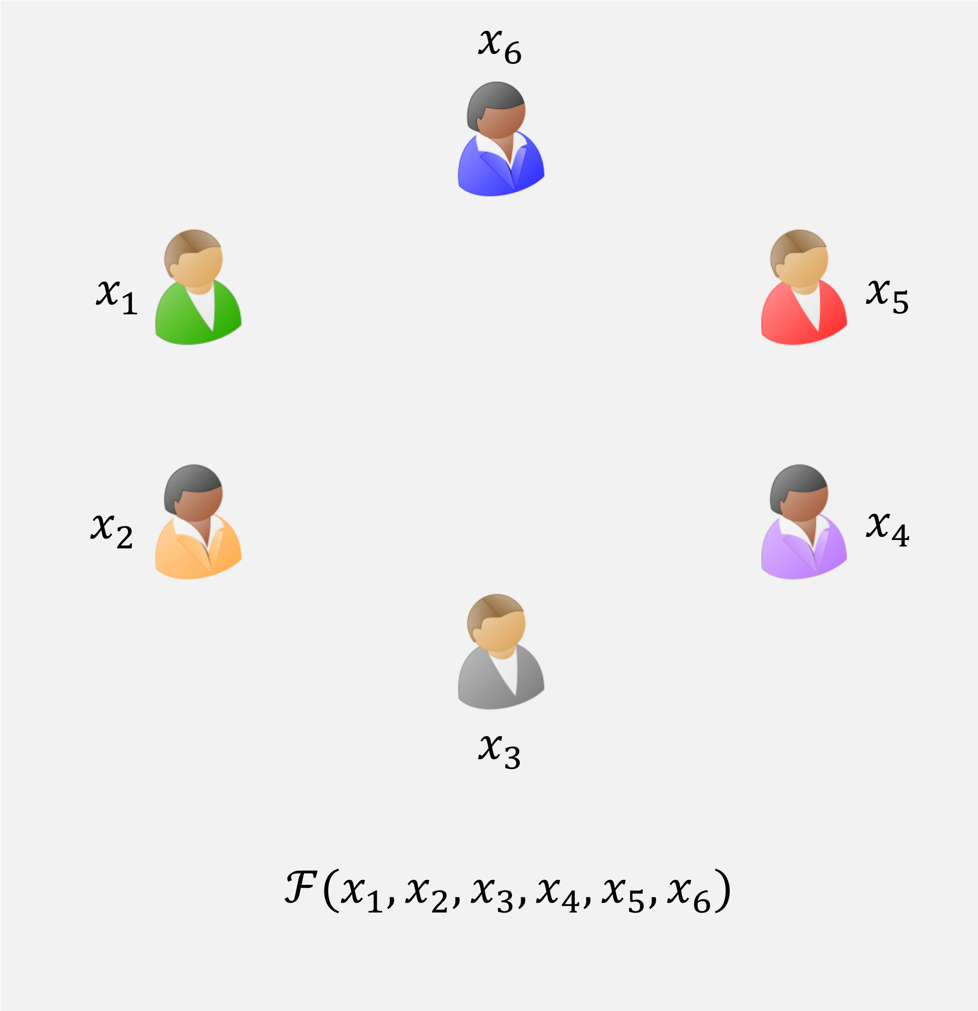
Plugging in most efficient protocols where $W(n, |C|)$ is

1. $\tilde{O}(|C| + nd)$ for semi-honest
2. $\tilde{O}(|C| + nd + n^2)$ for malicious

Total computation cost can be made to match total communication costs with an additional round.

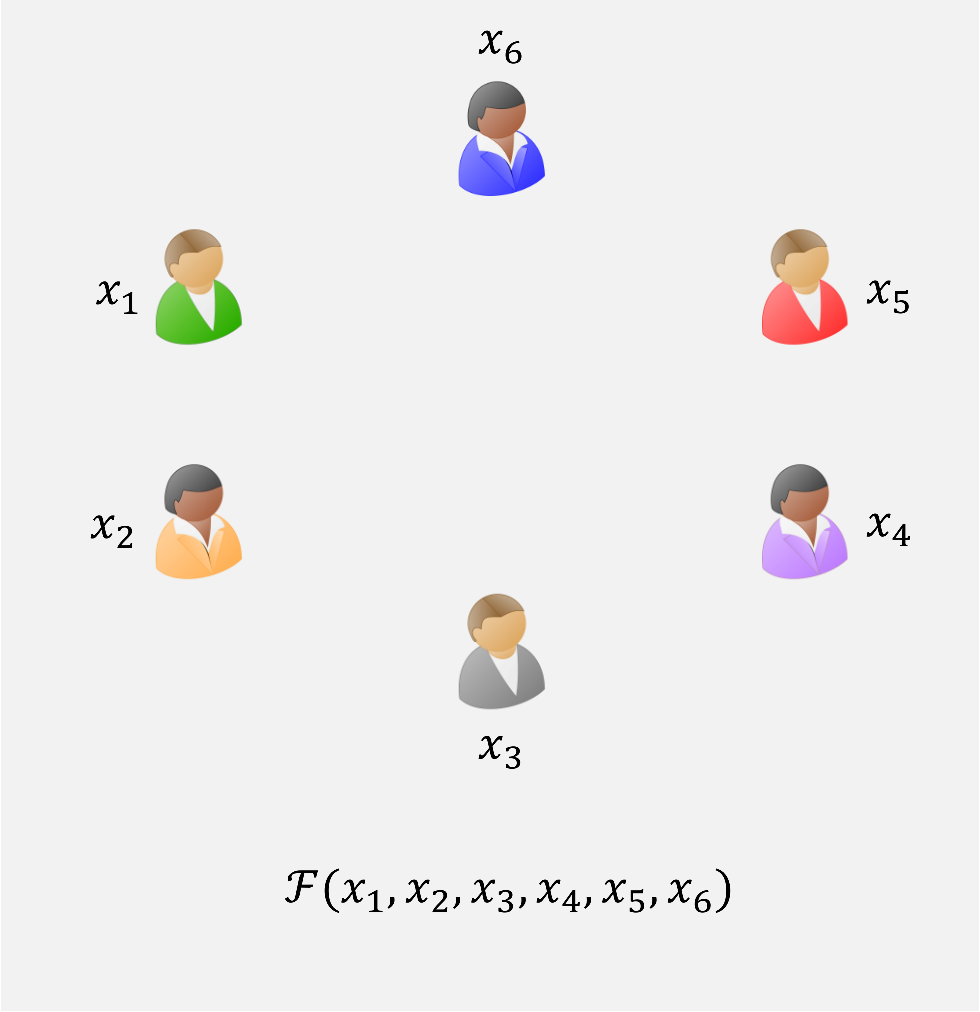
* Malicious protocols in prior work only require two rounds.

