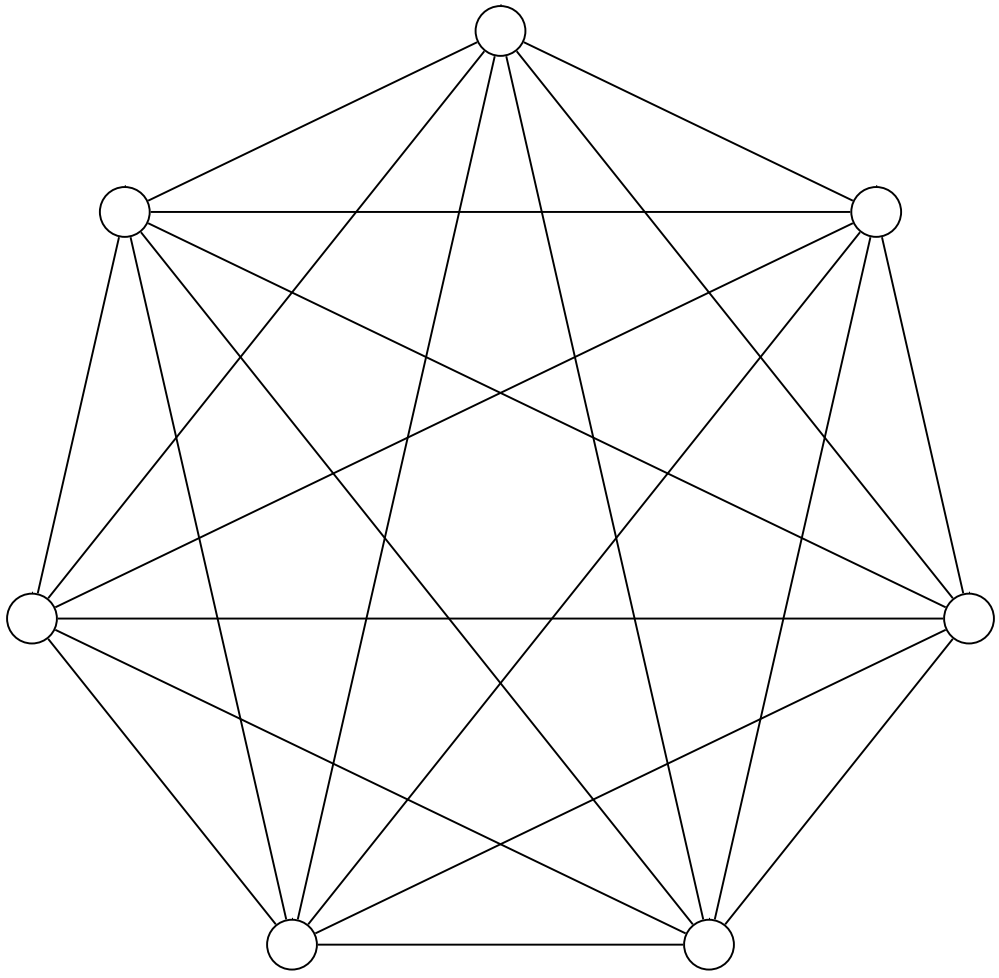
Topology-Hiding Communication from Minimal Assumptions

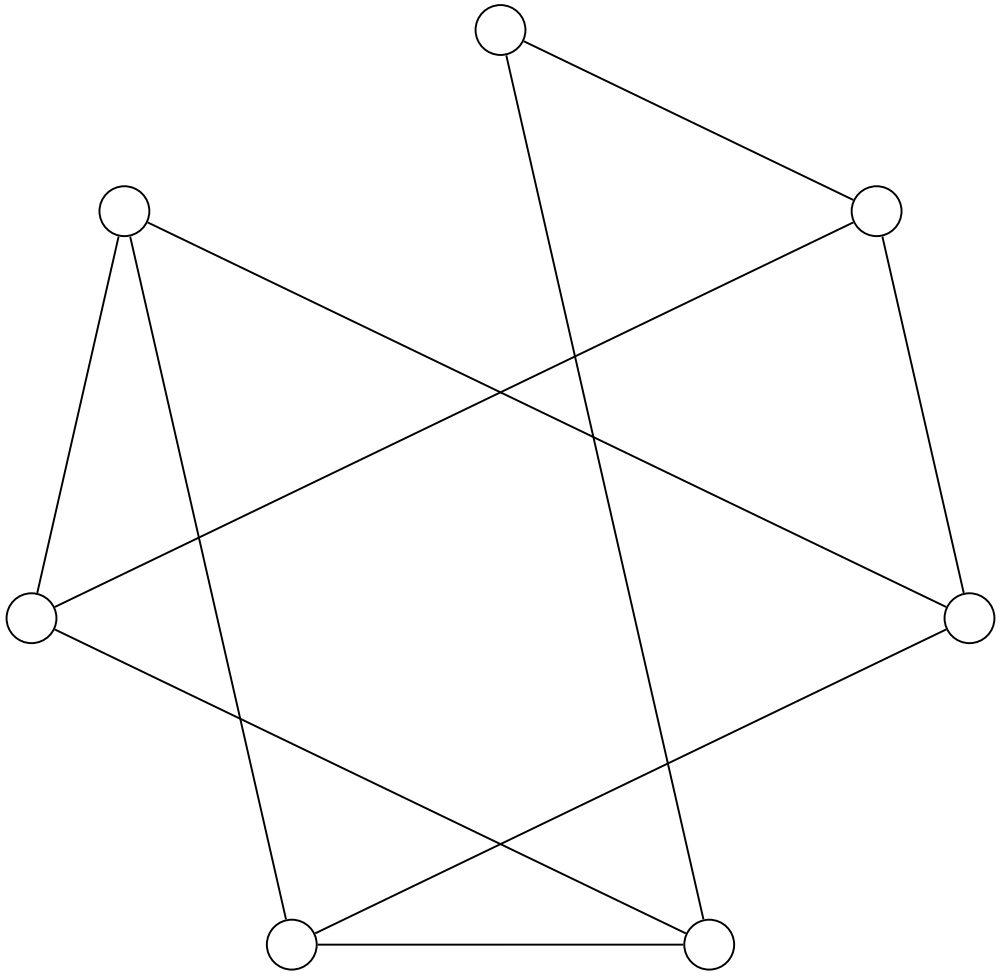
Marshall Ball, Elette Boyle, Ran Cohen, Lisa Kohl,
Tal Malkin, Pierre Meyer, Tal Moran

BUSec Seminar – Feb 3rd 2021

Topology-Hiding Computation



Topology-Hiding Computation



Defining Topology-Hiding Computation

Topology-Hiding Computation:
[Moran-Orlov-Richelson'15]

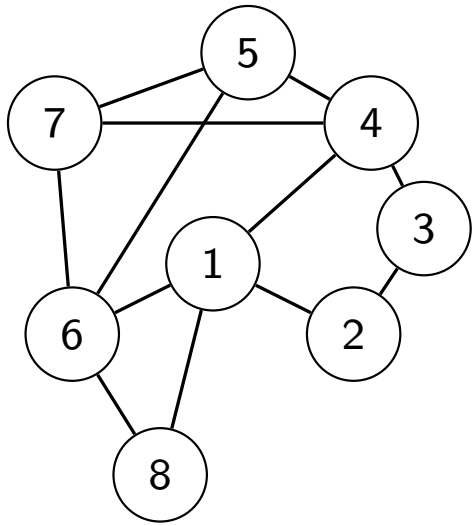
- (MPC over Incomplete Network)
- (Topology-Hiding) Parties can only see their local view
- (Topology-Hiding) Reveals nothing else about the graph

Defining Topology-Hiding Computation

Topology-Hiding Computation:

[Moran-Orlov-Richelson'15]

- (MPC over Incomplete Network)
- (Topology-Hiding) Parties can only see their local view
- (Topology-Hiding) Reveals nothing else about the graph

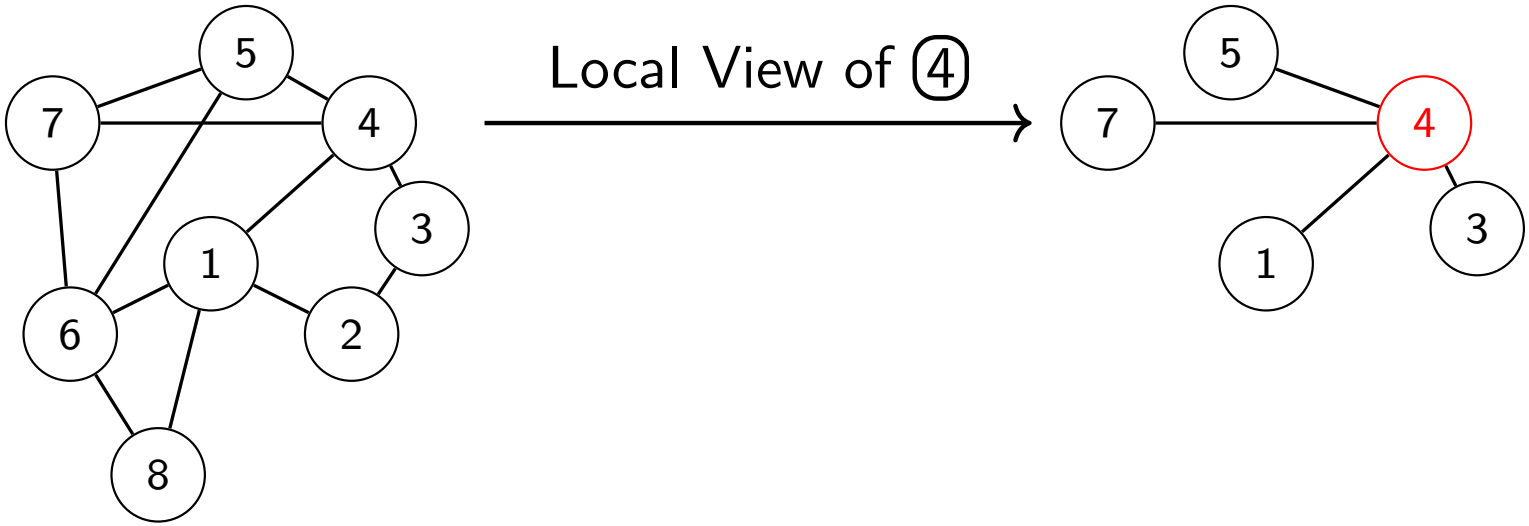


Defining Topology-Hiding Computation

Topology-Hiding Computation:

[Moran-Orlov-Richelson'15]

- (MPC over Incomplete Network)
- (Topology-Hiding) Parties can only see their local view
- (Topology-Hiding) Reveals nothing else about the graph



Defining Topology-Hiding Computation

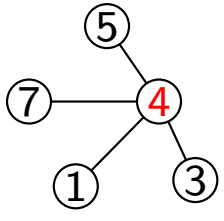
Topology-Hiding Computation:

[Moran-Orlov-Richelson'15]

- (MPC over Incomplete Network)
- (Topology-Hiding) Parties can only see their local view
- (Topology-Hiding) Reveals nothing else about the graph

THC protocol for graph class $\mathcal{G}$:

$\mathsf{Sim}(\mathcal{G}, \mathsf{LocalViewGraph}_4) \quad \simeq \quad \mathsf{ViewProt}_4$



