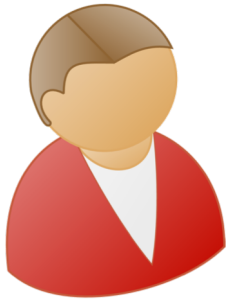
Round-Optimal Secure Multiparty Computation with Honest Majority

Prabhanjan Ananth Arka Rai Choudhuri **Aarushi Goel** Abhishek Jain

CRYPTO 2018

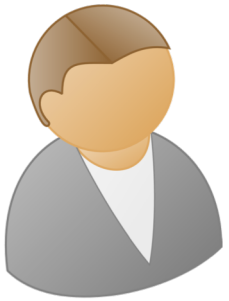
Secure Multiparty Computation

x_2



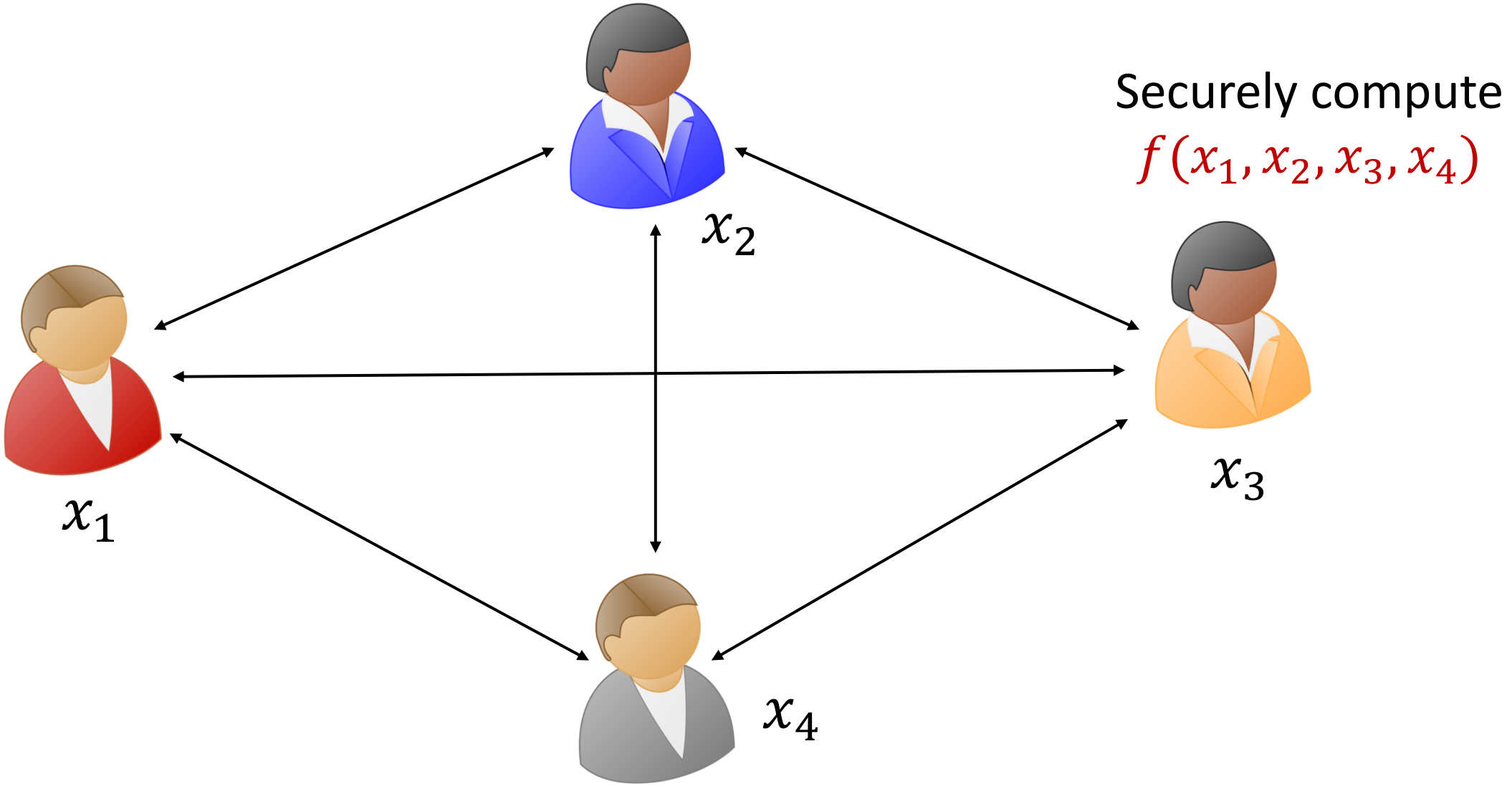
x_1

x_3

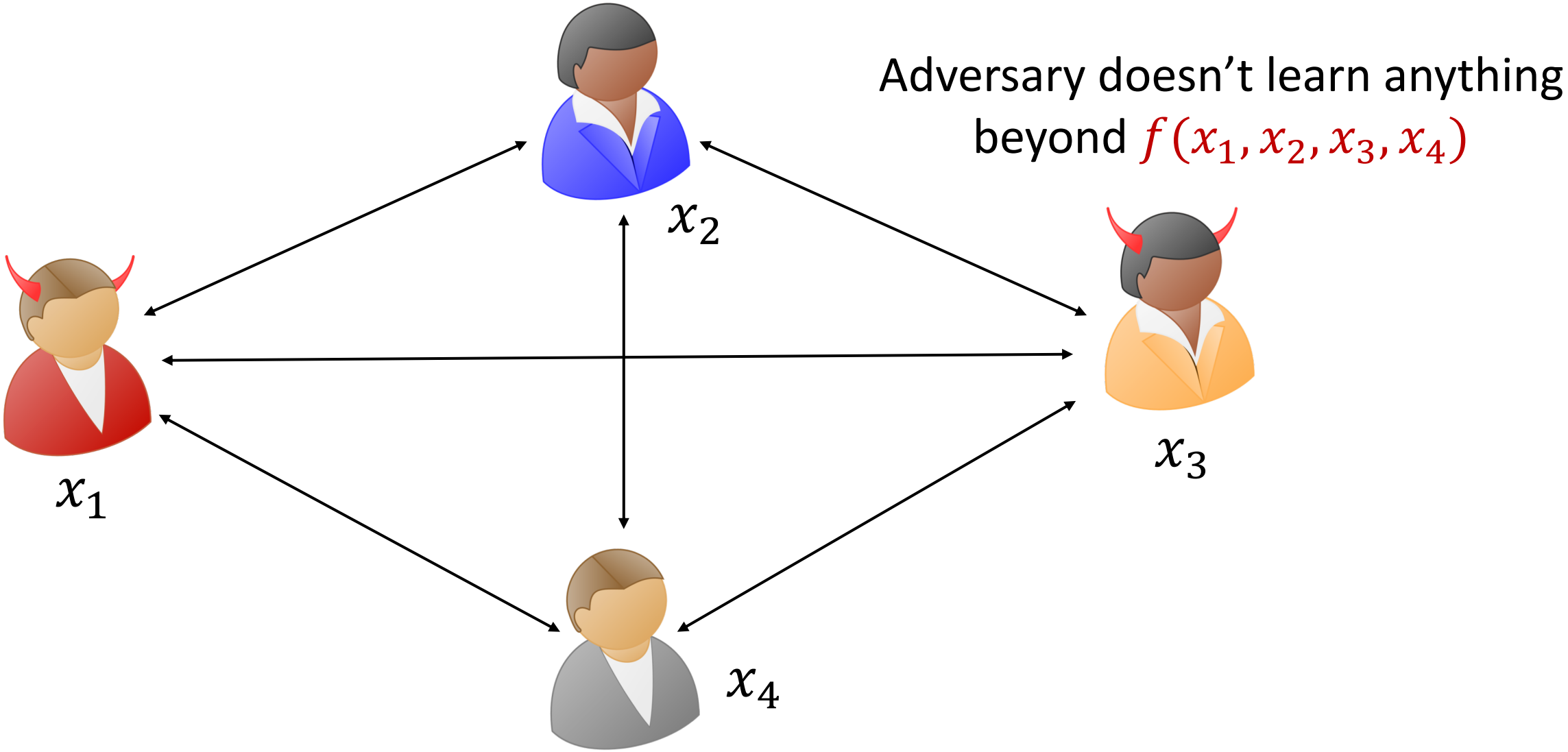


x_4

Secure Multiparty Computation



Secure Multiparty Computation



Honest Majority MPC

Honest Majority MPC (up to $t < N/2$ corrupted parties)

Honest Majority MPC (up to $t < N/2$ corrupted parties)

- Oblivious Transfer is not necessary.

  Necessary for dishonest majority [Kil88].

Honest Majority MPC (up to $t < N/2$ corrupted parties)

- Oblivious Transfer is not necessary.

  Necessary for dishonest majority [Kil88].

- Fairness and Guaranteed output delivery can be achieved.

Honest Majority MPC (up to $t < N/2$ corrupted parties)

- Oblivious Transfer is not necessary.

  Necessary for dishonest majority [Kil88].

- Fairness and Guaranteed output delivery can be achieved.

- UC security without external trusted setups

Honest Majority MPC (up to $t < N/2$ corrupted parties)

- Oblivious Transfer is not necessary.

  Necessary for dishonest majority [Kil88].

- Fairness and Guaranteed output delivery can be achieved.

- UC security without external trusted setups

- Round complexity lower bounds of dishonest majority do not apply.

  4 rounds necessary for dishonest majority in the plain model [Garg-Mukherjee-Pandey-Polychroniadou16]

Problem Statement

What is the exact round complexity of honest majority MPC in the plain model?

Honest Majority MPC: Security Notions

Honest Majority MPC: Security Notions

- Security with Abort:

Honest Majority MPC: Security Notions

- Security with Abort:

  Adversary may learn the output but can prevent honest parties from doing so.

Honest Majority MPC: Security Notions

- Security with Abort:

  Adversary may learn the output but can prevent honest parties from doing so.

- Guaranteed output Delivery:

Honest Majority MPC: Security Notions

- Security with Abort:

  Adversary may learn the output but can prevent honest parties from doing so.

- Guaranteed output Delivery:

  Honest parties always learn the output even if some parties abort prematurely.

Honest Majority MPC: Security Notions

- Security with Abort:

  Adversary may learn the output but can prevent honest parties from doing so.

- Guaranteed output Delivery:

  Honest parties always learn the output even if some parties abort prematurely.

  Guaranteed output delivery $\Longrightarrow$ Fairness

Honest Majority MPC: Security Notions

- Security with Abort:

  Adversary may learn the output but can prevent honest parties from doing so.

- Guaranteed output Delivery:

  Honest parties always learn the output even if some parties abort prematurely.

  Guaranteed output delivery $\Longrightarrow$ Fairness

Goal: Develop round optimal protocols in these settings.