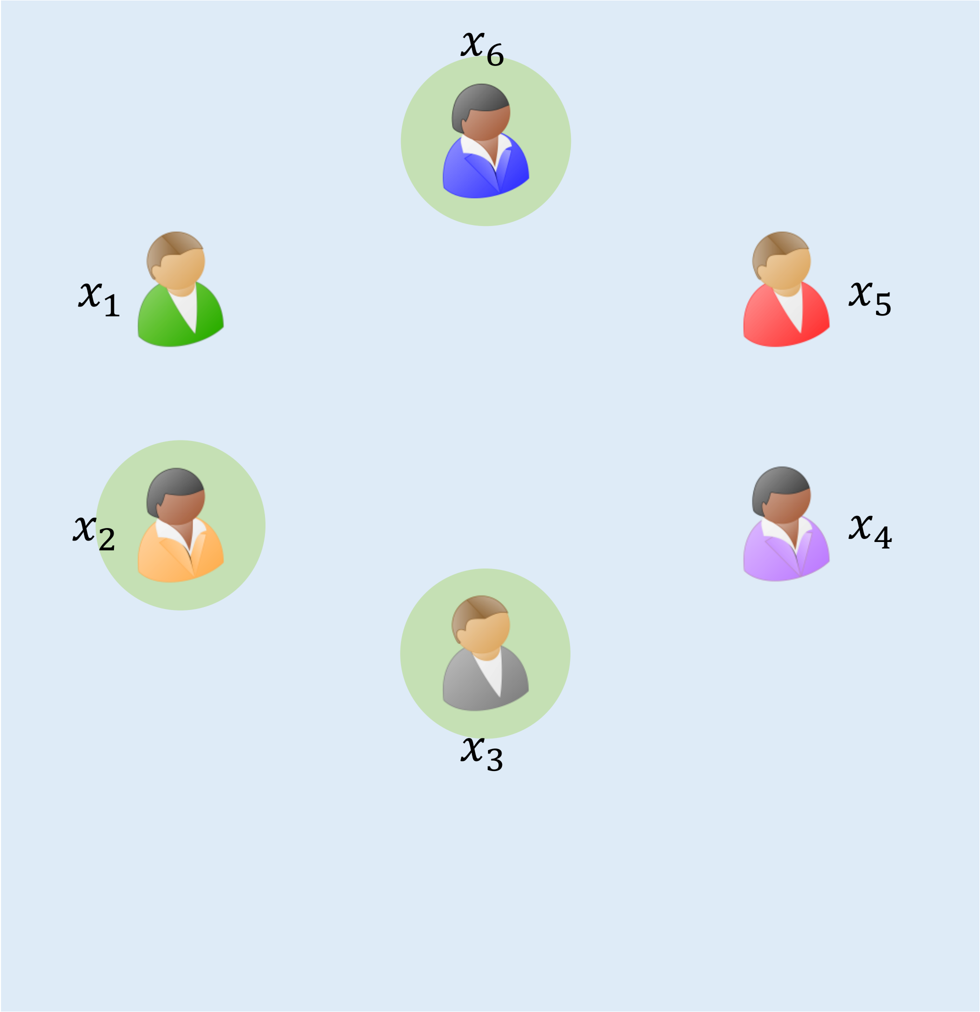
Natural Approach: Delegation of Computation