Topology-Hiding Functionalities

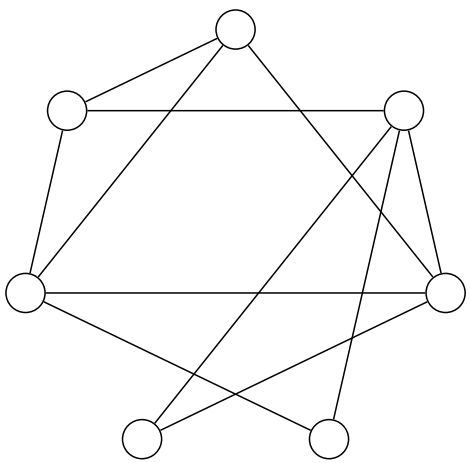
$$\text{Topology-Hiding Computation (THC)} \Rightarrow \text{Topology-Hiding Communication}$$

$$\text{Topology-Hiding Computation (THC)} \overset{?}{\Leftarrow} \text{Topology-Hiding Communication}$$

Topology-Hiding Functionalities

Topology-Hiding Computation (THC) $\Rightarrow$ Topology-Hiding Communication

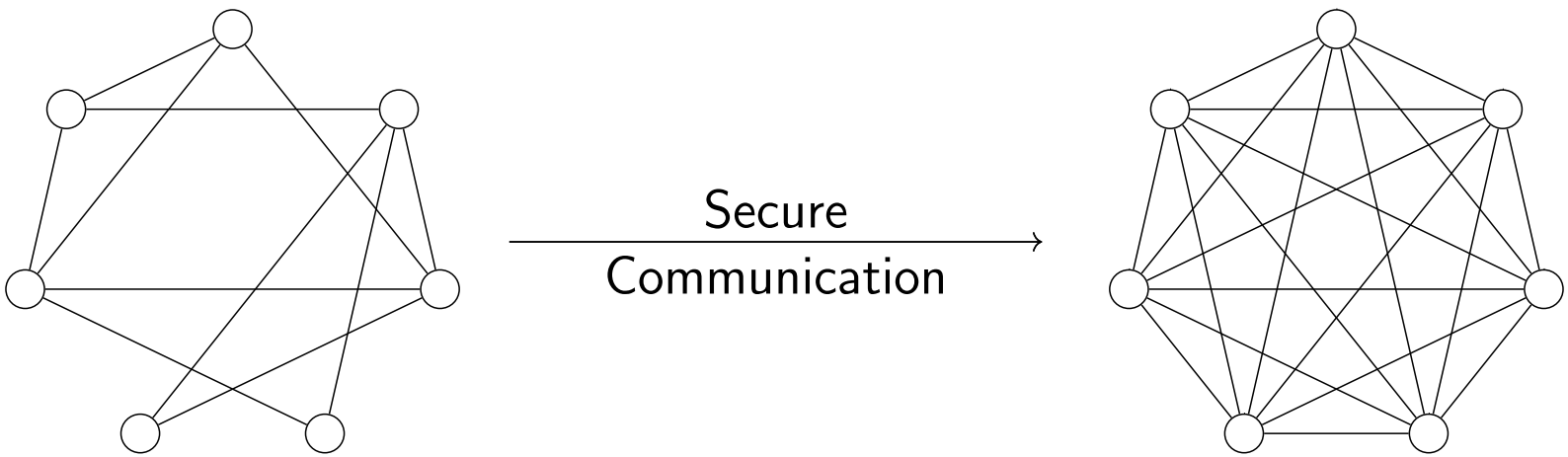
$\overset{?}{\Leftarrow}$



Topology-Hiding Functionalities

Topology-Hiding Computation (THC) $\Rightarrow$ Topology-Hiding Communication

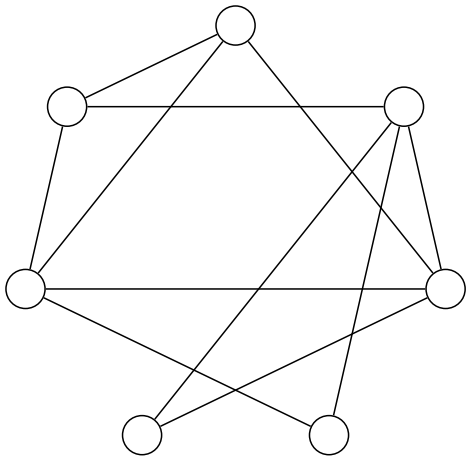
$\Leftarrow^{?}$



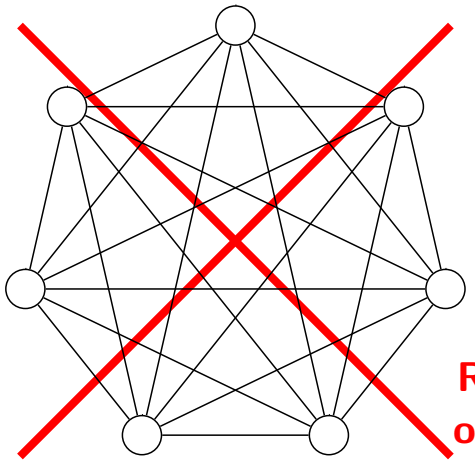
Topology-Hiding Functionalities

Topology-Hiding Computation (THC) $\Rightarrow$ Topology-Hiding Communication

$\overset{?}{\Leftarrow}$



Secure Communication



Reveals nb of vertices!

Topology-Hiding Functionalities

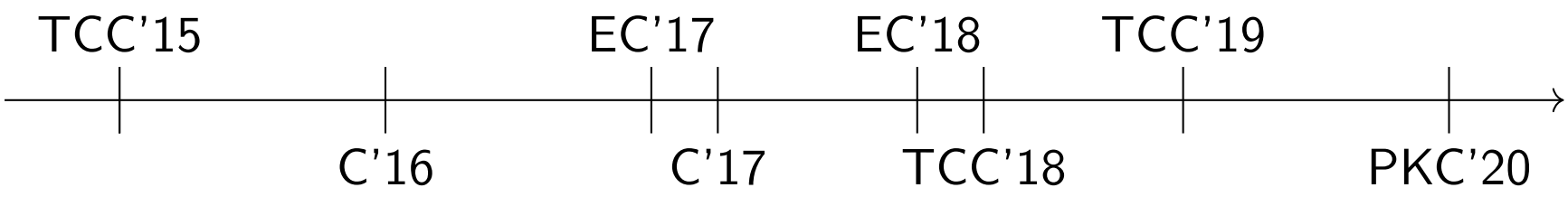
Topology-Hiding Computation (THC) $\Rightarrow$ Topology-Hiding Communication

$\overset{?}{\Leftarrow}$

Topology-Hiding Communication:

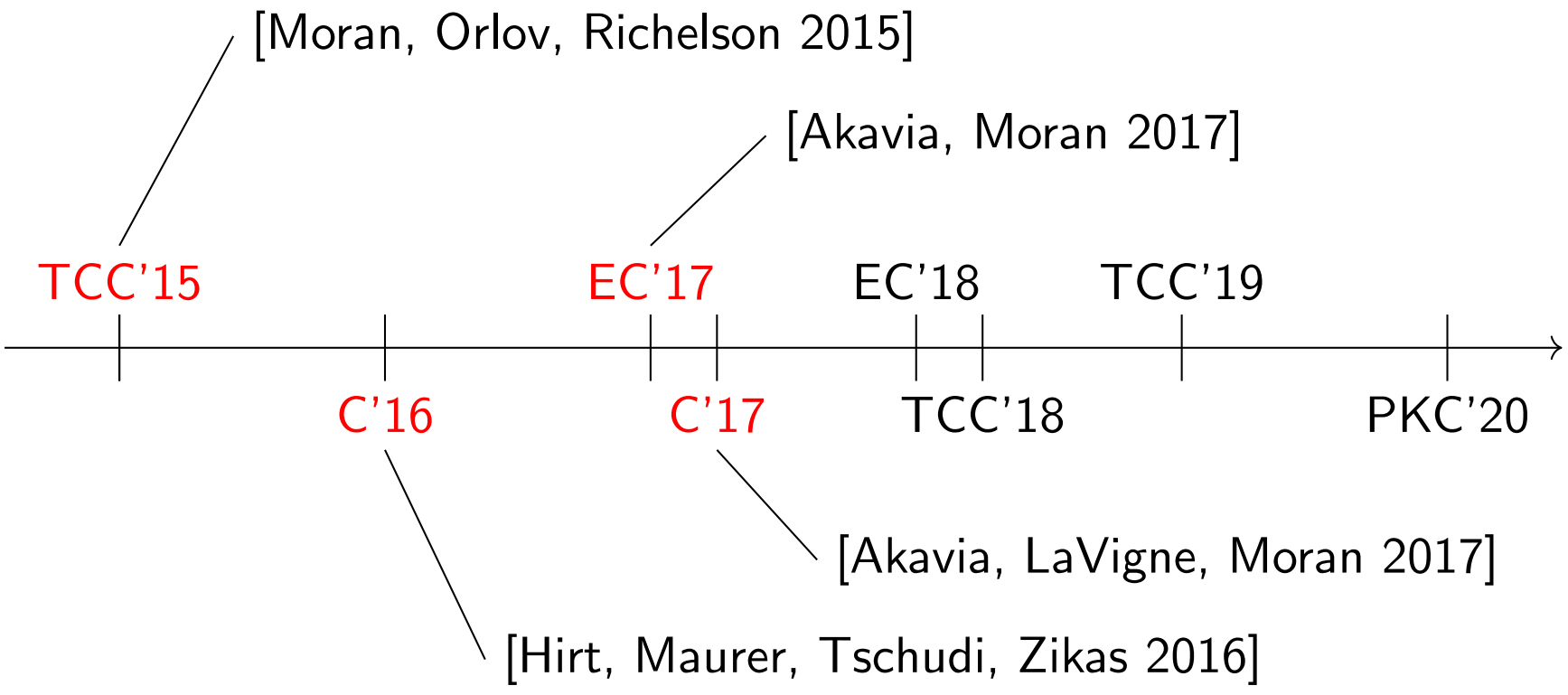
Broadcast (THB)
Anonymous Broadcast (THAB)

Previous Works – (Simplified) Classification



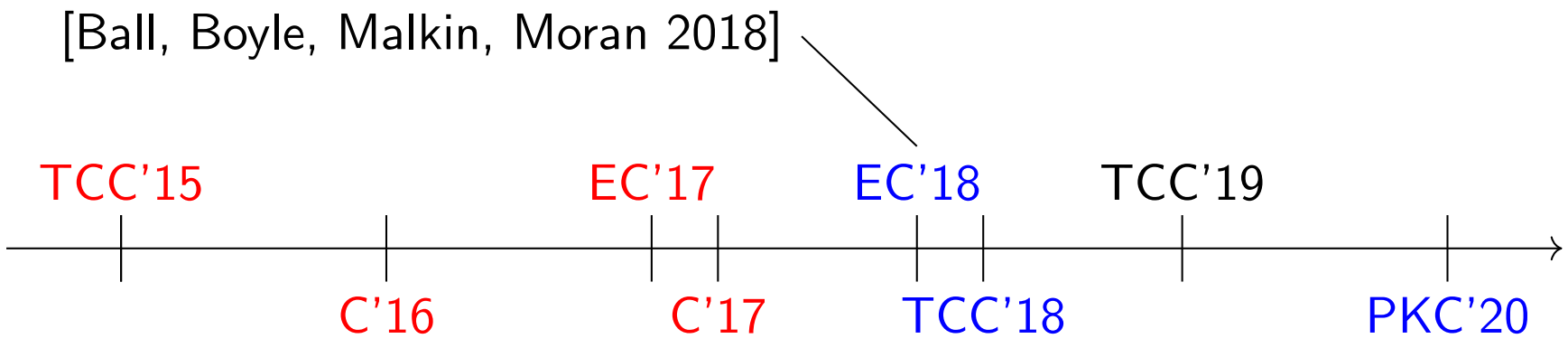
Previous Works – (Simplified) Classification