Brief History: Security with Abort

Brief History: Security with Abort

Polynomial round protocols

- [Goldreich-Micali-Wigderson87, Ben-Or-Goldwasser-Wigderson88, Chaum-Crépeau-Damgård88]

Brief History: Security with Abort

Polynomial round protocols

- [Goldreich-Micali-Wigderson87, Ben-Or-Goldwasser-Wigderson88, Chaum-Crépeau-Damgård88]

Constant round protocols

- [Beaver-Micali-Rogaway90]

Brief History: Security with Abort

Polynomial round protocols

- [Goldreich-Micali-Wigderson87, Ben-Or-Goldwasser-Wigderson88, Chaum-Crépeau-Damgård88]

Constant round protocols

- [Beaver-Micali-Rogaway90]
- And subsequently many works investigated improvements.

Brief History: Security with Abort

Polynomial round protocols

- [Goldreich-Micali-Wigderson87, Ben-Or-Goldwasser-Wigderson88, Chaum-Crépeau-Damgård88]

Constant round protocols

- [Beaver-Micali-Rogaway90]
- And subsequently many works investigated improvements.

Two round protocols

- [Ishai-Kushilevitz00, Ishai-Kushilevitz-Paskin10]: Unconditional security, $t < N/3$ corruptions.

Brief History: Security with Abort

Polynomial round protocols

- [Goldreich-Micali-Wigderson87, Ben-Or-Goldwasser-Wigderson88, Chaum-Crépeau-Damgård88]

Constant round protocols

- [Beaver-Micali-Rogaway90]
- And subsequently many works investigated improvements.

Two round protocols

- [Ishai-Kushilevitz00, Ishai-Kushilevitz-Paskin10]: Unconditional security, $t < N/3$ corruptions.
- [Benhomouda-Lin17, Garg-Srinivasan17]: $t < N$ semi-honest corruptions based on OT. Malicious corruptions in the CRS model.

Question: Security with Abort

Does there exist a two round MPC protocol secure against $t < N/2$ malicious corruptions in the plain model?

Question: Security with Abort

Does there exist a two round MPC protocol secure against $t < N/2$ malicious corruptions in the plain model?

Open regardless of assumptions.

Question: Security with Abort

Does there exist a two round MPC protocol secure against $t < N/2$ malicious corruptions in the plain model?

Open regardless of assumptions.

Impossible for dishonest majority [Garg- Mukherjee-Pandey-Polychroniadou16]

Question: Security with Abort

Does there exist a two round MPC protocol secure against $t < N/2$ malicious corruptions in the plain model?

Open regardless of assumptions.

Impossible for dishonest majority [Garg- Mukherjee-Pandey-Polychroniadou16]

Open even in semi-honest case from assumptions weaker than OT.

Brief History: Guaranteed Output Delivery

Brief History: Guaranteed Output Delivery

Upper Bounds

- [Ben-Or-Goldwasser-Wigderson88, Chaum-Crépeau-Damgård88]: Feasibility

Brief History: Guaranteed Output Delivery

Upper Bounds

- [Ben-Or-Goldwasser-Wigderson88, Chaum-Crépeau-Damgård88]: Feasibility
- [Ishai-Kushilevitz-Paskin10]: Two-round MPC in the plain model with n>4, t=1 malicious corruptions from OWFs.
- [Ishai-Kumaresan-Kushilevitz-Paskin15]: Two-round MPC in the plain model with n=4, t=1 malicious corruptions from injective OWFs.

Brief History: Guaranteed Output Delivery

Upper Bounds

- [Ben-Or-Goldwasser-Wigderson88, Chaum-Crépeau-Damgård88]: Feasibility
- [Ishai-Kushilevitz-Paskin10]: Two-round MPC in the plain model with n>4, t=1 malicious corruptions from OWFs.
- [Ishai-Kumaresan-Kushilevitz-Paskin15]: Two-round MPC in the plain model with n=4, t=1 malicious corruptions from injective OWFs.
- [Gordon-Liu-Shi15]: Three-round maliciously secure protocol in the CRS model from LWE and NIZKs.

Brief History: Guaranteed Output Delivery

Upper Bounds

- [Ben-Or-Goldwasser-Wigderson88, Chaum-Crépeau-Damgård88]: Feasibility
- [Ishai-Kushilevitz-Paskin10]: Two-round MPC in the plain model with n>4, t=1 malicious corruptions from OWFs.
- [Ishai-Kumaresan-Kushilevitz-Paskin15]: Two-round MPC in the plain model with n=4, t=1 malicious corruptions from injective OWFs.
- [Gordon-Liu-Shi15]: Three-round maliciously secure protocol in the CRS model from LWE and NIZKs.

Lower Bounds

Brief History: Guaranteed Output Delivery

Upper Bounds

- [Ben-Or-Goldwasser-Wigderson88, Chaum-Crépeau-Damgård88]: Feasibility
- [Ishai-Kushilevitz-Paskin10]: Two-round MPC in the plain model with n>4, t=1 malicious corruptions from OWFs.
- [Ishai-Kumaresan-Kushilevitz-Paskin15]: Two-round MPC in the plain model with n=4, t=1 malicious corruptions from injective OWFs.
- [Gordon-Liu-Shi15]: Three-round maliciously secure protocol in the CRS model from LWE and NIZKs.

Lower Bounds

- [Gennaro-Ishai-Kushilevitz-Rabin'02]: Impossibility of two-round protocols with t>2 malicious corruptions in the plain model.

Brief History: Guaranteed Output Delivery

Upper Bounds

- [Ben-Or-Goldwasser-Wigderson88, Chaum-Crépeau-Damgård88]: Feasibility
- [Ishai-Kushilevitz-Paskin10]: Two-round MPC in the plain model with n>4, t=1 malicious corruptions from OWFs.
- [Ishai-Kumaresan-Kushilevitz-Paskin15]: Two-round MPC in the plain model with n=4, t=1 malicious corruptions from injective OWFs.
- [Gordon-Liu-Shi15]: Three-round maliciously secure protocol in the CRS model from LWE and NIZKs.

Lower Bounds

- [Gennaro-Ishai-Kushilevitz-Rabin'02]: Impossibility of two-round protocols with t>2 malicious corruptions in the plain model.
- [Gordon-Liu-Shi'15]: Impossibility of two-round broadcast channel protocols against fail-stop corruptions.

Question: Guaranteed Output Delivery

Question: Guaranteed Output Delivery

Does there exist a two round MPC protocol secure against $t < N/2$ fail-stop corruptions in the plain model?

Question: Guaranteed Output Delivery

Does there exist a two round MPC protocol secure against $t < N/2$ fail-stop corruptions in the plain model?

Does there exist a three round MPC protocol secure against $t < N/2$ malicious corruptions in the plain model?

Question: Guaranteed Output Delivery

Does there exist a two round MPC protocol secure against $t < N/2$ fail-stop corruptions in the plain model?

Does there exist a three round MPC protocol secure against $t < N/2$ malicious corruptions in the plain model?

Both questions open regardless of assumptions.

Our Results: Security with Abort

Two round MPC for general functionalities in the plain model, assuming one-way functions.

Our Results: Guaranteed Output Delivery

Fail-Stop Corruptions: Two round MPC for general functions:

Broadcast channel protocol in the bare-public-key model, assuming PKE.

Point-to-point channel protocol in the plain model, assuming OT.

Our Results: Guaranteed Output Delivery

Fail-Stop Corruptions:

Broadcast channel protocol in the bare-public-key model, assuming PKE.

g OT.

Three round MPC from one-way functions in the plain model.

Our Results: Guaranteed Output Delivery

Fail-Stop Corruptions: Two round MPC for general functions:

Broadcast channel protocol in the bare-public-key model, assuming PKE.

Point-to-point channel protocol in the plain model, assuming OT.

Malicious Corruptions: Three round MPC for general functions:

Broadcast channel protocol in the plain model, assuming Zaps and PKE.

Security with Abort against Malicious Adversaries

[Garg-Srinivasan17]

A compiler from any polynomial round MPC protocol to a two round protocol using two round UC secure OT.

[Garg-Srinivasan17]

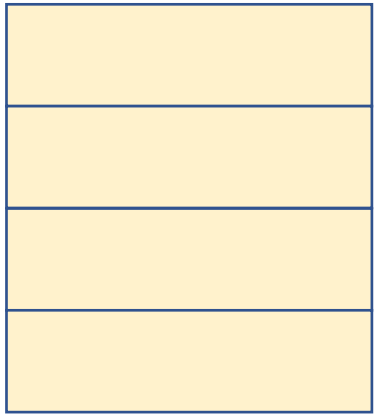
A compiler from any polynomial round MPC protocol to a two round protocol using two round UC secure OT.

Starting Idea: Leverage honest majority to remove OT.

[Garg-Srinivasan17]

Use of OT in [GS17]

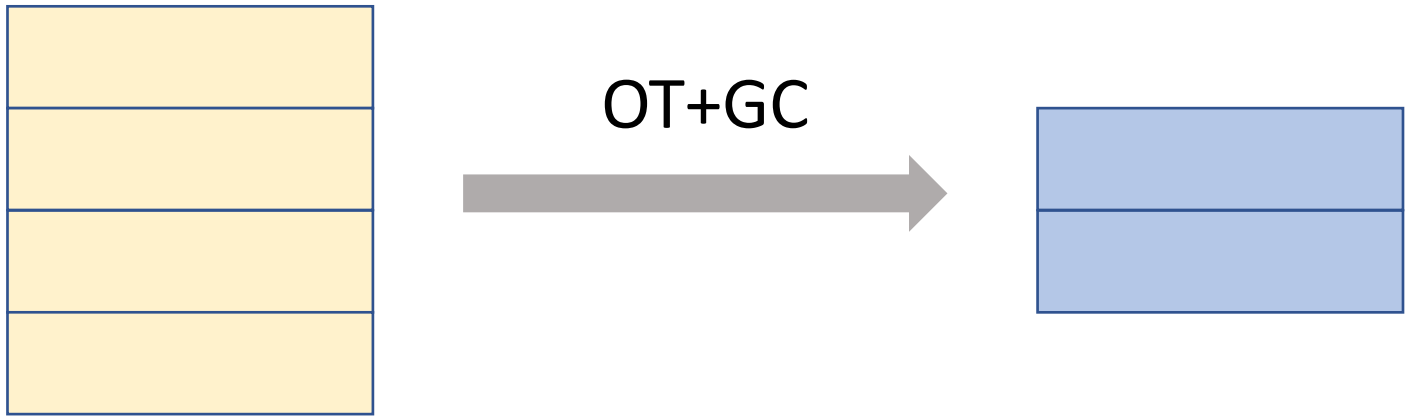
[Garg-Srinivasan17]



Any polynomial round MPC Protocol

Use of OT in [GS17]
Start with any dishonest majority protocol based on OT over broadcast channels

[Garg-Srinivasan17]



Any polynomial round MPC Protocol

Two-round MPC Protocol

Use of OT in [GS17]
Start with any dishonest majority protocol based on OT over broadcast channels
Compile it into a 2 round protocol using OT and Garbled circuits