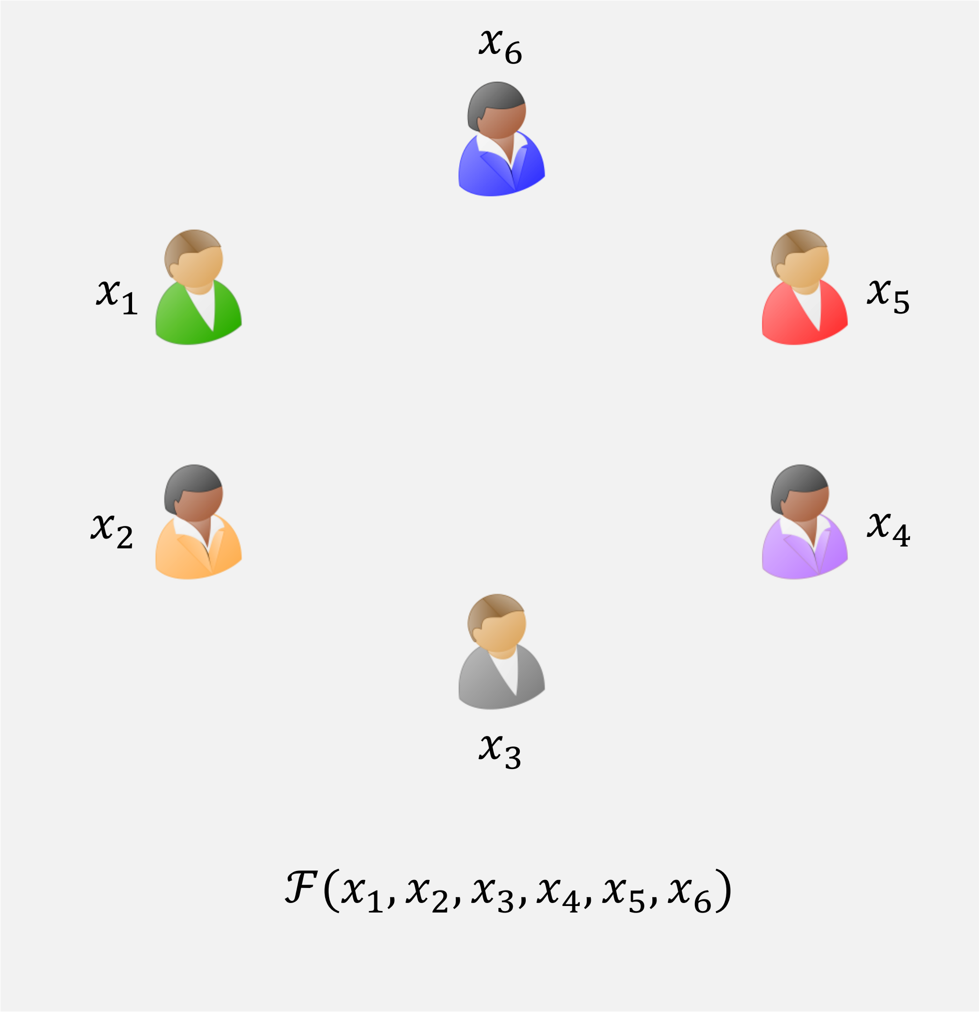
Elect a committee of servers to delegate the heavy computation.

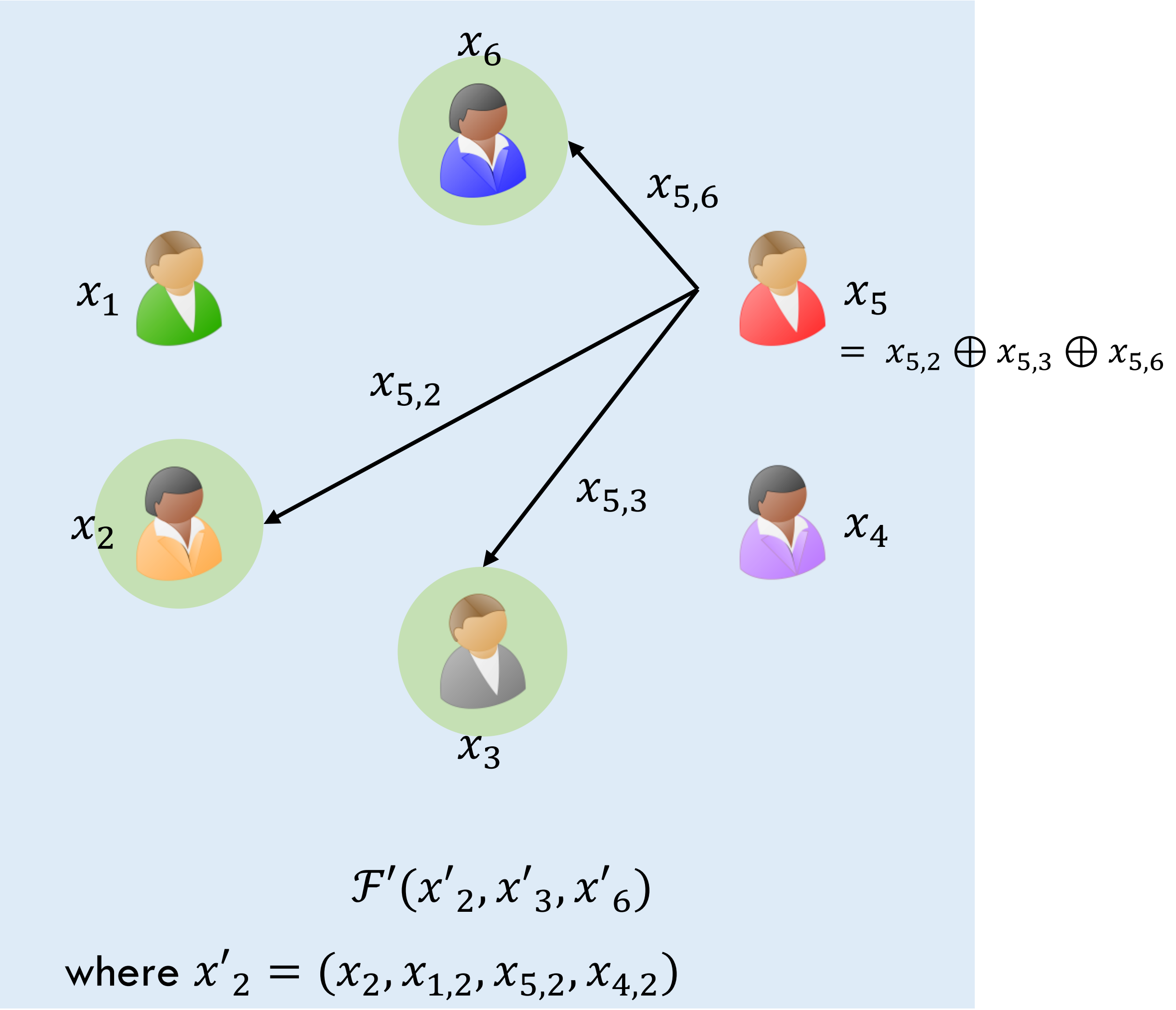
Natural Approach: Delegation of Computation