- Computational (DDH, QR, or LWE), ($t = n - 1$) passive



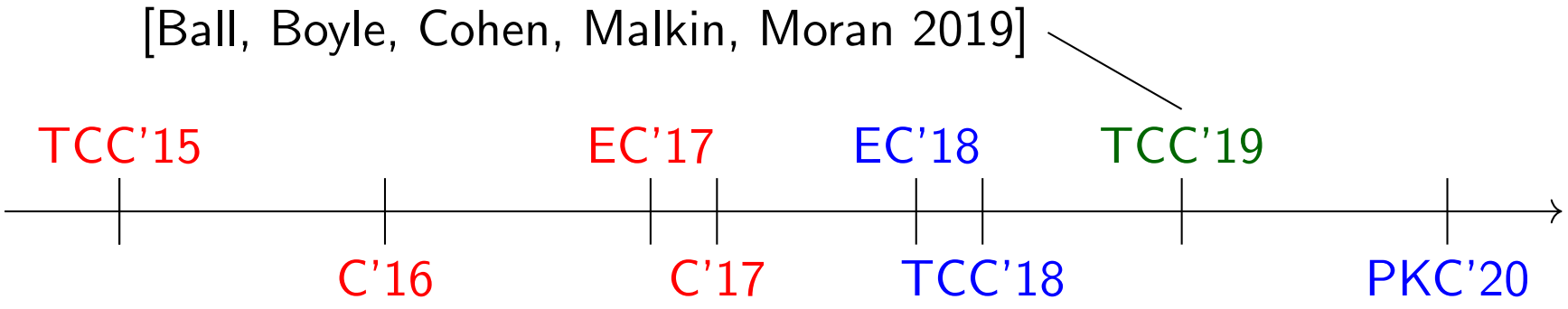
Previous Works – (Simplified) Classification

- Computational (DDH, QR, or LWE), $(t = n - 1)$ passive
- Computational, $(t = n - 1)$ Fail-stop + Asynchronous model



Previous Works – (Simplified) Classification

- Computational (DDH, QR, or LWE), $(t = n - 1)$ passive
- Computational, $(t = n - 1)$ Fail-stop + Asynchronous model
- Revisiting information-theoretic setting, $t = 1$ passive



Previous Works – (Simplified) Classification

- Computational (DDH, QR, or LWE), ($t = n - 1$) passive
- Computational, ($t = n - 1$) Fail-stop + Asynchronous model
- Revisiting information-theoretic setting, $t = 1$ passive

[Ball, Boyle, Cohen, Malkin, Moran 2019]

Is Information-Theoretic Topology-Hiding Computation Possible?
[BBCMM19]

- $\exists$ a class where 1-THB is possible information-theoretically
- $\exists$ a class where 1-THB requires key-agreement

Our Work: The Simplest Setting

- **Broadcast** Only
  (and Anonymous Broadcast)
- **One Semi-Honest** Corruption
- Synchronous Communication

Our Work: The Simplest Setting

- **Broadcast** Only
  (and Anonymous Broadcast)
- **One Semi-Honest** Corruption
- Synchronous Communication

1-THB

1-THAB

Our Work: The Simplest Setting

- **Broadcast** Only
  (and Anonymous Broadcast)
- **One Semi-Honest** Corruption
- Synchronous Communication

1-THB

1-THAB

Trivial without Topology-Hiding

Very Rich with Topology-Hiding!

Our Work: The Simplest Setting

- **Broadcast** Only
  (and Anonymous Broadcast)
- **One Semi-Honest** Corruption

Trivial without

For each graph class, what is the minimal (cryptographic) assumption required for 1-THB and 1-THAB?

Our Work: The Simplest Setting

- **Broadcast** Only
  (and Anonymous Broadcast)
- **One Semi-Honest** Corruption

Trivial without

For each graph class, what is the minimal (cryptographic) assumption required for 1-THB and 1-THAB?

Information Theoretic (IT) / Key-Agreement (KA) / Oblivious Transfer (OT)

1-THB

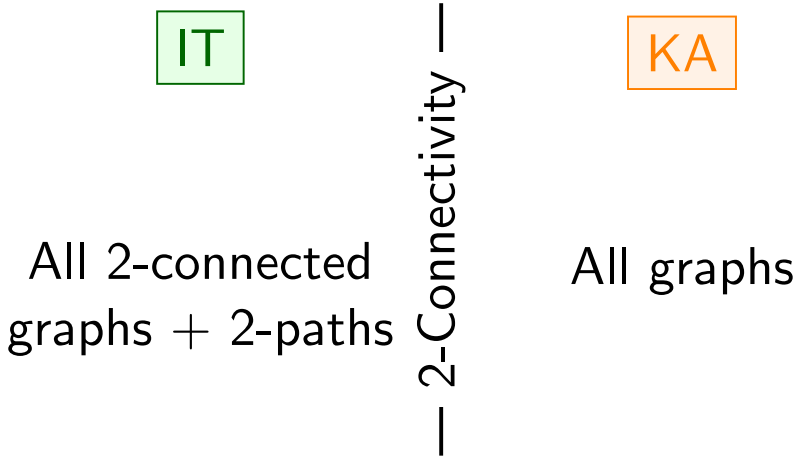
IT

All 2-connected graphs + 2-paths

KA

All graphs

1-THB


IT
KA
2-Connectivity
All 2-connected graphs + 2-paths
All graphs

1-THB

IT

All 2-connected graphs + 2-paths

Cycles of fixed length

2-Connectivity

KA

All graphs

Paths of length four

Previously:

1-THB

IT

All 2-connected graphs + 2-paths

2-Connectivity

KA

All graphs

1-THAB

IT

All 2-connected graphs + 2-paths

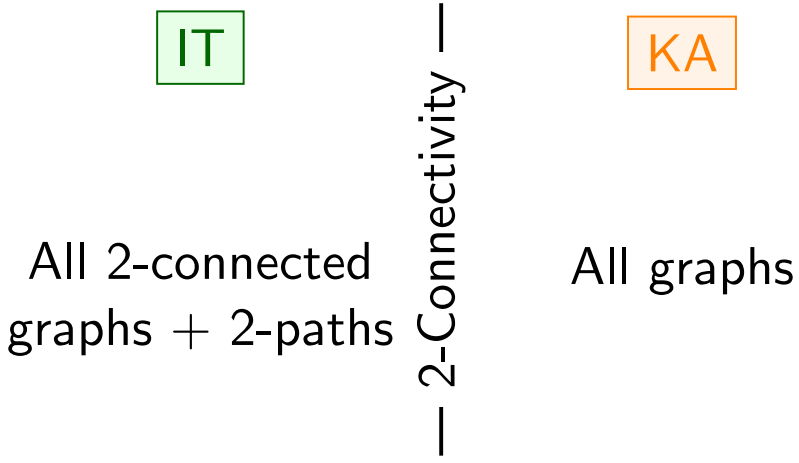
KA

All graphs with ≥ 3 nodes

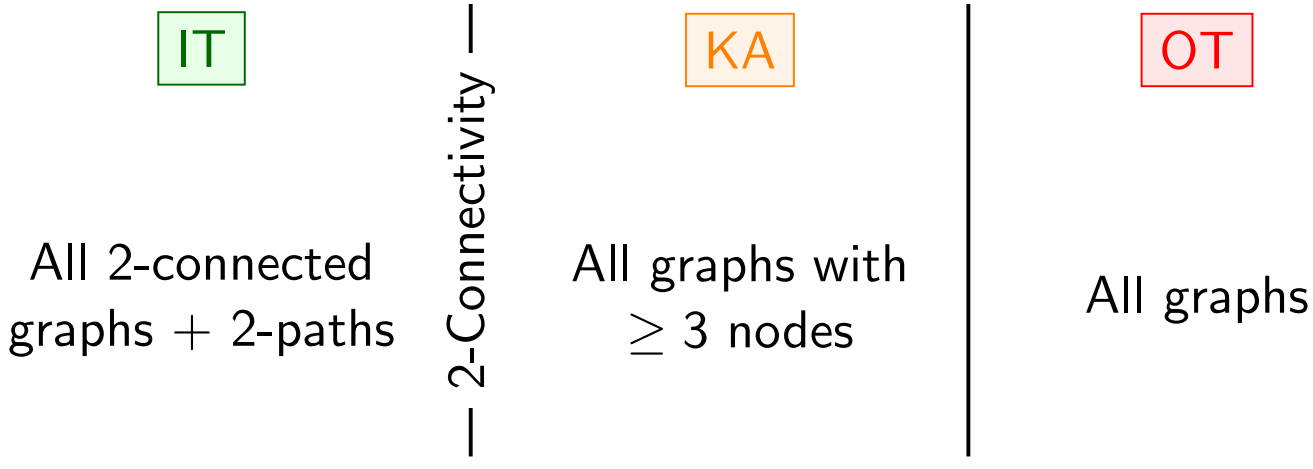
OT

All graphs

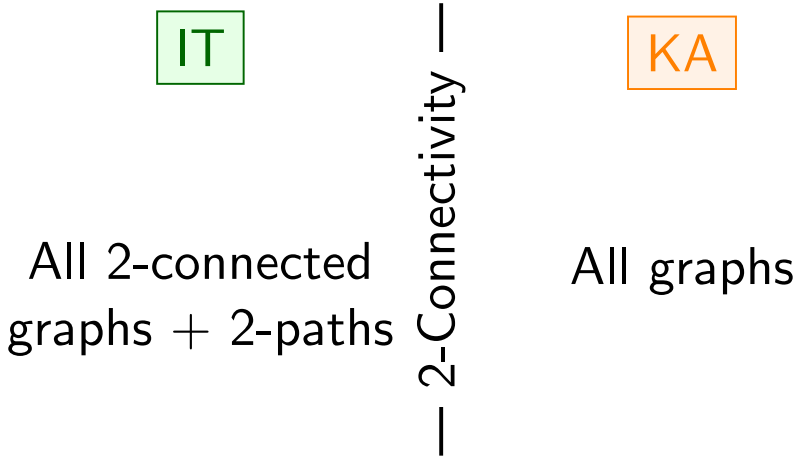
1-THB


IT
KA
2-Connectivity
All 2-connected graphs + 2-paths
All graphs

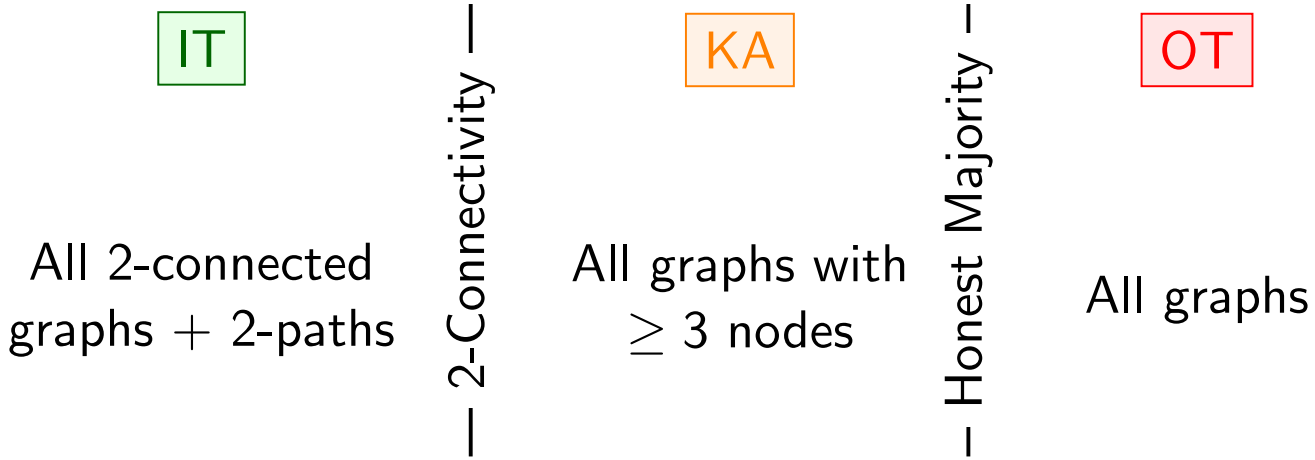
1-THAB


IT
KA
OT
2-Connectivity
All 2-connected graphs + 2-paths
All graphs with ≥ 3 nodes
All graphs