Our Strategy

	Use of OT in [GS17]	Our approach
1	Start with any dishonest majority protocol based on OT over broadcast channels	
2	Compile it into a 2 round protocol using OT and Garbled circuits	

Our Strategy

	Use of OT in [GS17]	Our approach
1	Start with any dishonest majority protocol based on OT over broadcast channels	Start with an unconditionally secure honest majority protocol
2	Compile it into a 2 round protocol using OT and Garbled circuits	

Our Strategy

	Use of OT in [GS17]	Our approach
1	Start with any dishonest majority protocol based on OT over broadcast channels	Start with an unconditionally secure honest majority protocol Require private channels
2	Compile it into a 2 round protocol using OT and Garbled circuits	

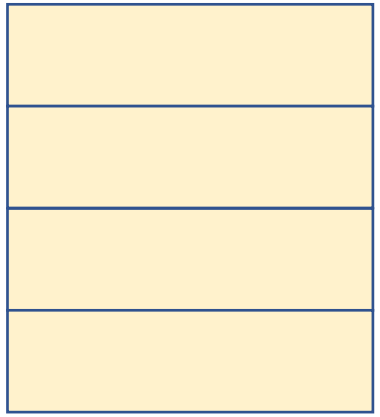
Our Strategy

	Use of OT in [GS17]	Our approach	Challenges
1	Start with any dishonest majority protocol based on OT over broadcast channels	Start with an unconditionally secure honest majority protocol Require private channels	How to compress protocols that use private channels?
2	Compile it into a 2 round protocol using OT and Garbled circuits		

Our Strategy

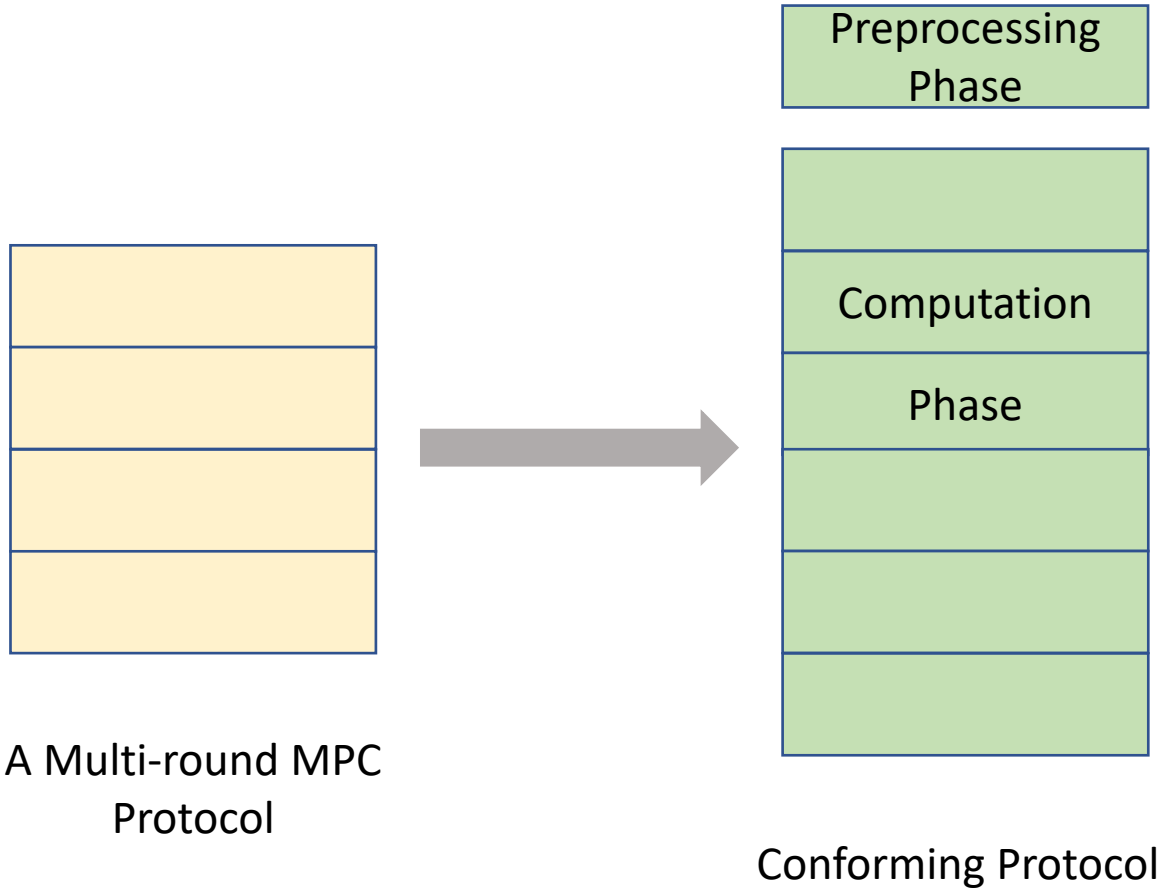
	Use of OT in [GS17]	Our approach	Challenges
1	Start with any dishonest majority protocol based on OT over broadcast channels	Start with an unconditionally secure honest majority protocol Require private channels	How to compress protocols that use private channels?
2	Compile it into a 2 round protocol using OT and Garbled circuits	Leverage honest majority to replace OT	How to achieve OT functionality without OT?

Recap of [Garg-Srinivasan17]



A Multi-round MPC Protocol

Recap of [Garg-Srinivasan17]

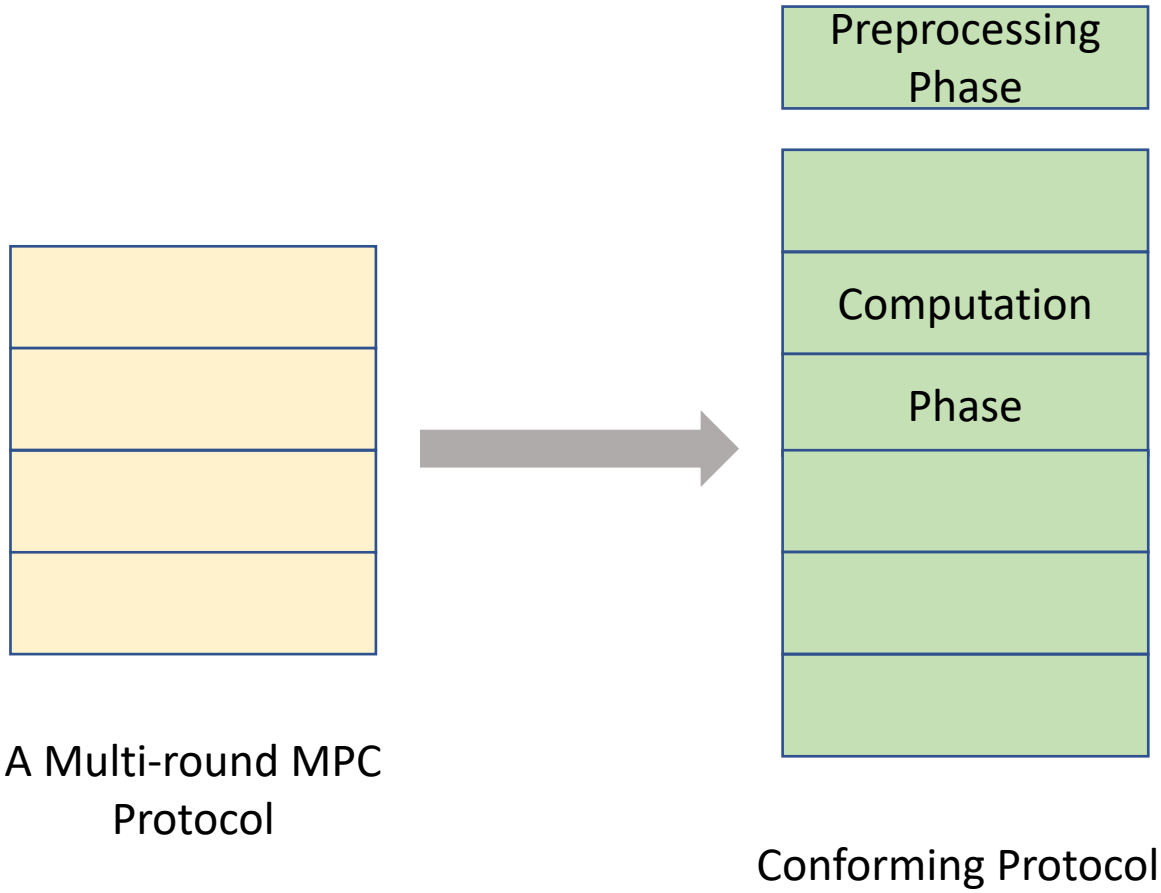


A Multi-round MPC Protocol

Conforming Protocol

Transform into a "conforming protocol" with a specific syntactic structure.

Recap of [Garg-Srinivasan17]



A Multi-round MPC Protocol

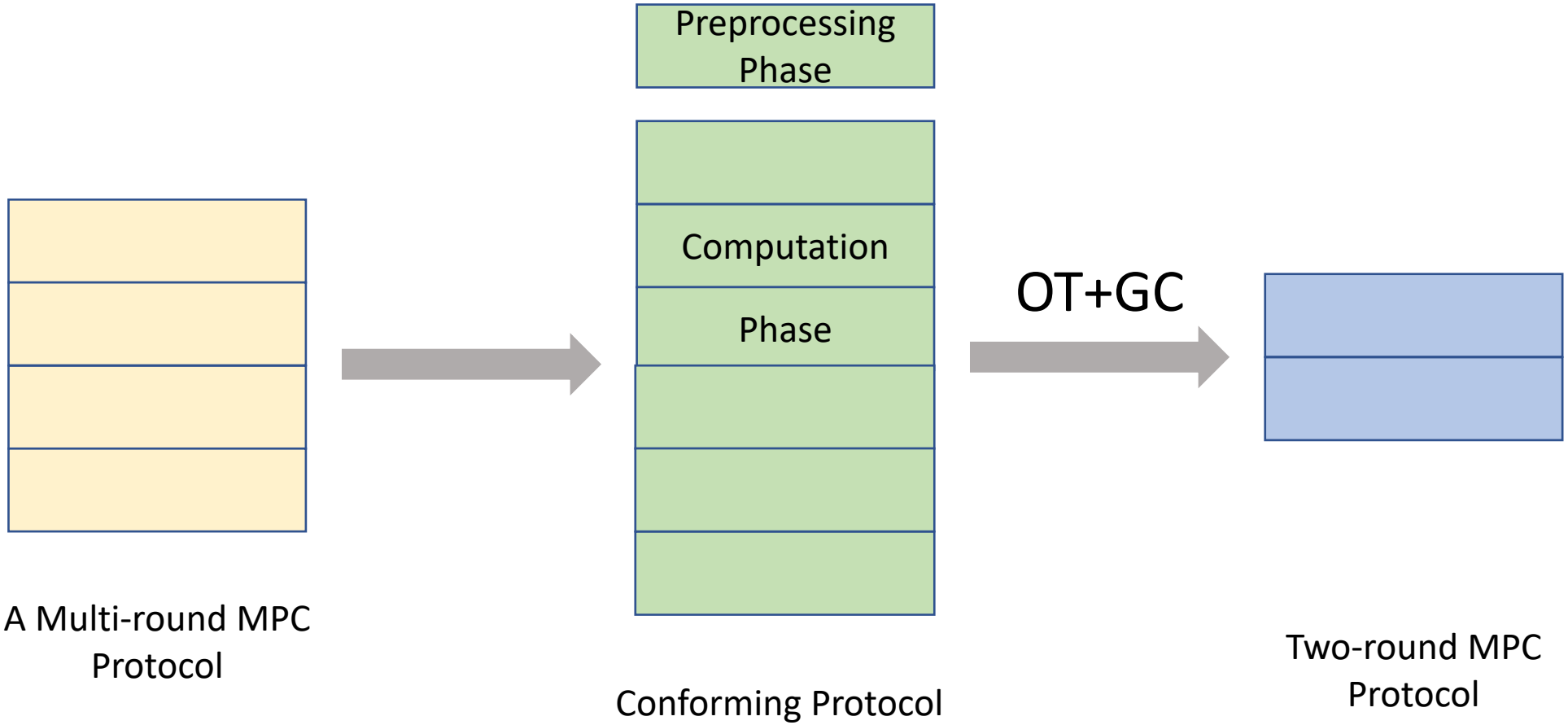
Conforming Protocol

Computation Phase:

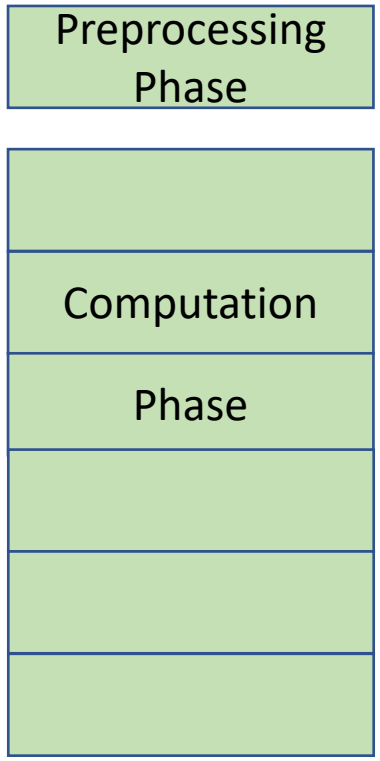
Only a single bit is broadcasted by a single party (speaker) in each round.

All other parties are listeners for that round.

Recap of [Garg-Srinivasan17]



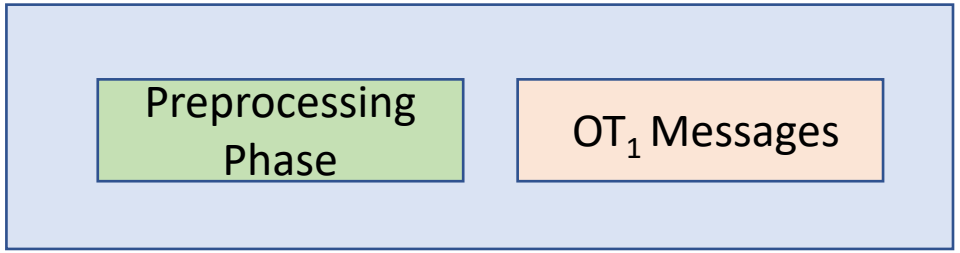
Recap of [Garg-Srinivasan17]



Conforming Protocol

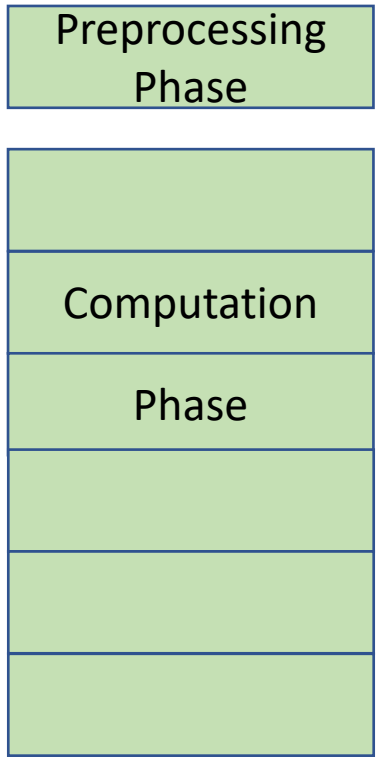
Two-round UC secure OT + Garbled Circuits

Round 1



- Each party sends OT receiver messages for the rounds in which it speaks.
- These messages commit to all its actions in the computation phase of the conforming protocol.

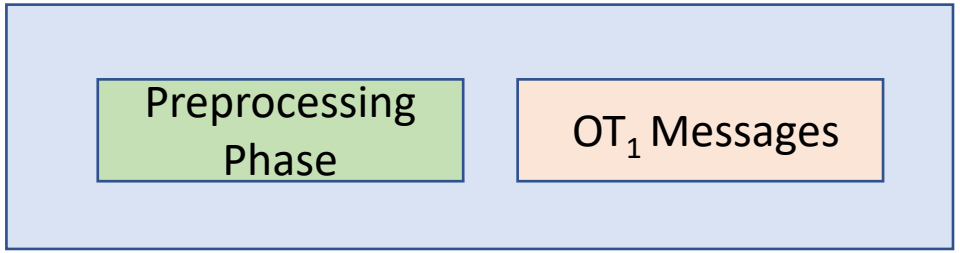
Recap of [Garg-Srinivasan17]



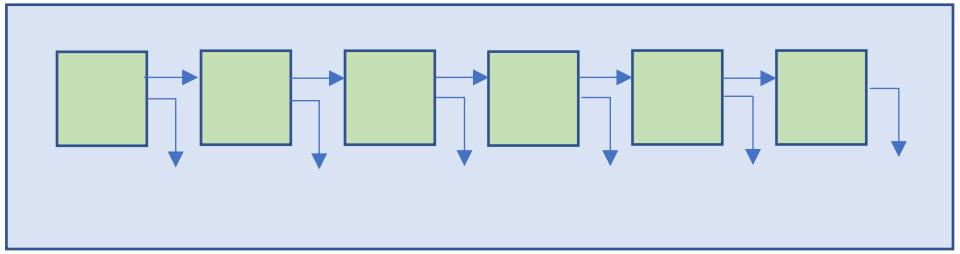
Conforming Protocol

Two-round UC secure OT
+
Garbled Circuits

Round 1

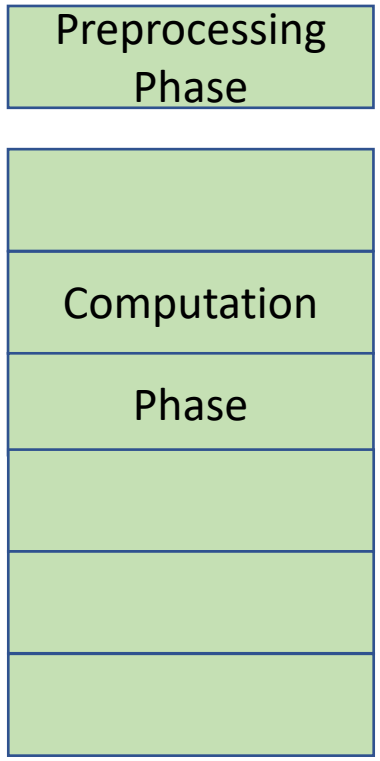


Round 2



Each party sends garbled circuits corresponding to each round in the computation phase.

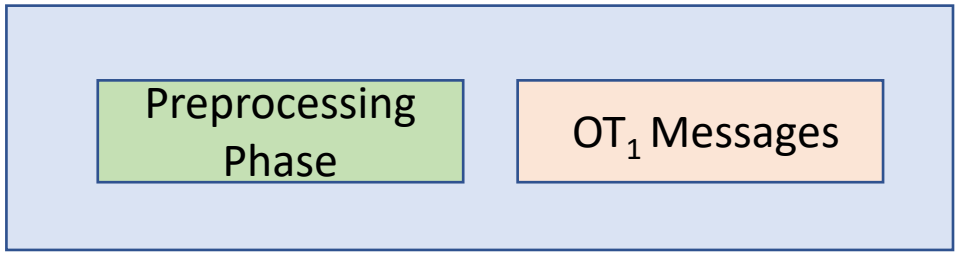
Recap of [Garg-Srinivasan17]



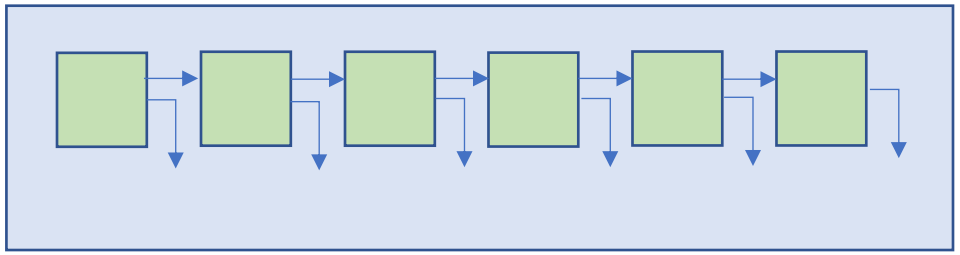
Conforming Protocol

Two-round UC secure OT + Garbled Circuits

Round 1



Round 2



GCs output the OT sender messages.

Goal of these OTs is to deliver wire labels of GC.

Our Strategy: Challenge 2

	Use of OT in [GS17]	Our approach	Challenges
1	Start with any dishonest majority protocol based on OT over broadcast channels	Start with an unconditionally secure honest majority protocol Require private channels	How to compress protocols that use private channels?
2	Compile it into a 2 round protocol using OT and Garbled circuits	Leverage honest majority to replace OT	How to achieve OT functionality without OT?

New Gadget for OT: Multi-party OT

Multi-party protocol.

New Gadget for OT: Multi-party OT

Multi-party protocol.

Only 2 parties have inputs, others have no input.

New Gadget for OT: Multi-party OT

Multi-party protocol.

Only 2 parties have inputs, others have no input.

Every party receives the output.

New Gadget for OT: Multi-party OT

Multi-party protocol.

Only 2 parties have inputs, others have no input.

Every party receives the output.

OT functionality for sender inputs (m_0, m_1) and receiver input (b) can be represented as a degree 2 polynomial in $\mathbb{F}_2$.

$$m_b = m_0(1 + b) + m_1(b)$$

New Gadget for OT: Multi-party OT

Multi-party protocol.

Only 2 parties have inputs, others have no input.

Every party receives the output.

OT functionality for sender inputs (m_0, m_1) and receiver input (b) can be represented as a degree 2 polynomial in $\mathbb{F}_2$.