Natural Approach: Delegation of Computation





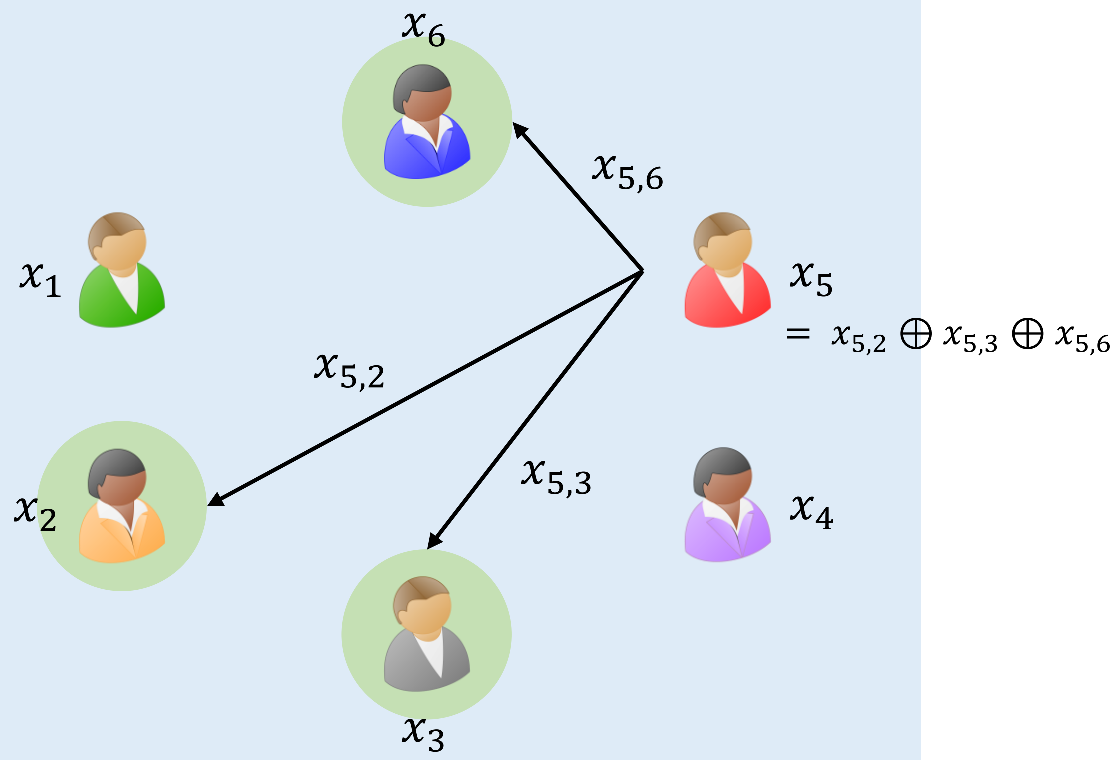
$\mathcal{F}'$: reconstruct client input from shares and compute $\mathcal{F}$ on inputs.

Natural Approach: Delegation of Computation

Delegation idea inherent:

For some functions, there does not exist a constant round balanced protocol where the total computational cost is $\tilde{O}(|C|)$.

We use MPC-in-the-head techniques for the lower bound.



$\mathcal{F}'(x'_2, x'_3, x'_6)$

where $x'_2 = (x_2, x_{1,2}, x_{5,2}, x_{4,2})$

$\mathcal{F}'$: reconstruct client input from shares and compute $\mathcal{F}$ on inputs.

Challenges in two rounds

1. Servers must commit to the input in the first round.

   Servers not in possession of complete input - Committee election and input sharing must happen in the first round.

Challenges in two rounds

1. Servers must commit to the input in the first round.

   Servers not in possession of complete input - Committee election and input sharing must happen in the first round.

2. Known compilers require private communication between servers.

   Servers do not know the identity of other servers in the first round.

Challenges in two rounds

1. Servers must commit to the input in the first round.

   Serv… …n and

   input…

2. Knov… …en

   servers.

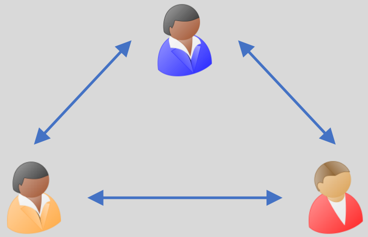
   Servers do not know the identity of other servers in the first round.