1-THB


IT
KA
2-Connectivity
All 2-connected graphs + 2-paths
All graphs

1-THAB


IT
KA
OT
2-Connectivity
Honest Majority
All 2-connected graphs + 2-paths
All graphs with ≥ 3 nodes
All graphs

Finer Points

1-THAB

IT	KA	OT
All 2-connected graphs + 2-paths	All graphs with ≥ 3 nodes	All graphs

- The classification is not 100% complete for KA vs OT

Finer Points

1-THAB

IT	KA	OT
All 2-connected graphs + 2-paths	All graphs with ≥ 3 nodes	All graphs

- The classification is not 100% complete for KA vs OT
- The information-theoretic protocol is not strictly *efficient*

Finer Points

1-THAB

IT	KA	OT
All 2-connected graphs + 2-paths	All graphs with ≥ 3 nodes	All graphs

- The classification is not 100% complete for KA vs OT
- The information-theoretic protocol is not strictly *efficient*
- ~~"OT is necessary for 1-THAB on the class of all graphs"~~
  "infinitely often OT is necessary for constant-round 1-THAB on the class of paths of length 2 and 3"

Finer Points

1-THAB

IT	KA	OT
All 2-connected graphs + 2-paths	All graphs with ≥ 3 nodes	All graphs

- The classification is not 100% complete for KA vs OT
- The information-theoretic protocol is not strictly *efficient*
- ~~"OT is necessary for 1-THAB on the class of all graphs"~~
  "infinitely often OT is necessary for constant-round 1-THAB on the class of paths of length 2 and 3"

Finer Points

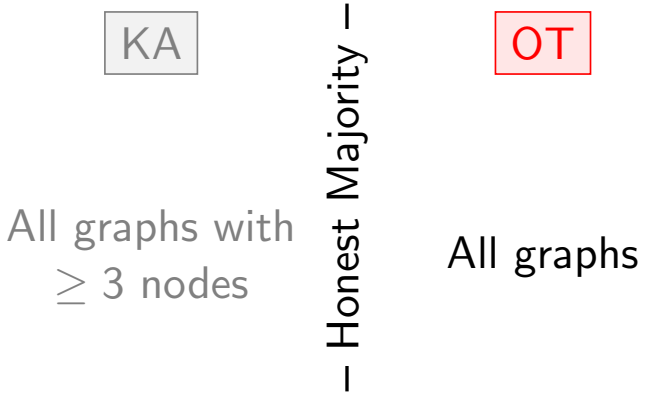
1-THAB

IT	KA	OT
All 2-connected graphs + 2-paths	All graphs with ≥ 3 nodes	All graphs

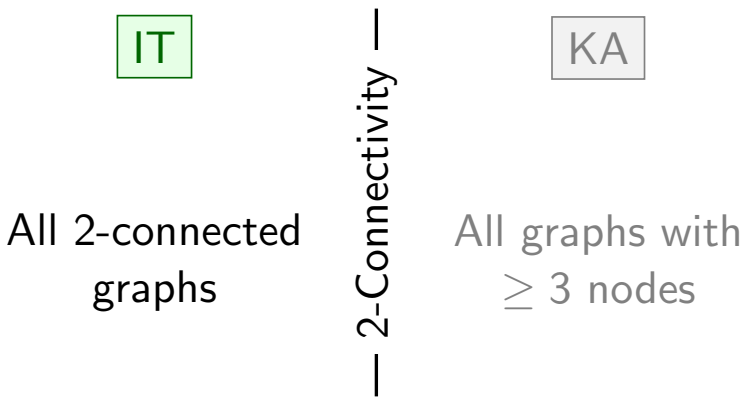
- The classification is not 100% complete for KA vs OT
- The information-theoretic protocol is not strictly *efficient*
- ~~"OT is necessary for 1-THAB on the class of all graphs"~~ "infinitely often OT is necessary for constant-round 1-THAB on the class of paths of length 2 and 3"

For Today

1. io-OT Requirement



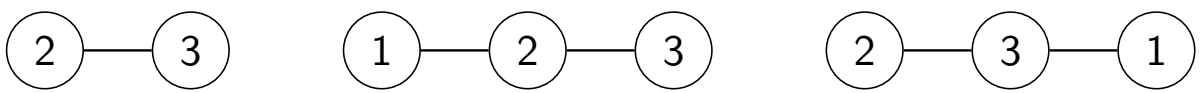
2. Unconditional Feasibility



1-THAB on Paths requires OT

$$[\text{2-round 1-THAB}(\mathcal{G}_{P_2\text{-vs-}P_3})] \Rightarrow \text{Semi-honest AND}$$

- *Functionality:* Anonymous Broadcast in 2 Rounds
- *Player Pool:* $\{①, ②, ③\}$
- *Graph Class:*



1-THAB on Paths requires OT

$$[\text{Cst-round 1-THAB}(\mathcal{G}_{P_2\text{-vs-}P_3})] \Rightarrow \text{Semi-honest io-AND}$$

- *Functionality:* Anonymous Broadcast in Constant Rounds
- *Player Pool:* $\{①, ②, ③\}$
- *Graph Class:* $\mathcal{G}_{P_2\text{-vs-}P_3}$

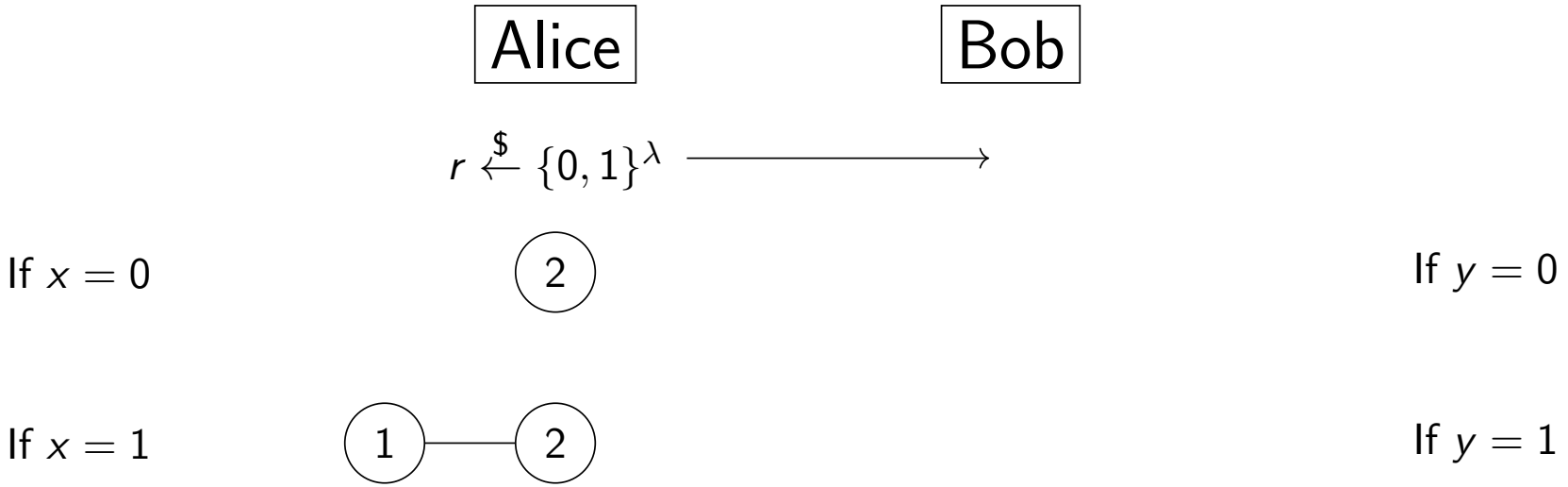
[2-round 1-THAB($\mathcal{G}_{P_2\text{-vs-}P_3}$)] $\Rightarrow$ OT (Correctness) 9/20

Alice	Bob
If $x = 0$	If $y = 0$
If $x = 1$	If $y = 1$

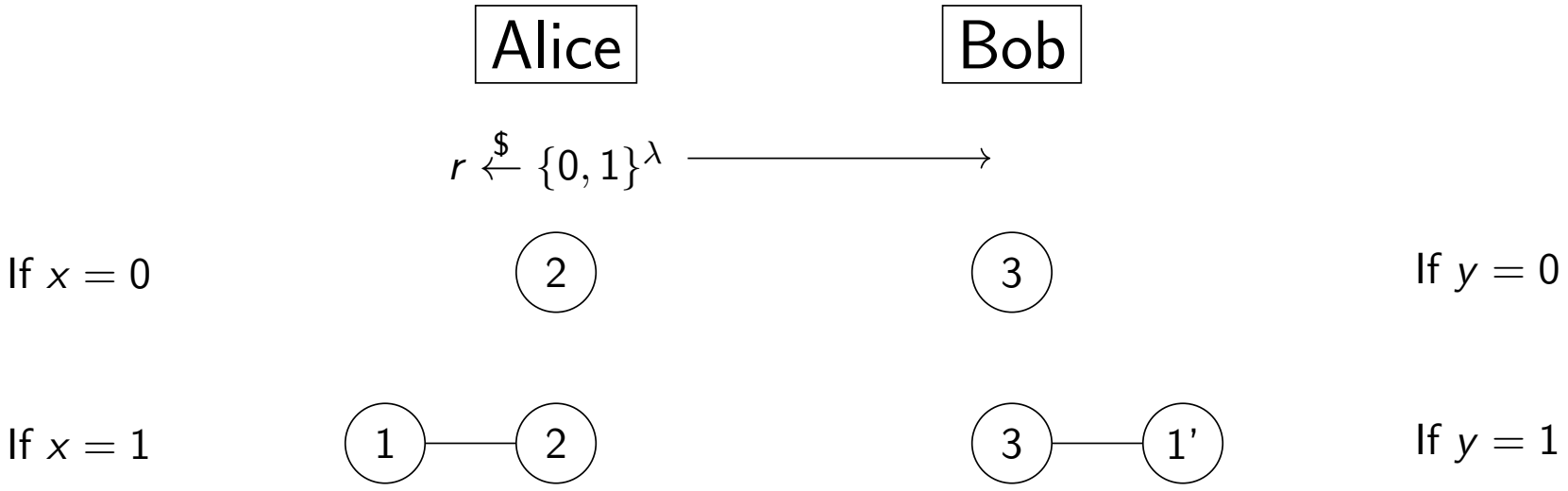
[2-round 1-THAB($\mathcal{G}_{P_2\text{-vs-}P_3}$)] $\Rightarrow$ OT (Correctness) 9/20