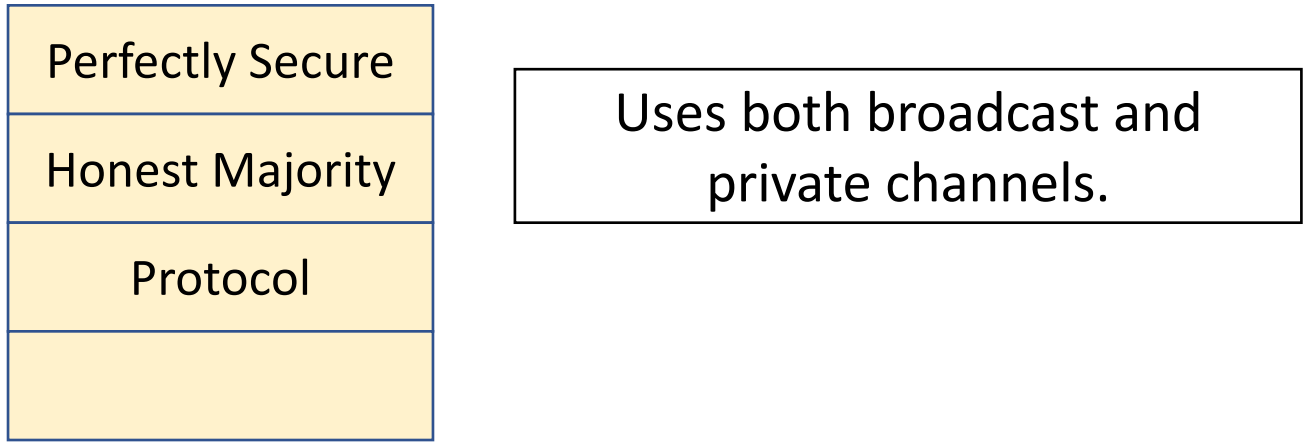
$$m_b = m_0(1 + b) + m_1(b)$$

Later: How to implement

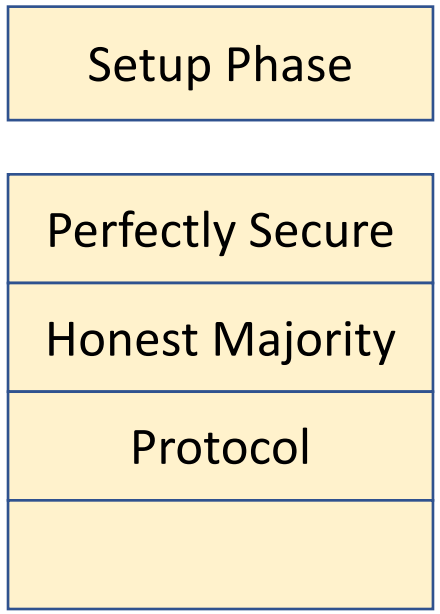
Our Strategy: Challenge 1

	Use of OT in [GS17]	Our approach	Challenges
1	Start with any dishonest majority protocol based on OT over broadcast channels	Start with an unconditionally secure honest majority protocol Require private channels	How to compress protocols that use private channels?
2	Compile it into a 2 round protocol using OT and Garbled circuits	Leverage honest majority to replace OT	How to achieve OT functionality without OT?