Main Idea:

Round Efficient Approach to Delegation of Computation

Special Two Round MPC Protocols

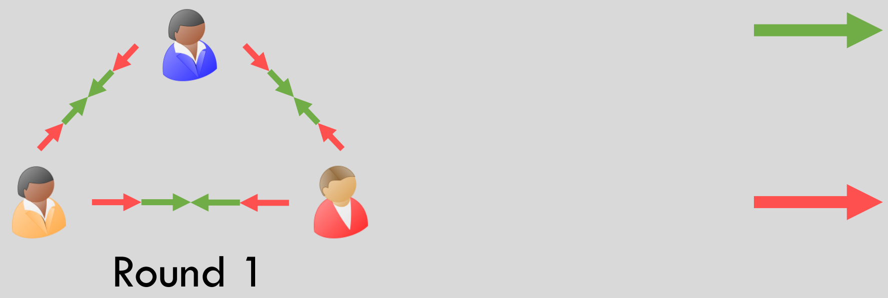
Decomposability



Round 1

Special Two Round MPC Protocols

Decomposability of first round messages

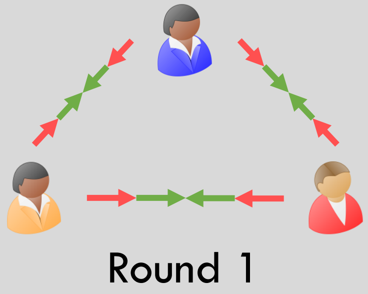


Light messages – depend on the input computational complexity independent of W.

Heavy messages – independent on the input computational complexity depends on W.

Special Two Round MPC Protocols

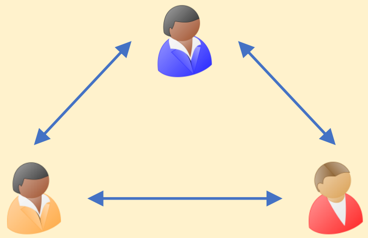
Decomposability of first round messages



Light messages – depend on the input computational complexity independent of W.

Heavy messages – independent on the input computational complexity depends on W.

Independence



Private channel messages between parties independent of the input.

Special Two Round MPC Protocols

We show how existing compilers can be suitably modified to achieve these properties.

Decomposability of first round messages

Light messages – depend on the input computational complexity independent of W.

Heavy messages – independent on the input computational complexity depends on W.

Independence

Private channel messages between parties independent of the input.

High Level Strategy

1. Parties self-elect into committee.

$\mathcal{F}'$: reconstruct client input from shares and compute $\mathcal{F}$ on inputs.

Decomposability of first round messages

Light messages – depend on the input computational complexity independent of W.

Heavy messages – independent on the input computational complexity depends on W.

Independence

Private channel messages between parties independent of the input.

High Level Strategy

1. Parties self-elect into committee.

2. Servers run special MPC protocol computing $\mathcal{F}'$.

$\mathcal{F}'$: reconstruct client input from shares and compute $\mathcal{F}$ on inputs.

Decomposability of first round messages

Light messages – depend on the input computational complexity independent of W.

Heavy messages – independent on the input computational complexity depends on W.

Independence

Private channel messages between parties independent of the input.

High Level Strategy

1. Parties self-elect into committee.

2. Servers run special MPC protocol computing $\mathcal{F}'$.

3. All clients help compute light messages.
   Decomposability keeps total cost low.

$\mathcal{F}'$: reconstruct client input from shares and compute $\mathcal{F}$ on inputs.

Decomposability of first round messages

Light messages – depend on the input computational complexity independent of W.

Heavy messages – independent on the input computational complexity depends on W.

Independence

Private channel messages between parties independent of the input.

High Level Strategy

1. Parties self-elect into committee.

2. Servers run special MPC protocol computing $\mathcal{F}'$.

3. All clients help compute light messages.

   Decomposability keeps total cost low.

4. Servers broadcast "encrypted" private channel messages.

   Independence allows this to be possible.

$\mathcal{F}'$: reconstruct client input from shares and compute $\mathcal{F}$ on inputs.

Decomposability of first round messages

Light messages – depend on the input computational complexity independent of W.

Heavy messages – independent on the input computational complexity depends on W.

Independence

Private channel messages between parties independent of the input.

High Level Strategy

1. Parties self-elect into committee.

2. Servers run special MPC protocol computing $\mathcal{F}'$.

3. All clients help compute light messages.

   Decomposability keeps total cost low.

4. Servers broadcast "encrypted" private channel messages.

   Independence allows this to be possible.

$\mathcal{F}'$: reconstruct client input from shares and compute $\mathcal{F}$ on inputs.