Alice		Bob
	$r \xleftarrow{\$} \{0,1\}^\lambda \longrightarrow$	
If $x = 0$		If $y = 0$
If $x = 1$		If $y = 1$

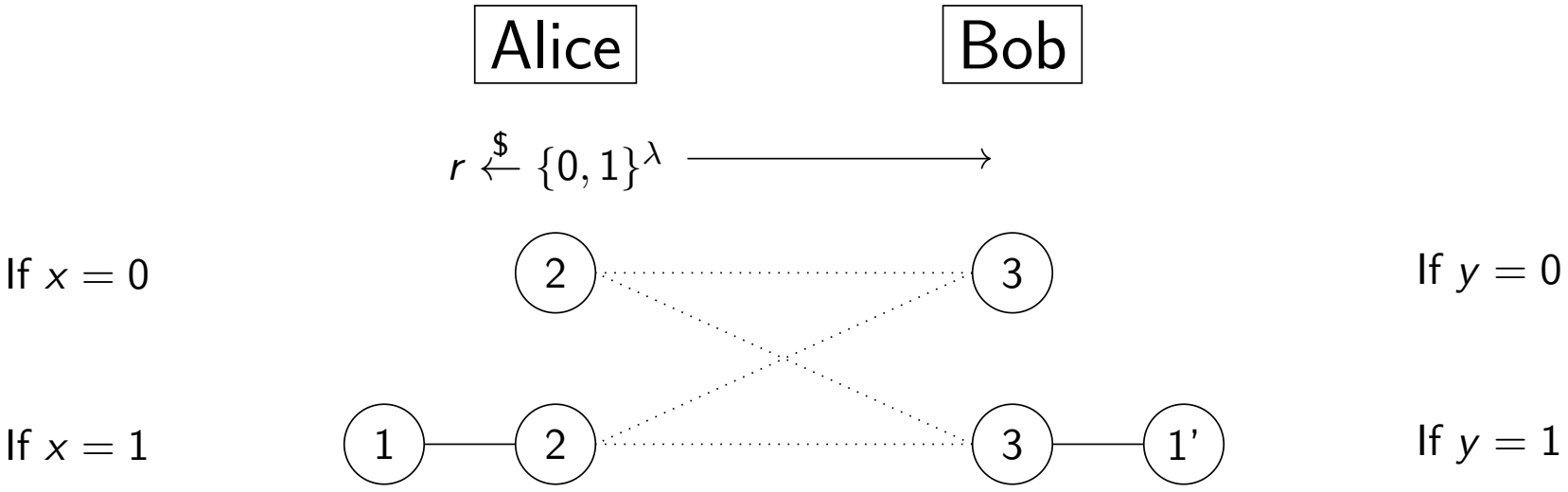
[2-round 1-THAB($\mathcal{G}_{P_2\text{-vs-}P_3}$)] $\Rightarrow$ OT (Correctness) 9/20



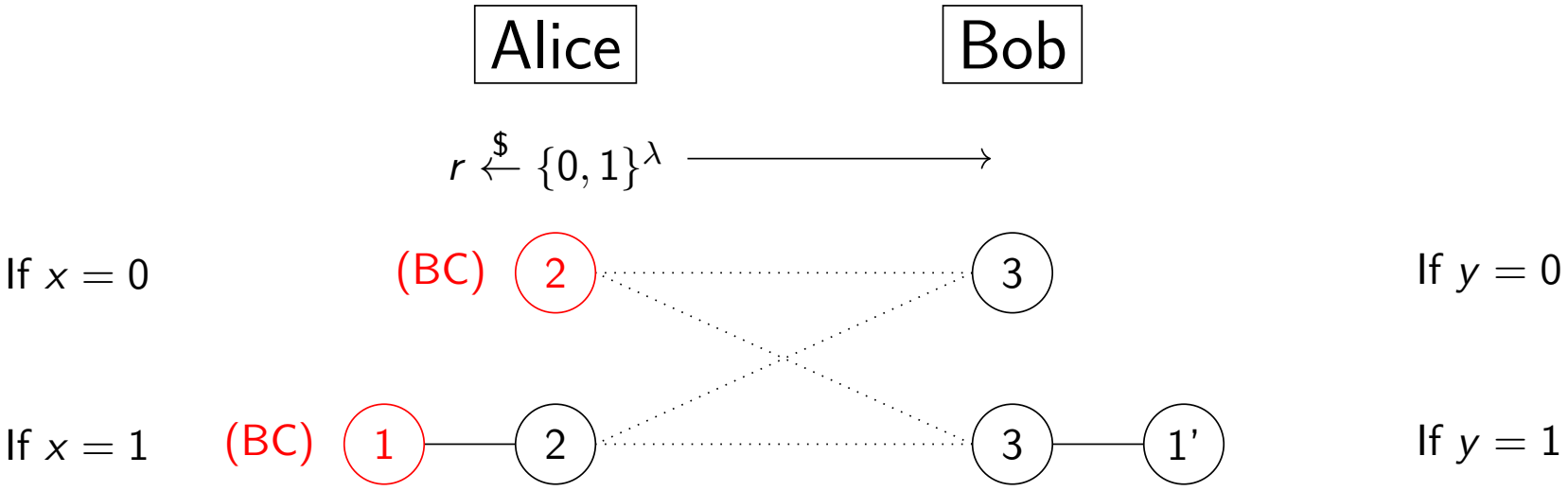
[2-round 1-THAB($\mathcal{G}_{P_2\text{-vs-}P_3}$)] $\Rightarrow$ OT (Correctness) 9/20



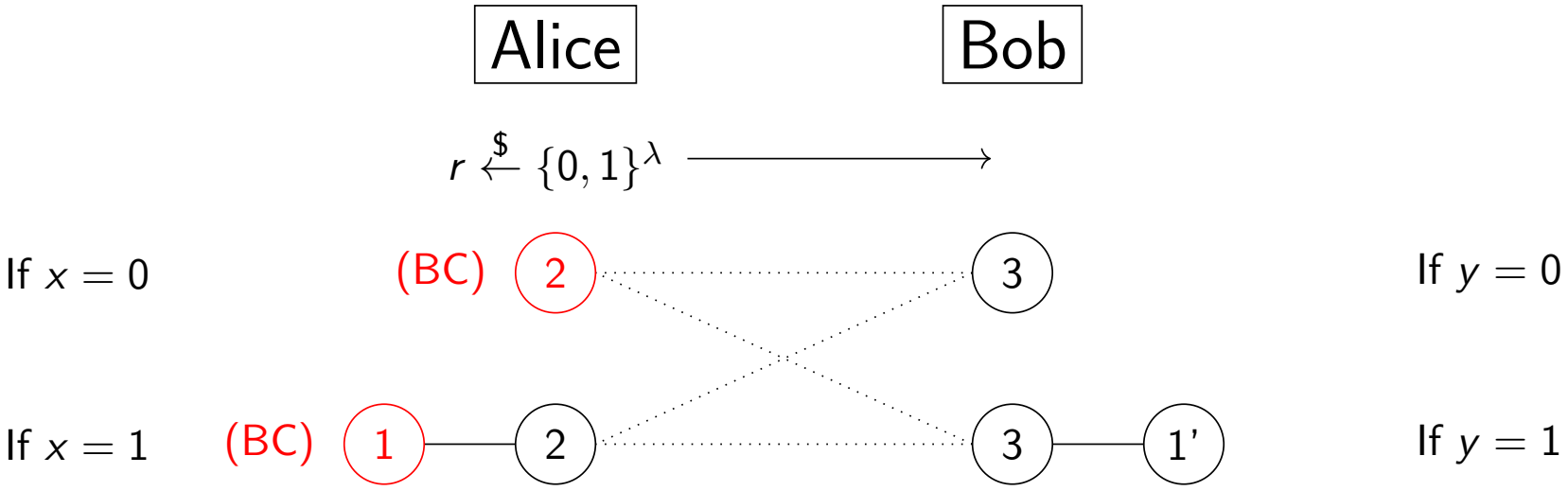
[2-round 1-THAB($\mathcal{G}_{P_2\text{-vs-}P_3}$)] $\Rightarrow$ OT (Correctness) 9/20



[2-round 1-THAB($\mathcal{G}_{P_2\text{-vs-}P_3}$)] $\Rightarrow$ OT (Correctness) 9/20



[2-round 1-THAB($\mathcal{G}_{P_2\text{-vs-}P_3}$)] $\Rightarrow$ OT (Correctness) 9/20



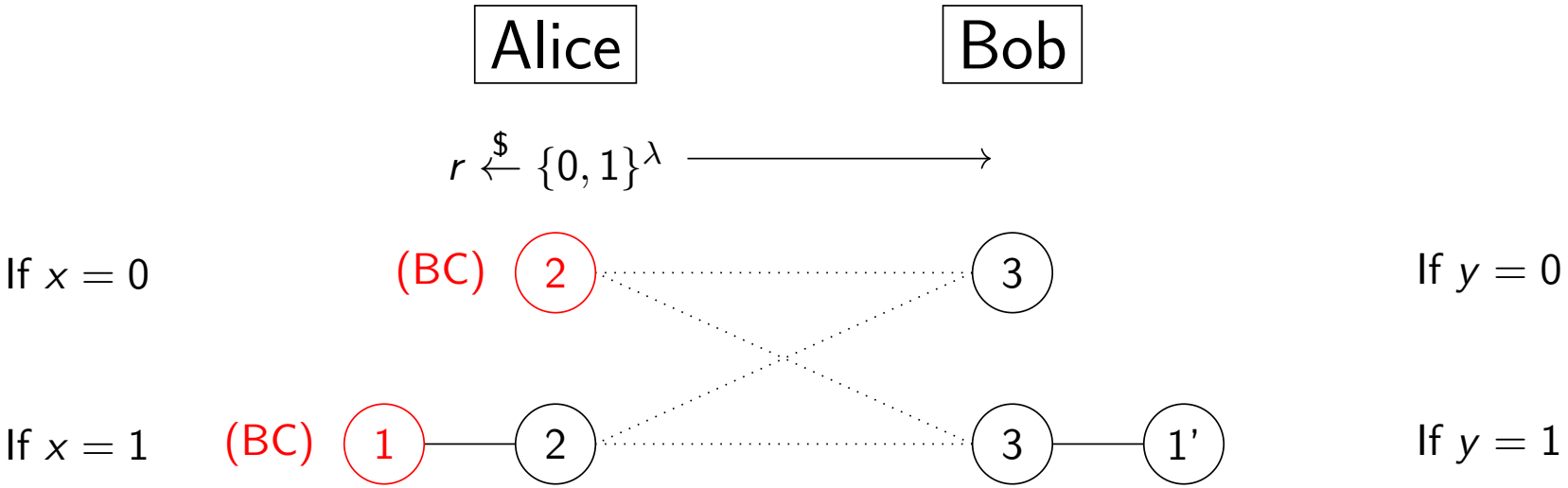
If $(x, y) = (0, 0)$:

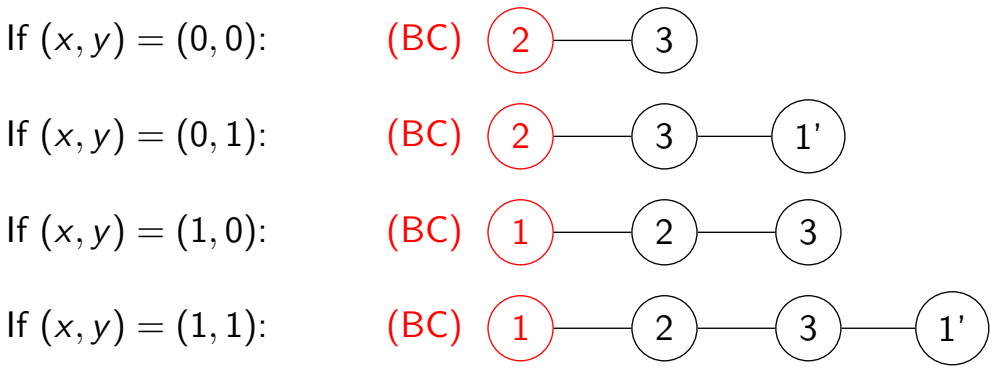
If $(x, y) = (0, 1)$:

If $(x, y) = (1, 0)$:

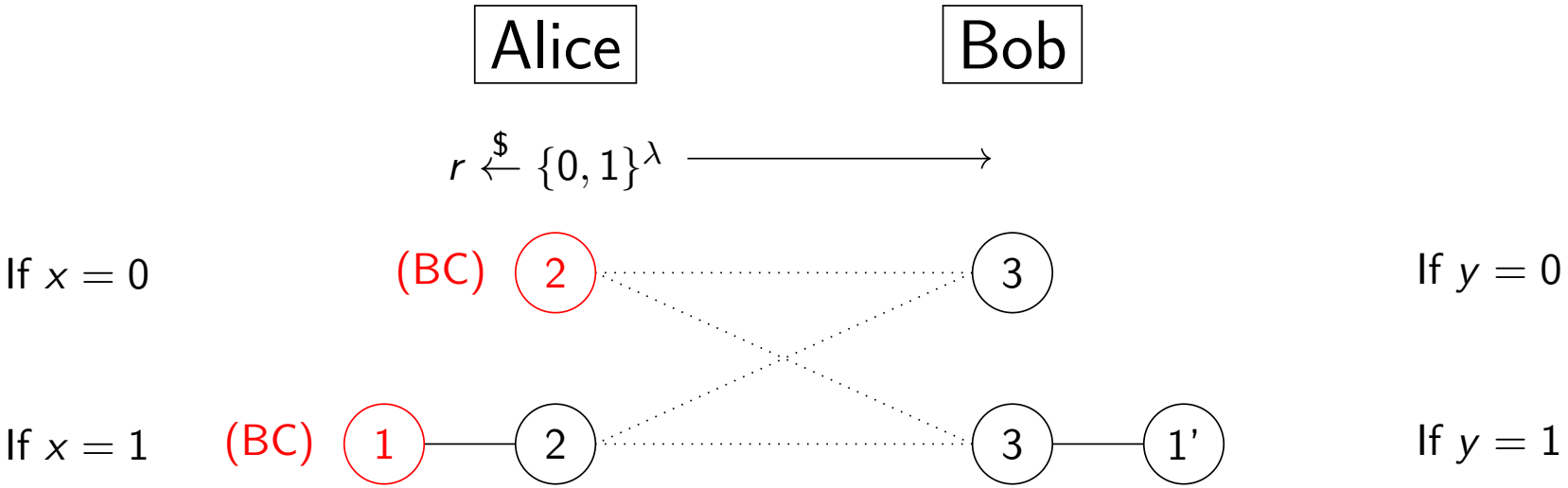
If $(x, y) = (1, 1)$:

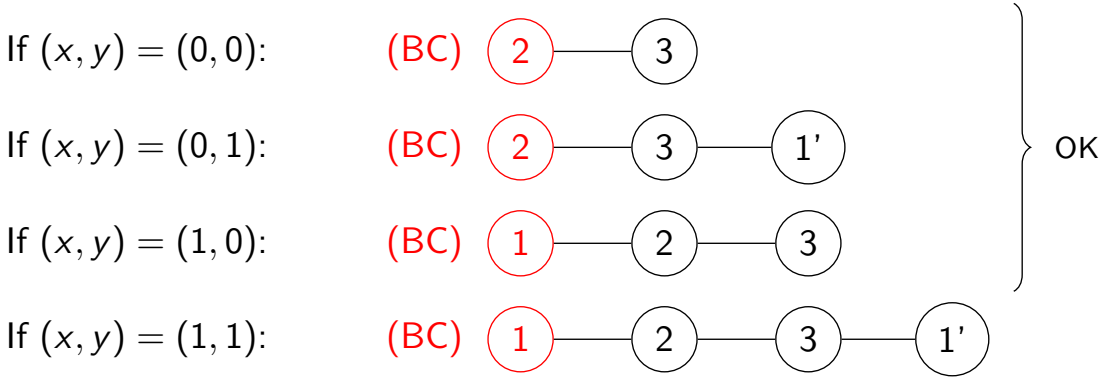
[2-round 1-THAB($\mathcal{G}_{P_2\text{-vs-}P_3}$)] $\Rightarrow$ OT (Correctness) 9/20



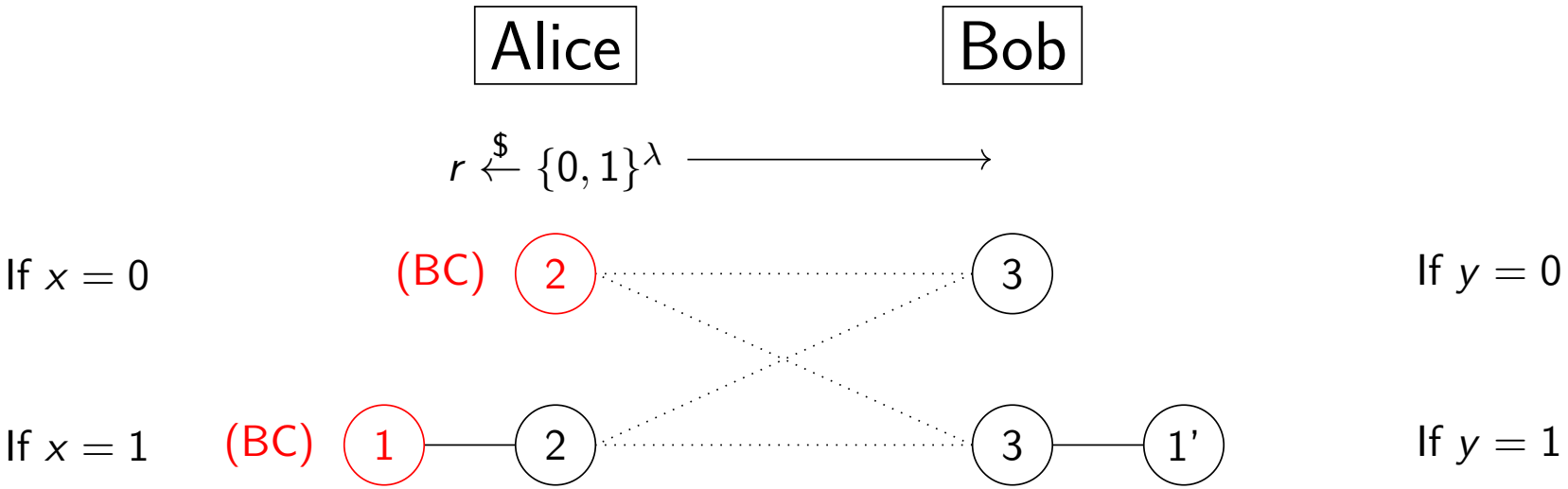


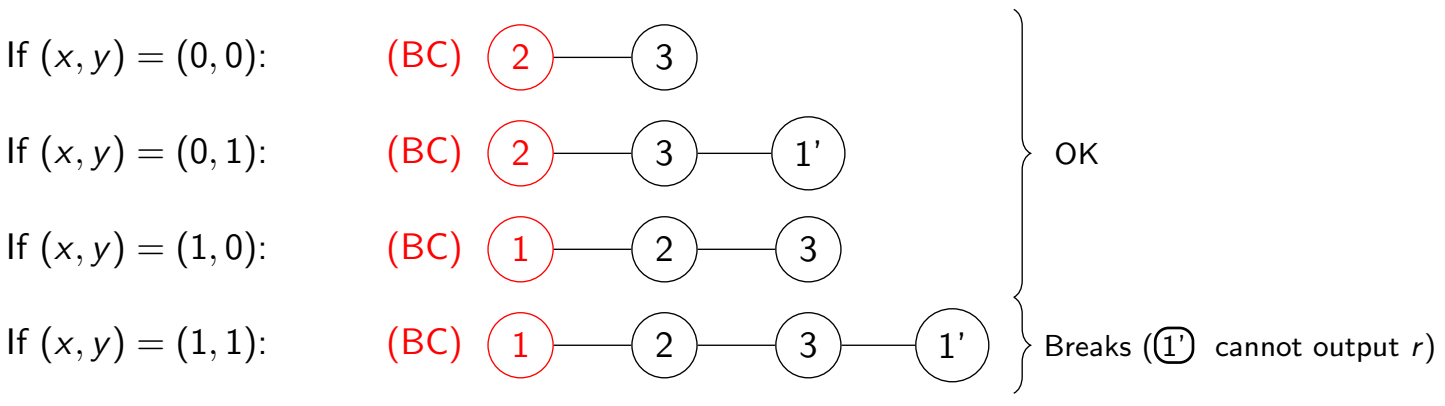
[2-round 1-THAB($\mathcal{G}_{P_2\text{-vs-}P_3}$)] $\Rightarrow$ OT (Correctness) 9/20





[2-round 1-THAB($\mathcal{G}_{P_2\text{-vs-}P_3}$)] $\Rightarrow$ OT (Correctness) 9/20





[2-round 1-THAB($\mathcal{G}_{P_2\text{-}vs\text{-}P_3}$)] $\Rightarrow$ OT (Security) 10/20