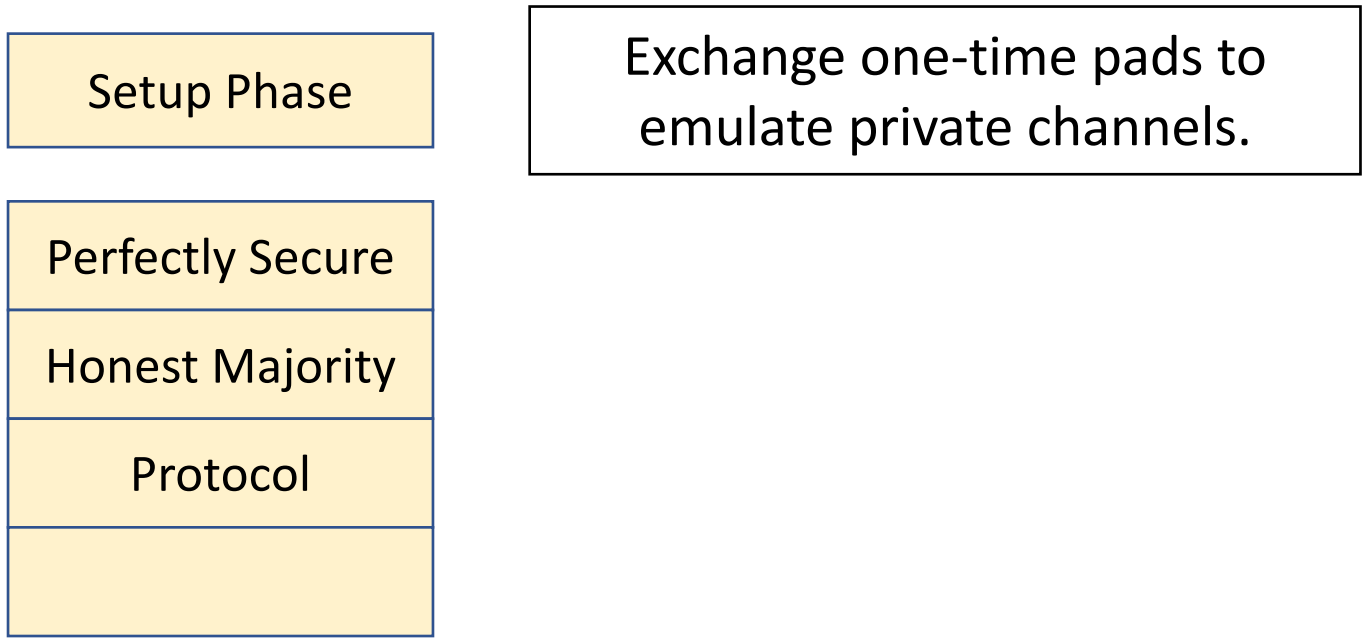
Compressing Private Channel Protocols



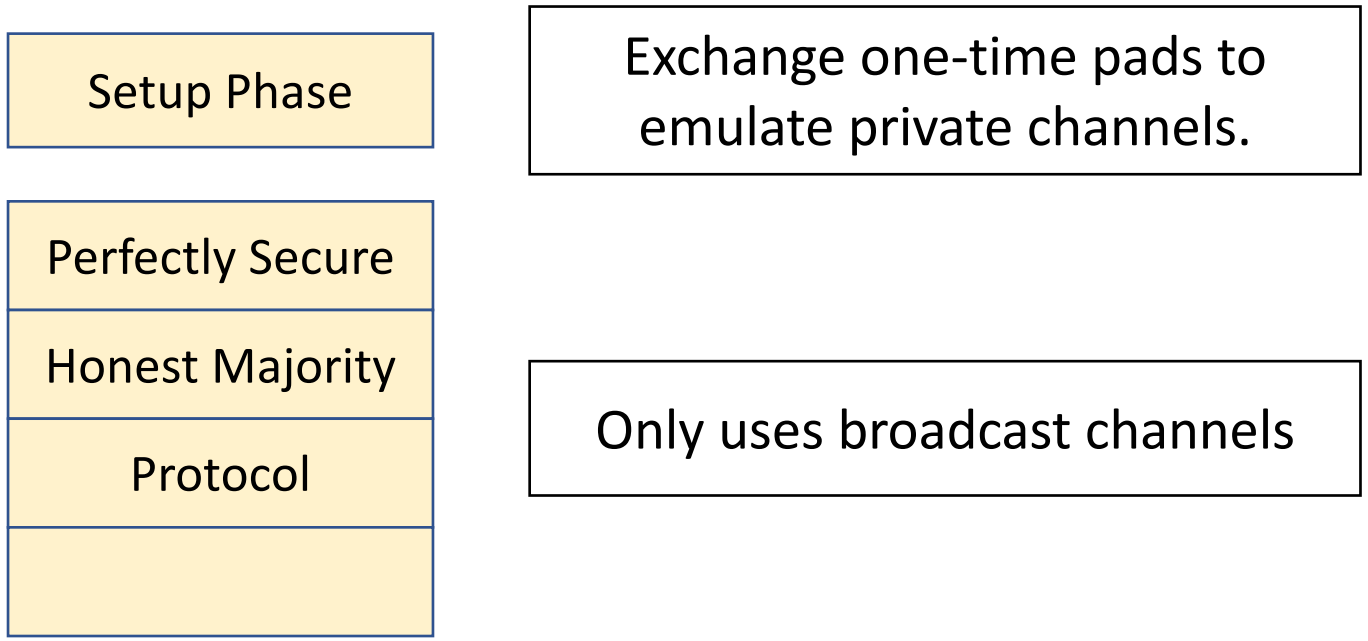
Compressing Private Channel Protocols



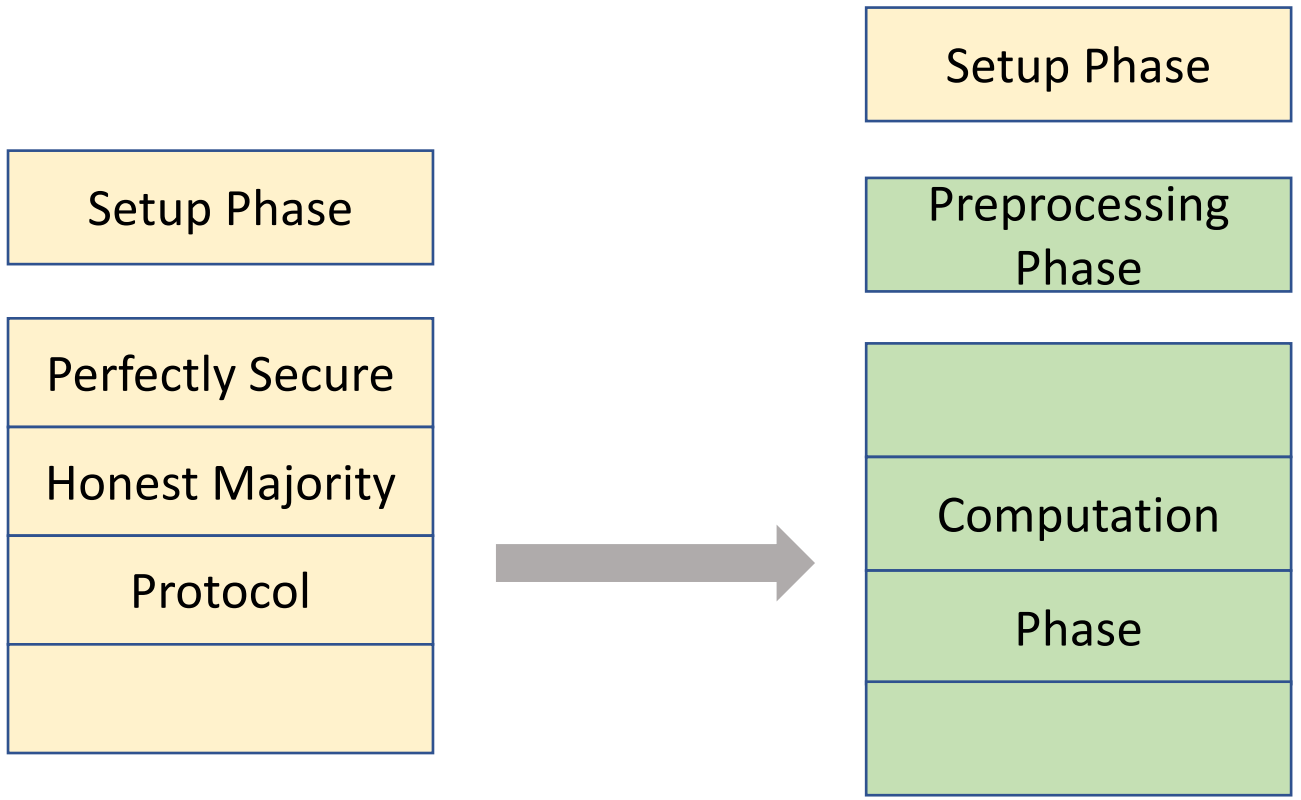
Compressing Private Channel Protocols



Compressing Private Channel Protocols



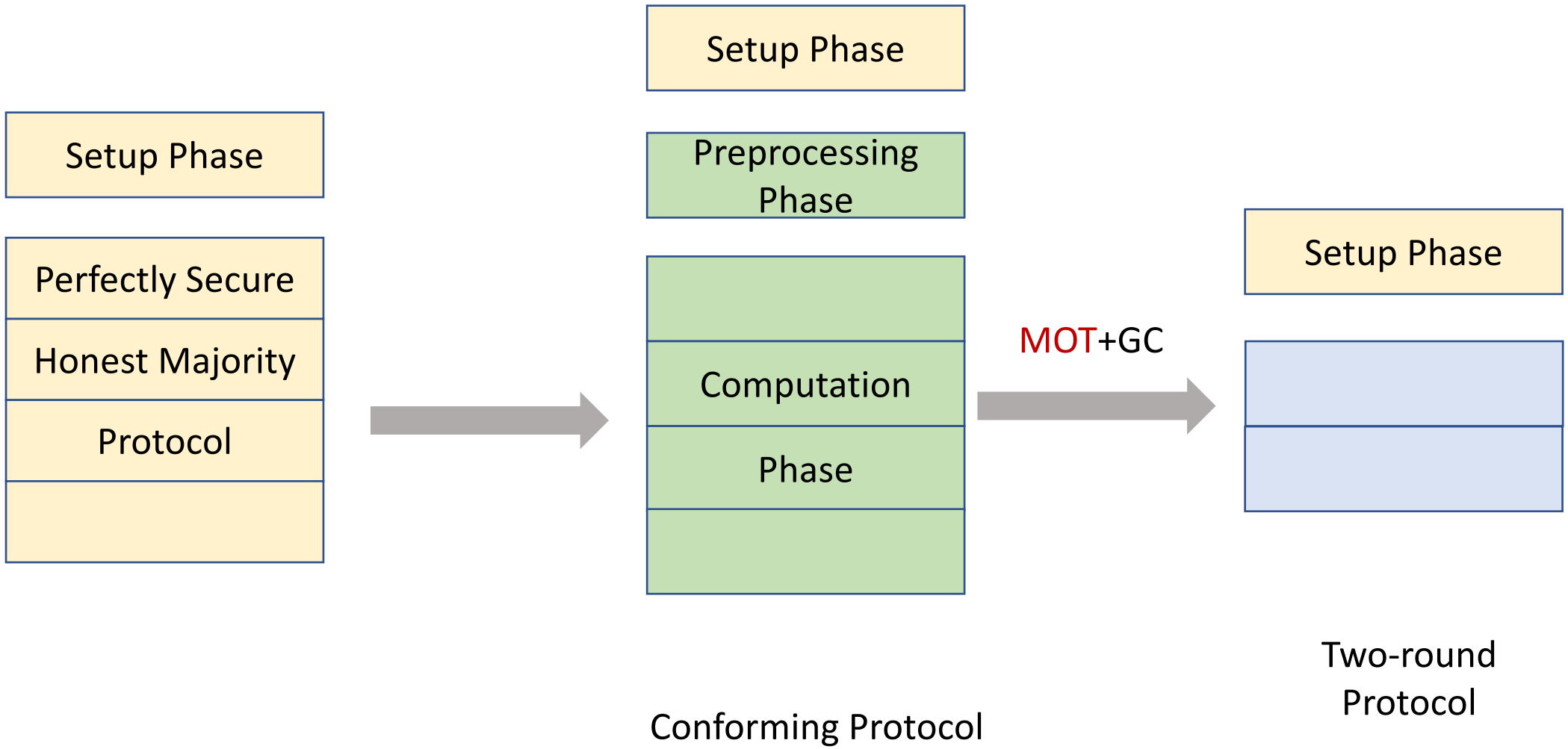
Compressing Private Channel Protocols



Transform to a conforming protocol with a setup phase

Conforming Protocol

Compressing Private Channel Protocols



Compressing Private Channel Protocols

Can we parallelize the first round with the setup phase?

Can we parallelize the first round with the setup phase?

Conforming Protocol with setup

Setup Phase

Listener of round t

R

Speaker of round t

Can we parallelize the first round with the setup phase?

Conforming Protocol with setup

Setup Phase

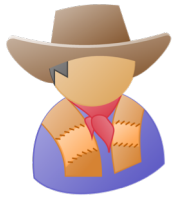
Listener of round t

R

Speaker of round t

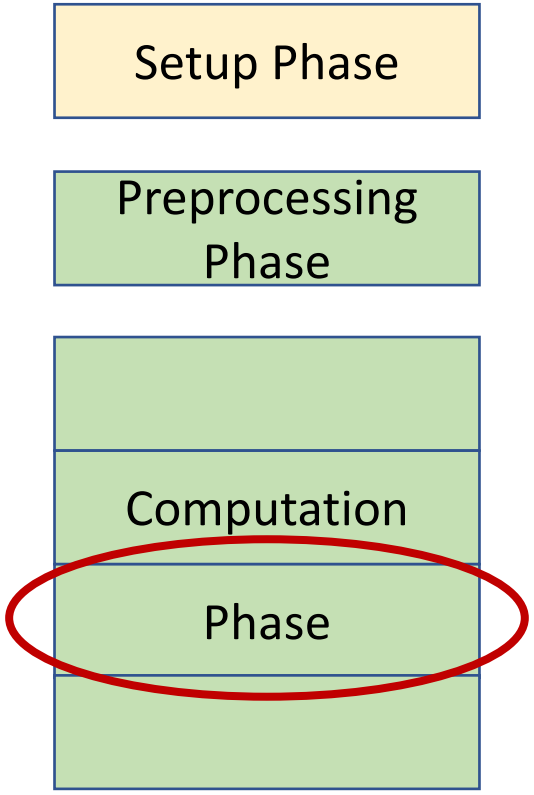
Round t

(computation phase)



Speaker of round t

$Broadcasts\ (m \oplus R)$



Can we parallelize the first round with the setup phase?

2 Round Protocol with setup

Setup Phase

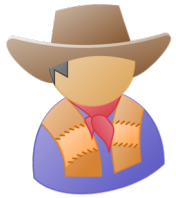
Listener of round t

R

Speaker of round t

Setup Phase

Round 1



Speaker of round t

$Broadcasts\ OT_1\ messages$

Round 1

Round 2

Can we parallelize the first round with the setup phase?

2 Round Protocol with setup

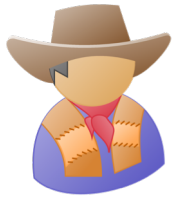
Setup Phase

Listener of round t

R

Speaker of round t

Round 1



Speaker of round t

$Broadcasts\ OT_1\ messages$

OT_1 messages commit to all actions in the first round.

Can we parallelize the first round with the setup phase?

2 Round Protocol with setup

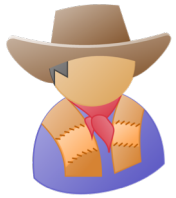
Setup Phase

Listener of round t

R

Speaker of round t

Round 1



Speaker of round t

$Broadcasts\ OT_1\ messages$

OT_1 messages depend on R which is not known before setup.

Can we parallelize the first round with the setup phase?

2 Round Protocol with setup

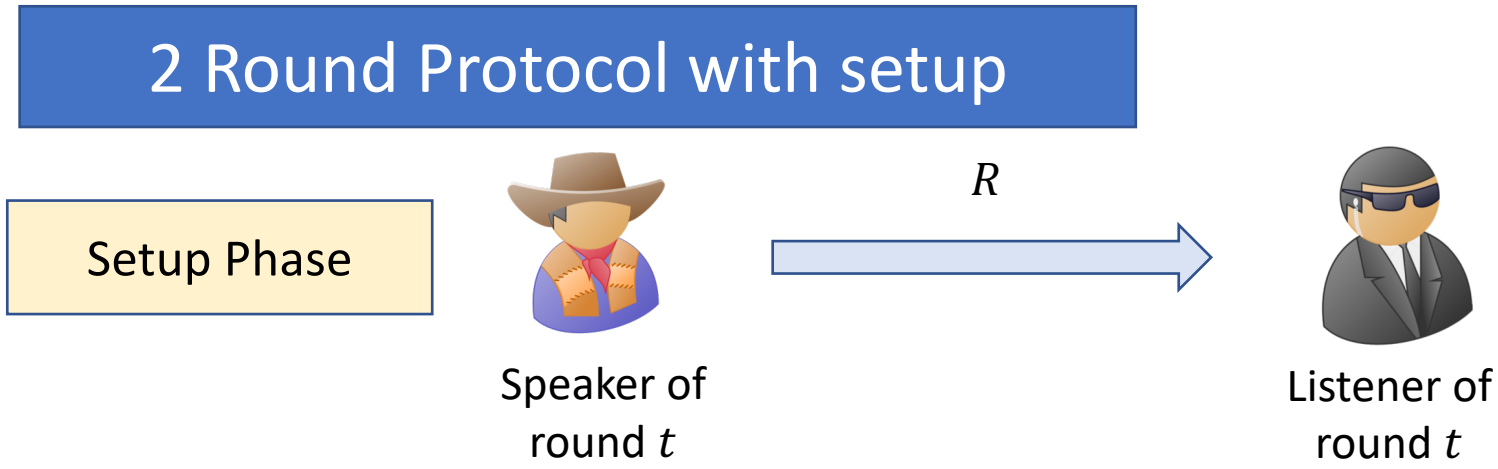
Setup Phase

Speaker of round t

R

Listener of round t

Can we parallelize the first round with the setup phase?



- Similar problem arises.
- Transfers the problem to another round.

Can we parallelize the first round with the setup phase?