Decomposability of first round messages

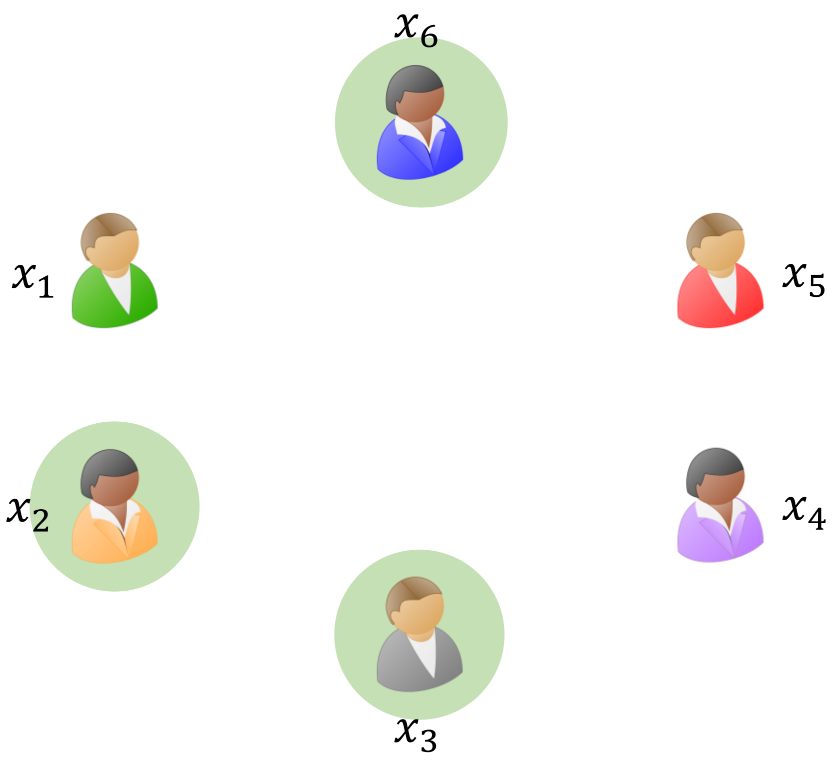
Light messages – depend on the input computational complexity independent of W.

Heavy messages – independent on the input computational complexity depends on W.

Independence

Private channel messages between parties independent of the input.

Helper Computation by the Clients



Servers computation:

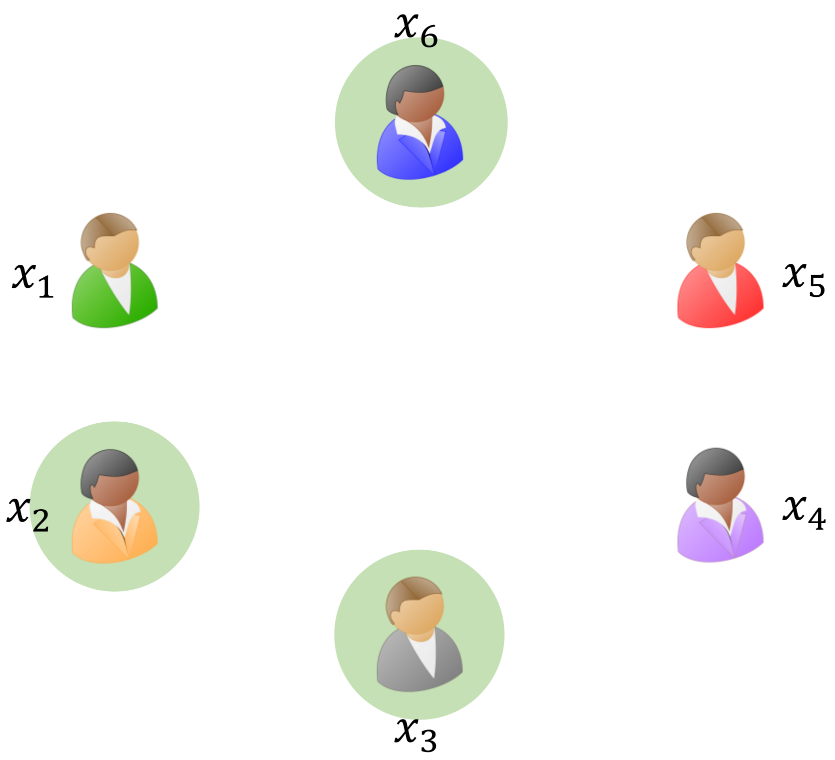
First round

1. Light messages dependent on input.
2. Heavy messages independent of input.

Second Round

1. Second round message that depends on entire first round message.

Helper Computation by the Clients



Servers computation:

First round

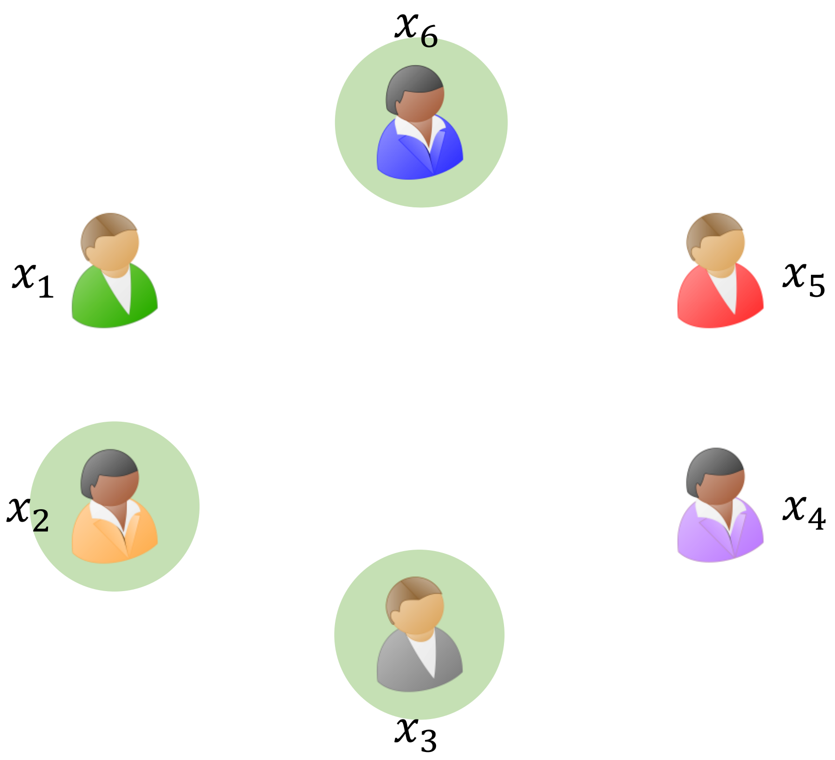
1. Light messages dependent on input.