Alice

Bob

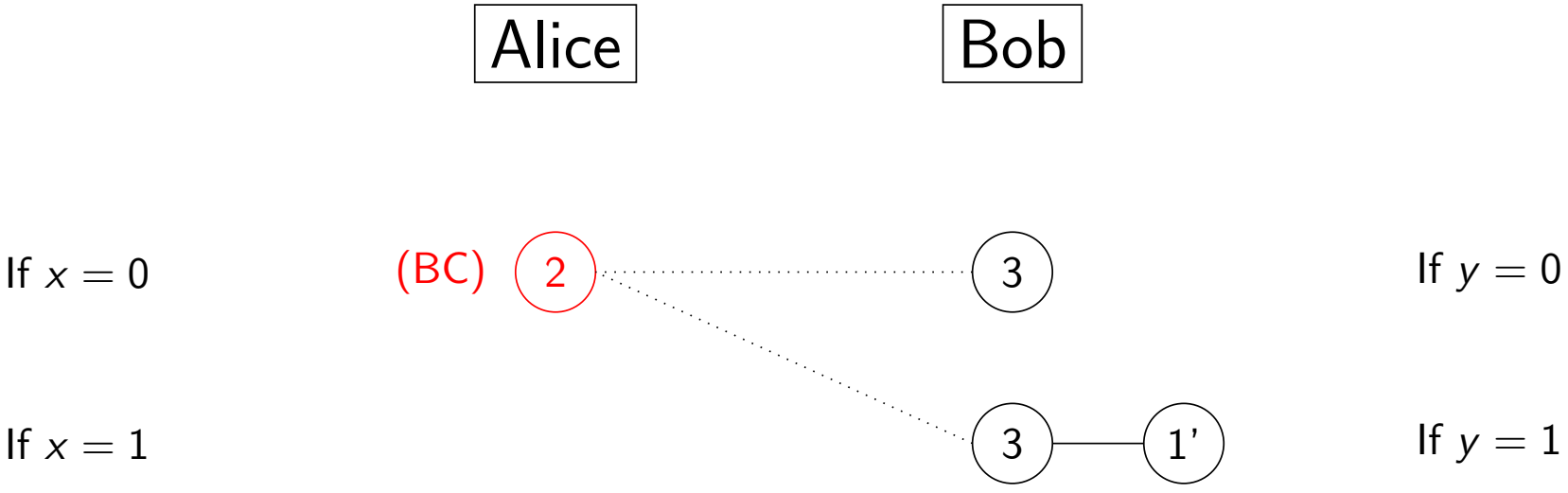
If $x = 0$

If $y = 0$

If $x = 1$

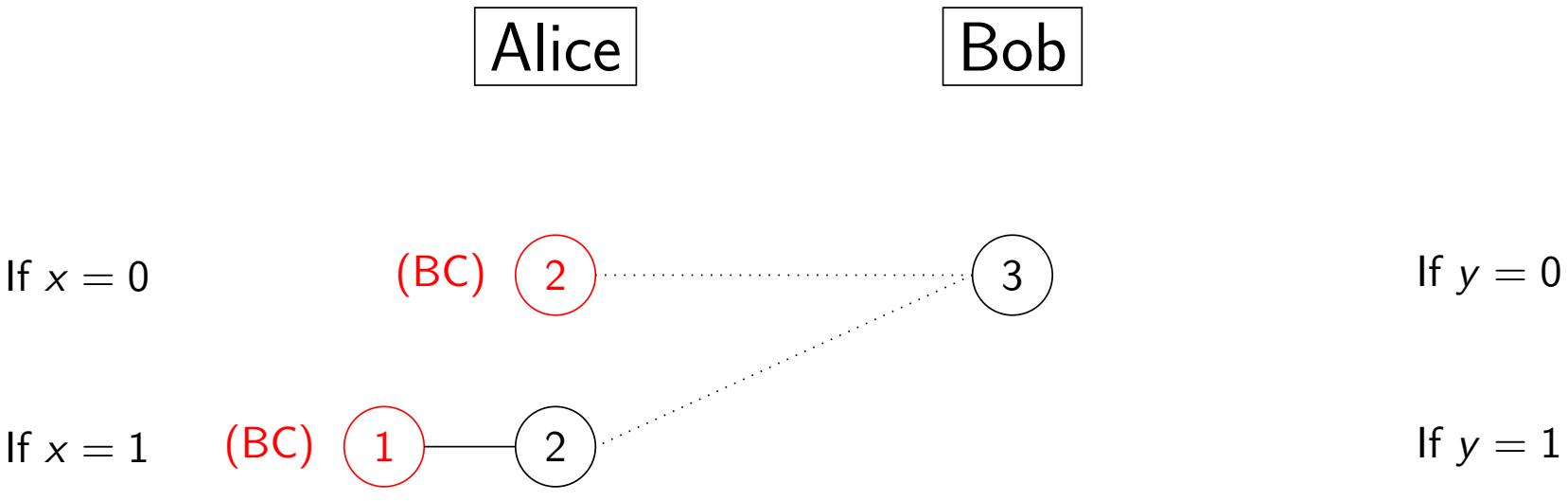
If $y = 1$

[2-round 1-THAB($\mathcal{G}_{P_2\text{-vs-}P_3}$)] $\Rightarrow$ OT (Security) 10/20



Claim: If $x = 0$, Alice cannot learn y

[2-round 1-THAB($\mathcal{G}_{P_2\text{-vs-}P_3}$)] $\Rightarrow$ OT (Security) 10/20

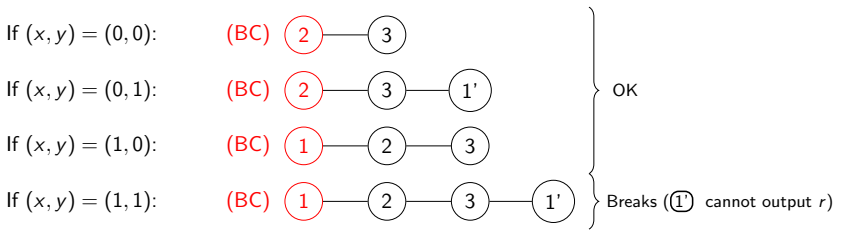


Claim: If $y = 0$, Bob cannot learn x

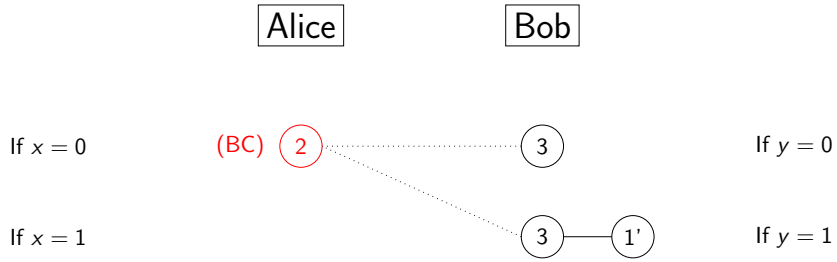
Summing-Up

$$[\text{2-round 1-THAB}(\mathcal{G}_{P_2\text{-vs-}P_3})] \Rightarrow \text{OT}$$

Correctness



Security

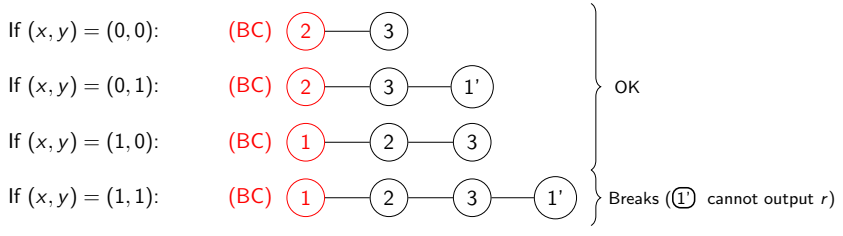


Claim: If $x = 0$, Alice cannot learn y

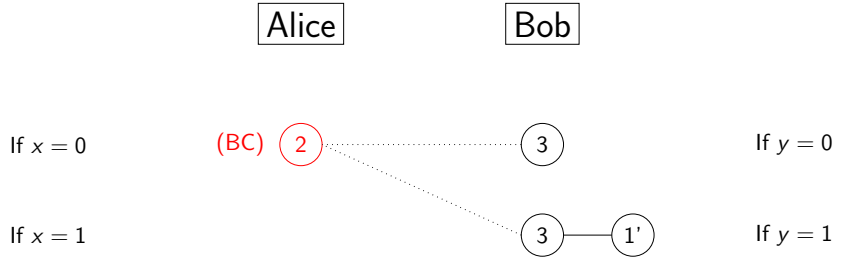
Summing-Up

$$[\text{2-round 1-THAB}(\mathcal{G}_{P_2\text{-vs-}P_3})] \Rightarrow \text{OT}$$

Correctness



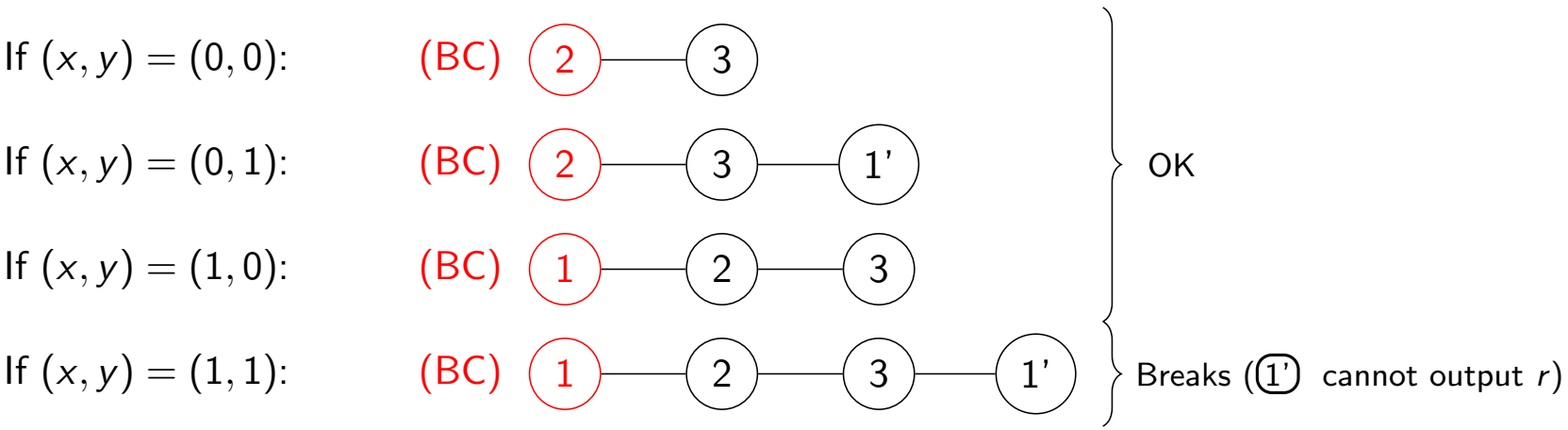
Security



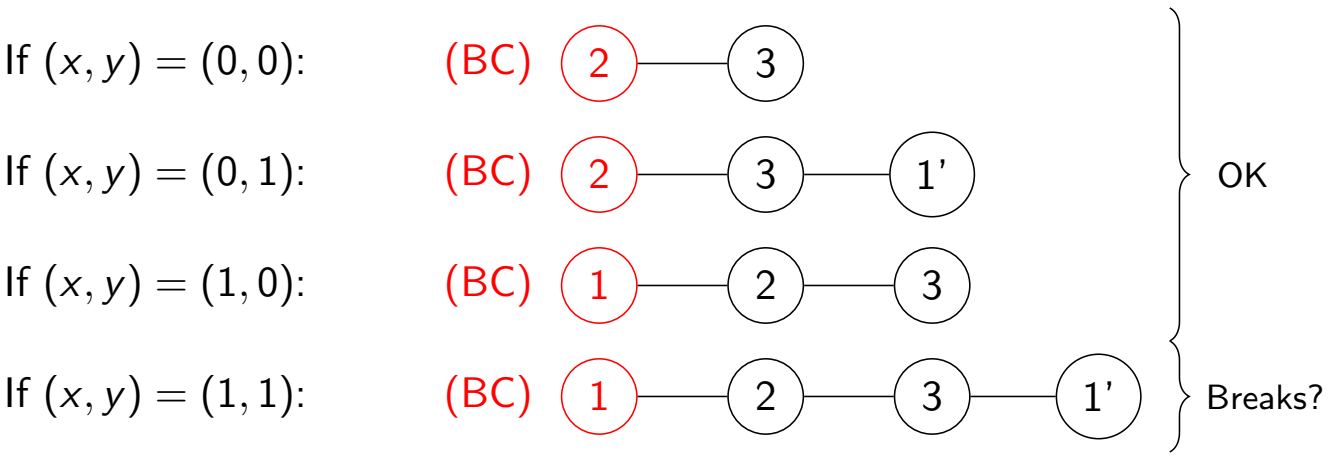
Claim: If $x=0$, Alice cannot learn y

How to extend this to "[constant-round 1-THAB$(\mathcal{G}_{P_2\text{-vs-}P_3})$] $\Rightarrow$ io-OT"?