2 Round Protocol with setup

Setup Phase

R

Speaker of round 2

Listener of round 2

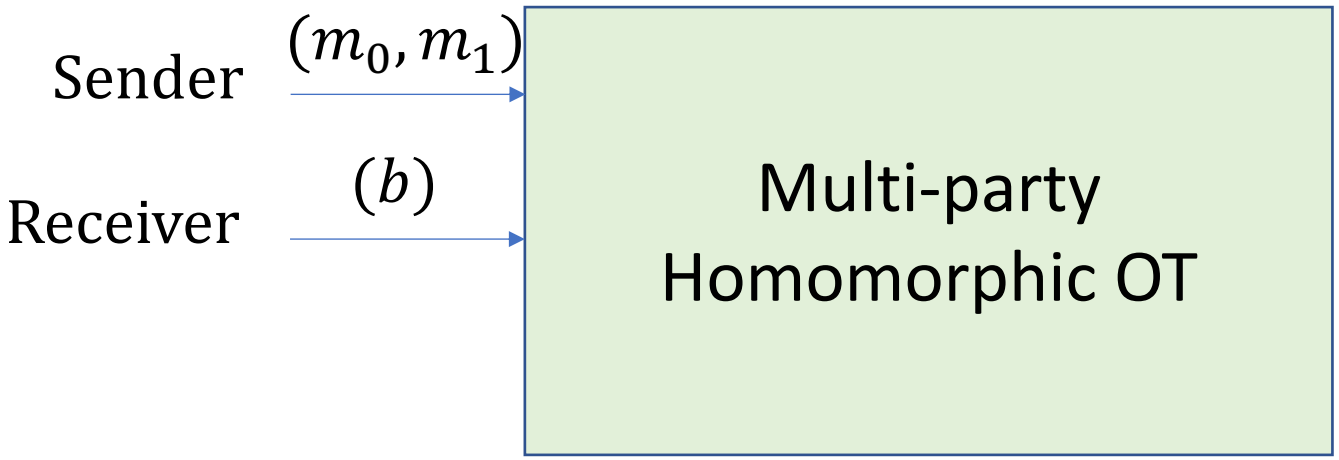
This approach doesn't seem to work!

- Similar problem arises.
- Transfers the problem to another round.

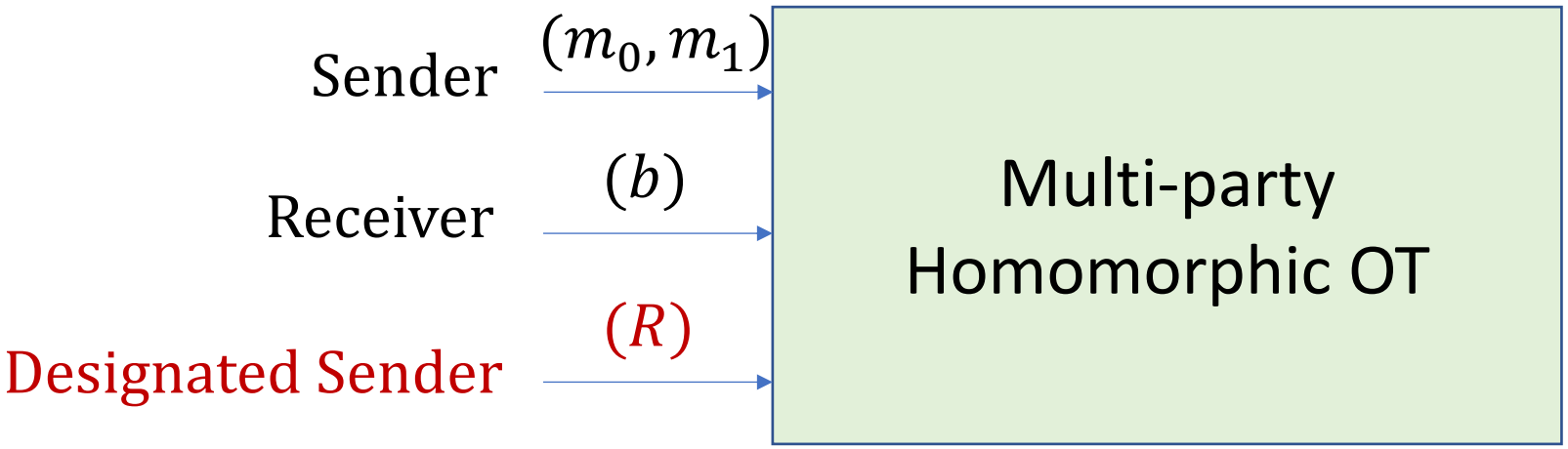
Multi-party Homomorphic OT

- Multi-party protocol.
- Only 3 parties have inputs, others have no input.
- Every party receives the output.

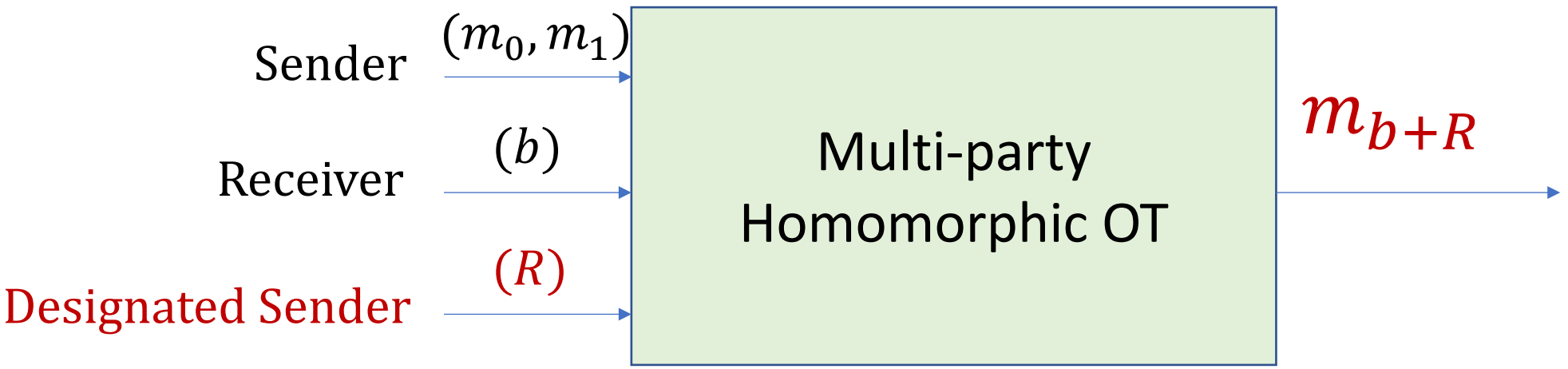
Multi-party Homomorphic OT



Multi-party Homomorphic OT



Multi-party Homomorphic OT



Multi-party Homomorphic OT

- The homomorphic OT functionality with sender inputs (m_0, m_1), receiver input (b) and designated sender input (R) can be represented as degree 2 polynomial in $\mathbb{F}_2$.

$$m_{b+R} = m_0(1 + b + R) + m_1(b + R)$$

Parallelizing using MHOT

2 Round Protocol with setup

Listener of round t

R

Speaker of round t

Setup Phase

Speaker of round t

$Broadcasts\ OT_1\ messages$

Round 1

Parallelizing using MHOT

2 Round Protocol with setup

Listener of round t

R

Speaker of round t

Setup Phase

Speaker of round t

$Broadcasts\ OT_1\ messages$

Round 1

Listener of round t

$Broadcasts\ OT_1\ message\ using\ input\ R$

Parallelizing using MHOT

2 Round Protocol with setup parallelized

Listener of round t

R

Speaker of round t

Setup Phase

Round 1

Speaker of round t

$Broadcasts\ OT_1\ messages$

Listener of round t

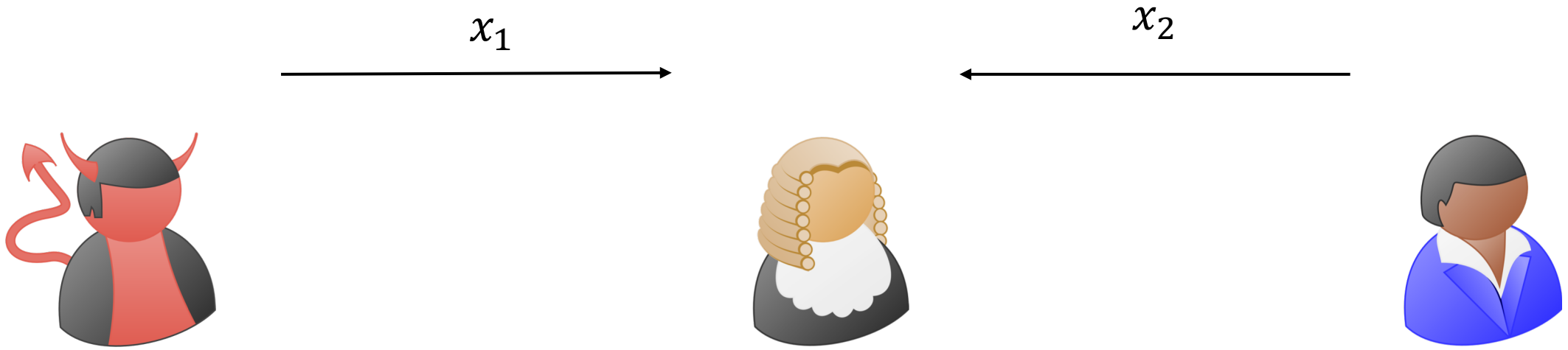
$Broadcasts\ OT_1\ message\ using\ input\ R$

The homomorphism property of the multi-party OT allows us to parallelize

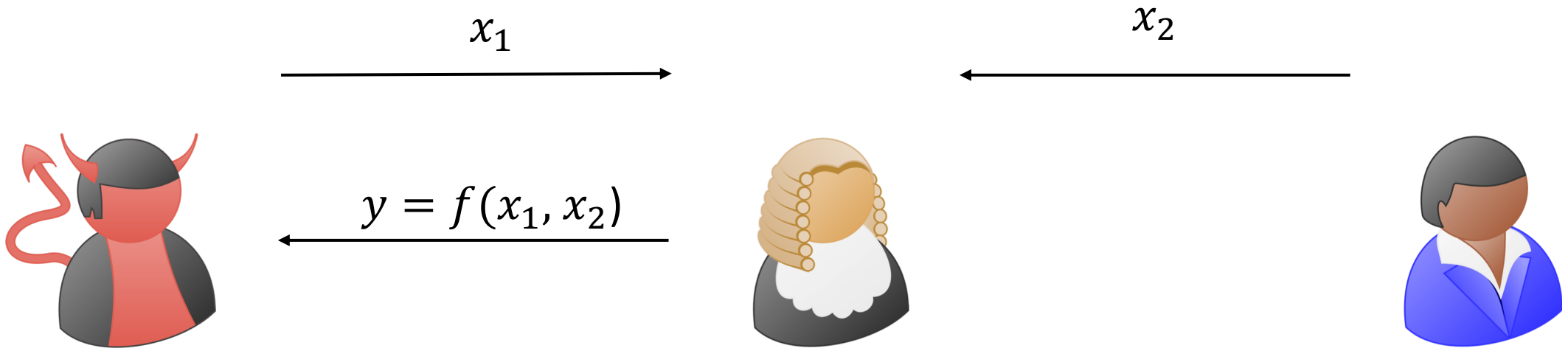
Instantiating Multi-party Homomorphic OT

- [Ishai-Kushilevitz-Paskin10] give a construction for such a degree 2 polynomial computation protocol that satisfies statistical t-privacy with knowledge of outputs.