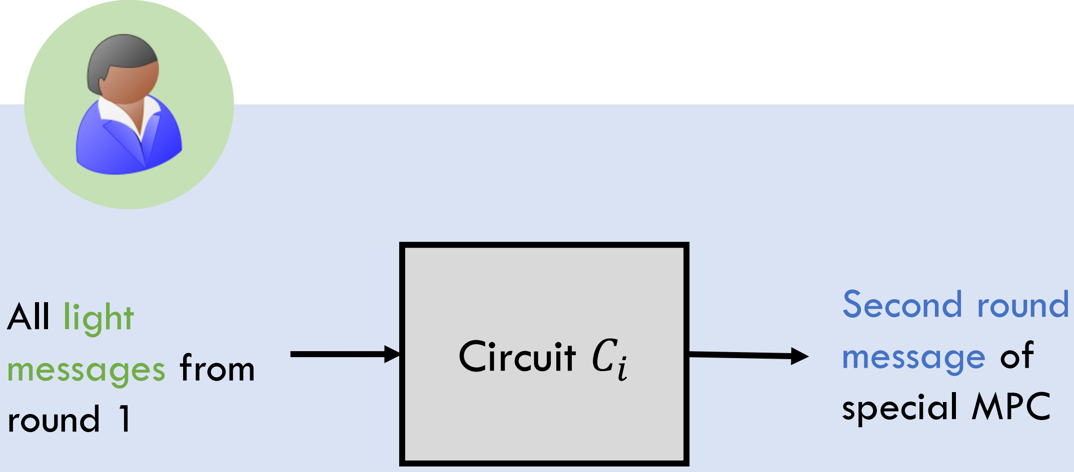
~~2. Heavy messages independent of input.~~