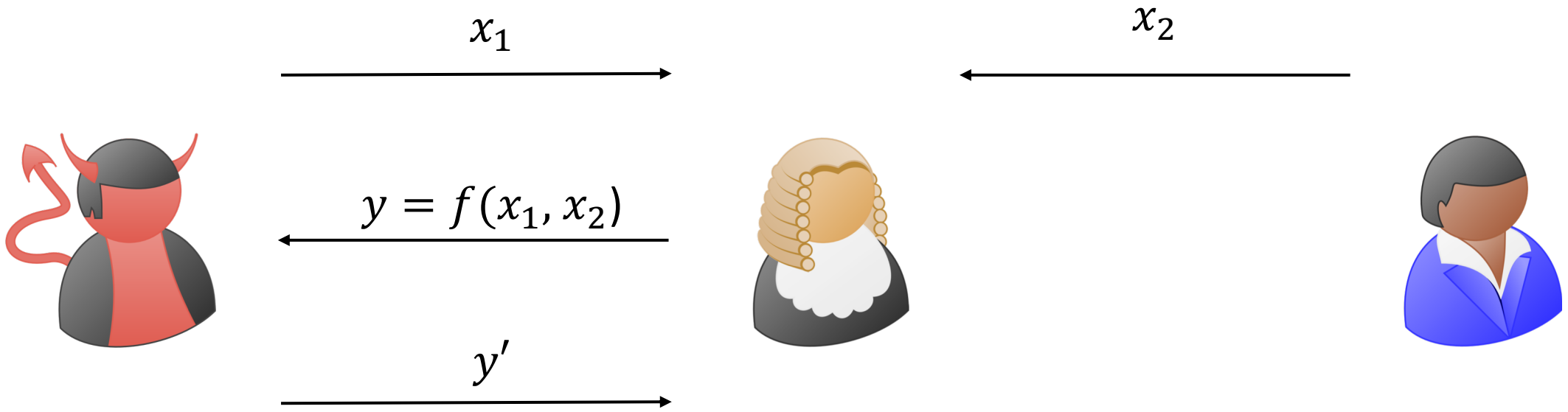
Ideal World: Privacy with Knowledge of Outputs



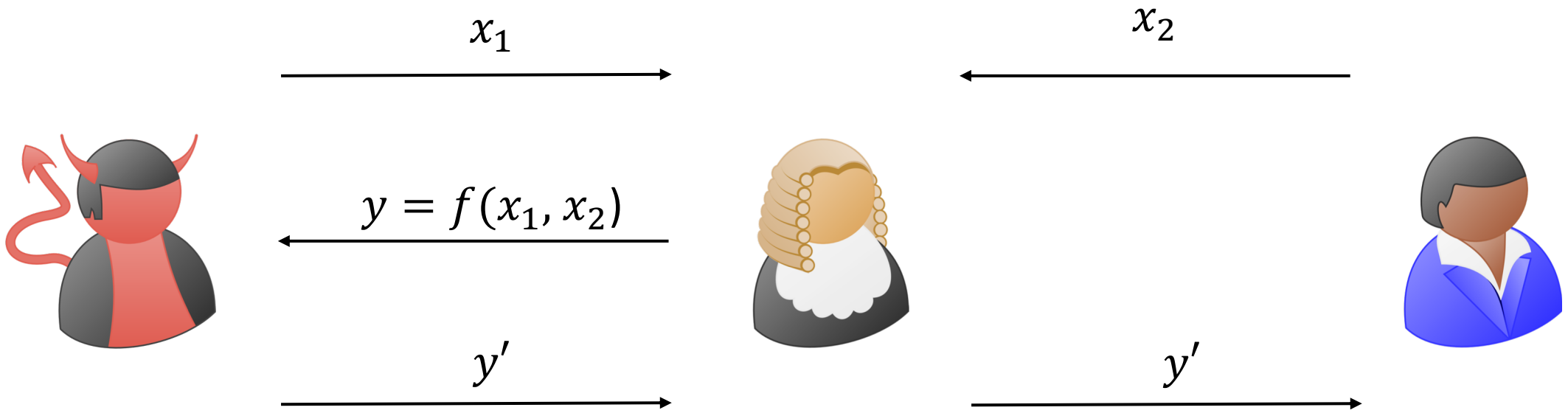
Ideal World: Privacy with Knowledge of Outputs



Ideal World: Privacy with Knowledge of Outputs



Ideal World: Privacy with Knowledge of Outputs



Instantiating Multi-party Homomorphic OT

- [Ishai-Kushilevitz-Paskin10] give a construction for such a degree 2 polynomial computation protocol that satisfies statistical t-privacy with knowledge of outputs.

Privacy with knowledge of outputs: A weaker notion than security with abort that does not guarantee correctness of output of the honest parties.

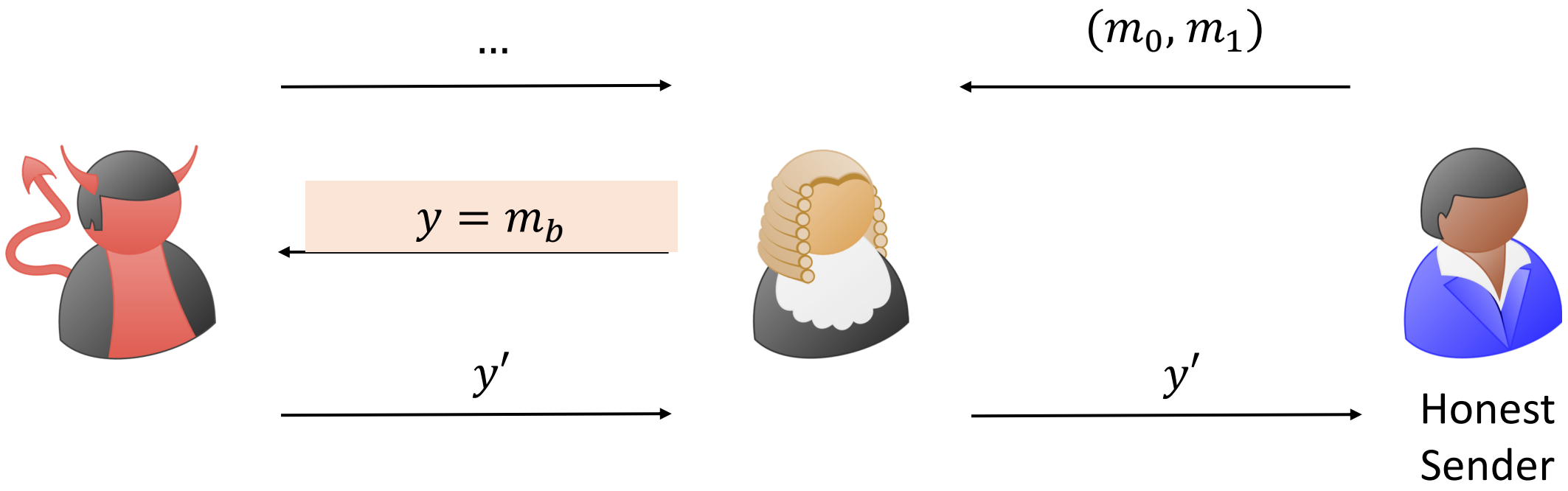
Instantiating Multi-party Homomorphic OT

- [Ishai-Kushilevitz-Paskin10] give a construction for such a degree 2 polynomial computation protocol that satisfies statistical t-privacy with knowledge of outputs.

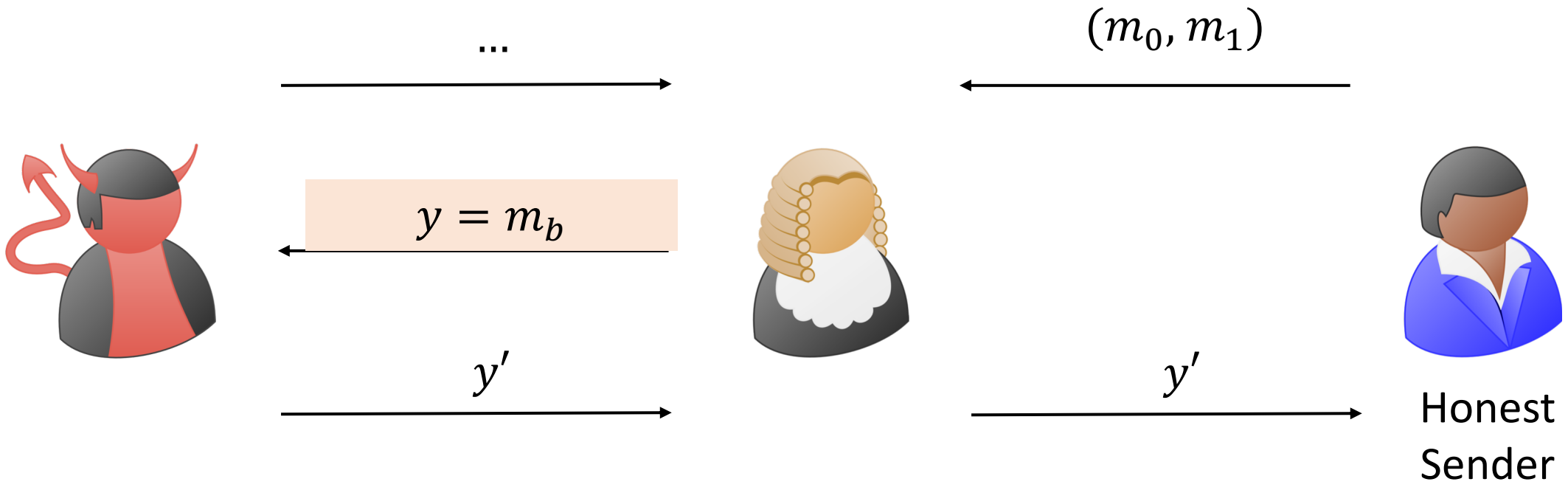
Privacy with knowledge of outputs: A weaker notion than security with abort that does not guarantee correctness of output of the honest parties.

Challenge: How to ensure correctness of honest party outputs?

Challenge: How to ensure correctness of honest party outputs?



Challenge: How to ensure correctness of honest party outputs?



y' does not depend on m_{1-b}

Challenge: How to ensure correctness of honest party outputs?

- OT functionality transmits wire labels for GC.
- Unless valid labels are transmitted, GC remains private.

https://eprint.iacr.org/2018/572

Thank You.

aarushig@cs.jhu.edu