Second Round

1. Second round message that depends on entire first round message.

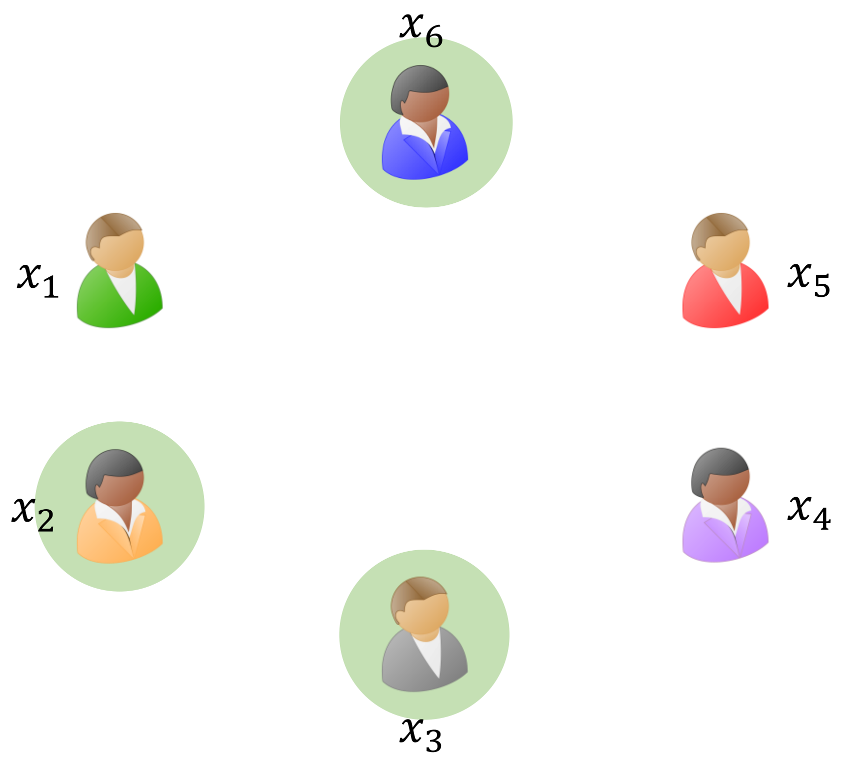
Helper Computation by the Clients

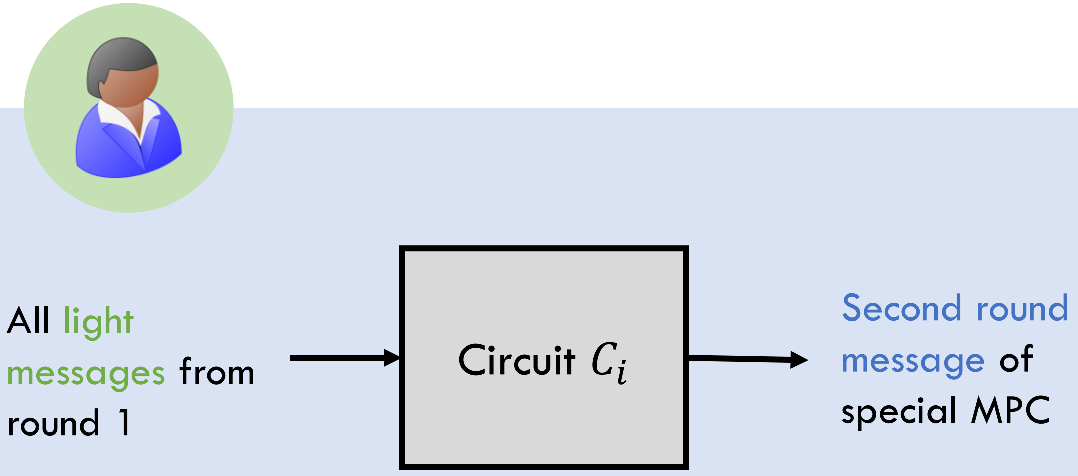




Each server garbles this circuit and sends it in the second round of the protocol.

Helper Computation by the Clients

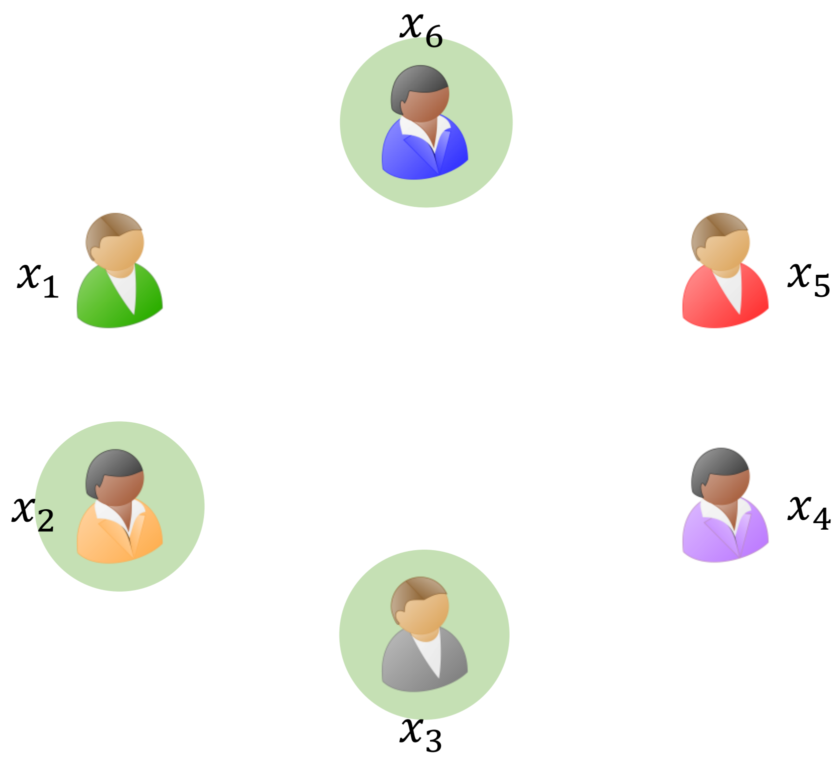




Each server garbles this circuit and sends it in the second round of the protocol.

Need mechanism to deliver labels to evaluate the circuit.

Helper Computation by the Clients



All parties run two round Helper protocol

1. Client Inputs: shares of input x_i.
2. Server Inputs: labels of the garbled circuit.

Protocol Output: labels corresponding to the light messages.

Mechanism to deliver labels to evaluate the circuit.

Helper Computation by the Clients

Helper protocol properties

1. Does not require knowledge of servers
   All parties participate.
2. Computation of only light messages
   Additional overhead is low.

All parties run two round Helper protocol

1. Client Inputs: shares of input x_i.
2. Server Inputs: labels of the garbled circuit.

Protocol Output: labels corresponding to the light messages.

High Level Strategy

1. Parties self-elect into committee.

2. Servers run special MPC protocol computing $\mathcal{F}'$.

3. All clients help compute light messages.
   Decomposability keeps total cost low.

4. Servers broadcast "encrypted" private channel messages.
   Independence allows this to be possible.

$\mathcal{F}'$: reconstruct client input from shares and compute $\mathcal{F}$ on inputs.

Decomposability of first round messages

Light messages – depend on the input computational complexity independent of W.

Heavy messages – independent on the input computational complexity depends on W.

Independence

Private channel messages between parties independent of the input.

High Level Strategy

$\mathcal{F}'$: reconstruct client input from shares and compute $\mathcal{F}$ on inputs.

1. Parties self-elect into committee.

For servers to obtain appropriate keys to decrypt broadcast message, run another helper protocol with all parties.

Similar to previously discussed approach

3. All clients help compute light messages.

Decomposability keeps total cost low.

4. Servers broadcast "encrypted" private channel messages.

Independence allows this to be possible.

Decomposability of first round messages

Light messages – depend on the input computational complexity independent of W.

Heavy messages – independent on the input computational complexity depends on W.

Independence

Private channel messages between parties independent of the input.

Towards Achieving Malicious Security

Malicious protocols similar ideas but requires:

- Special MPC to be maliciously secure
- Committee Election robust to malicious behavior
  - Additional round OR Setup assumptions

Thank you. Questions?

Arka Rai Choudhuri

achoud@cs.jhu.edu

ia.cr/2020/1100