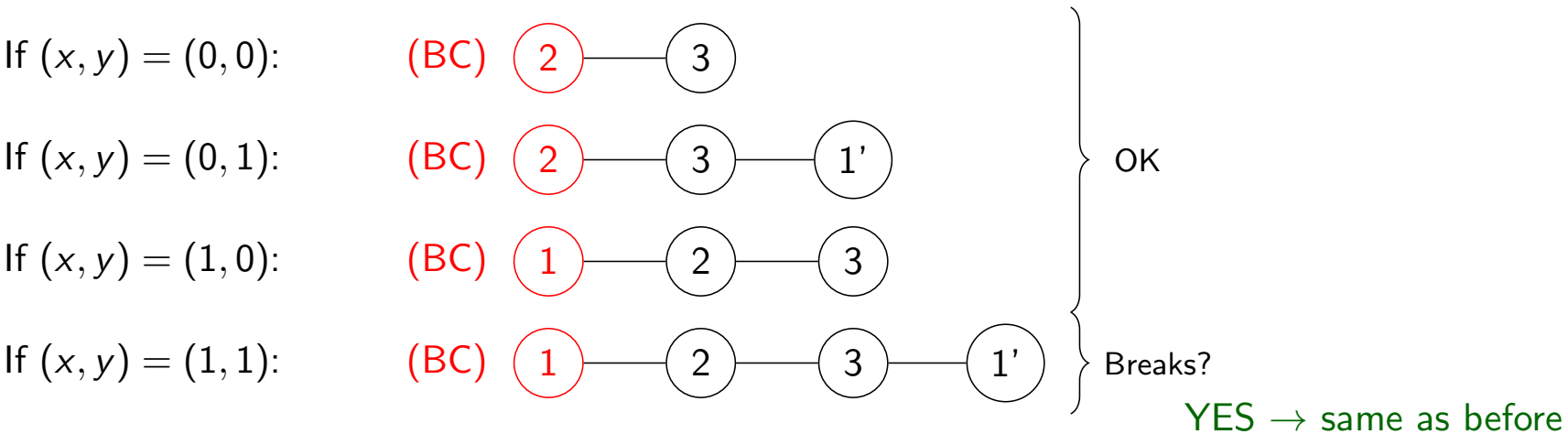
Where we used the 2-round hyothesis



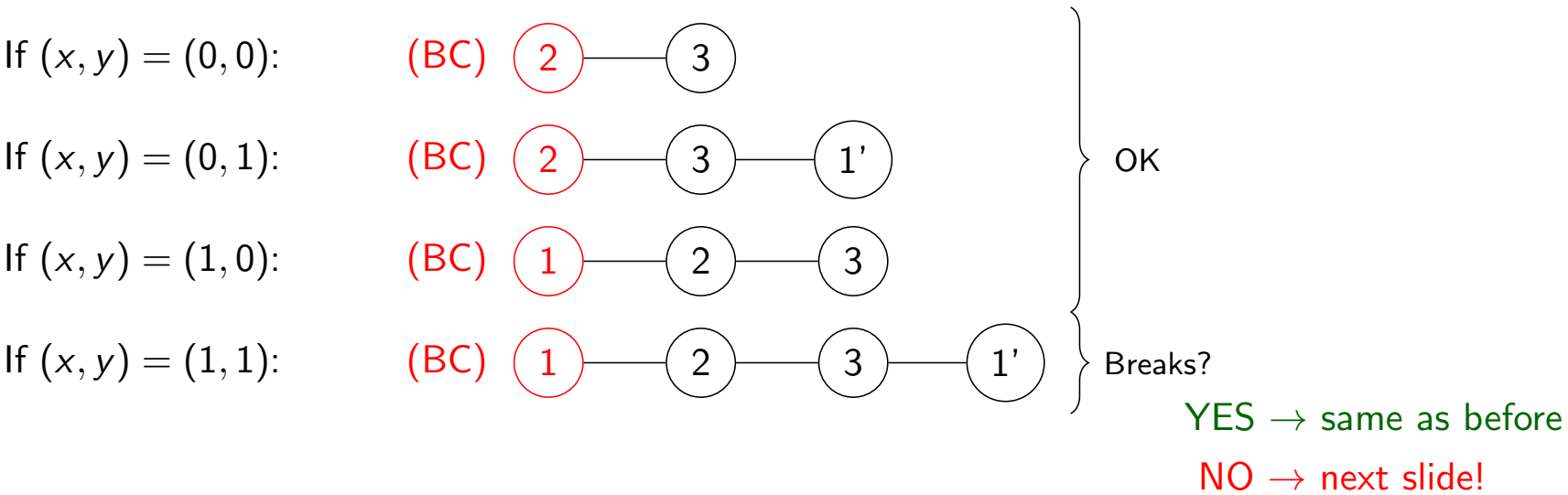
Where we used the 2-round hyothesis



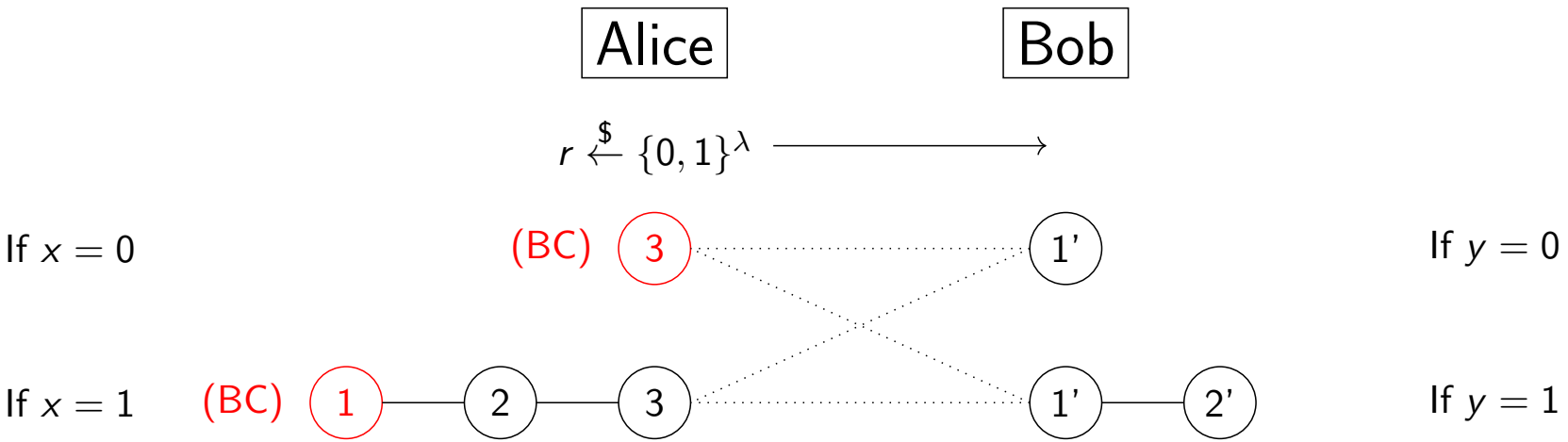
Where we used the 2-round hyothesis

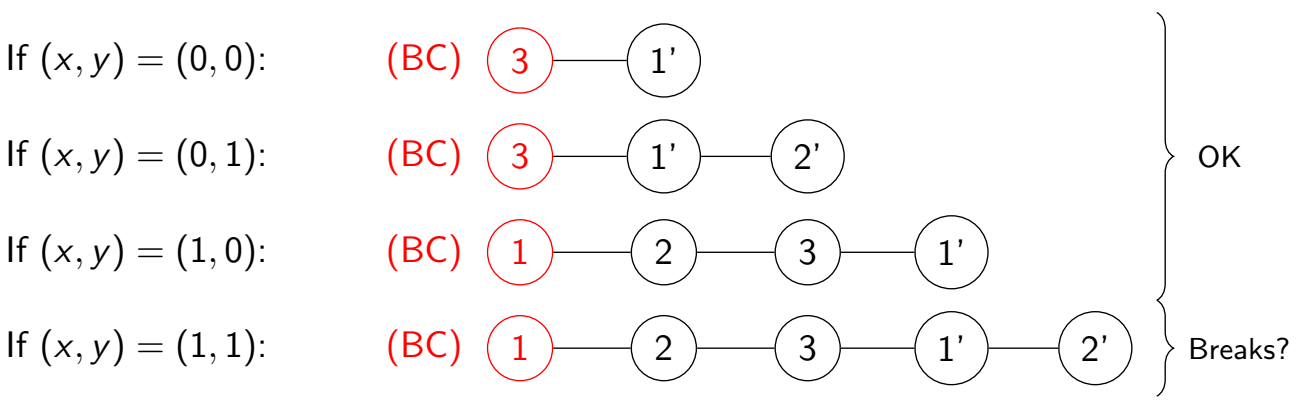


Where we used the 2-round hyothesis



Extending Lower Bound to Constant-Round THAB Protocols





1-THAB($\mathcal{G}_{2\text{-}conn}$) is possible

1-THAB($\mathcal{G}_{2\text{-}conn}$) is possible Unconditionally

- *Functionality:* Anonymous Broadcast
- *Player Pool:* $\{\boxed{P_1}, \ldots, \boxed{P_N}\}$
- *Graph Class:* All two-connected graphs

1-THAB($\mathcal{G}_{2\text{-}conn}$) is possible

1-THAB($\mathcal{G}_{2\text{-}conn}$) is possible Unconditionally

- *Functionality:* Anonymous Broadcast
- *Player Pool:* $\{P_1, \ldots, P_N\}$
- *Graph Class:* All two-connected graphs with all N players

1-THAB($\mathcal{G}_{2\text{-}conn}$) is possible

1-SMT$_{s \to t}(\mathcal{G}_{2\text{-}conn})$ is possible Unconditionally

- *Functionality:* Secure Message Transmission from Ⓢ to Ⓣ
- *Player Pool:* $\{P_1, \ldots, P_N\}$
- *Graph Class:* All two-connected graphs with all N players

Roadmap

1. Primitive: SMT on a single 2-connected graph
   - Correctness and Message-Security

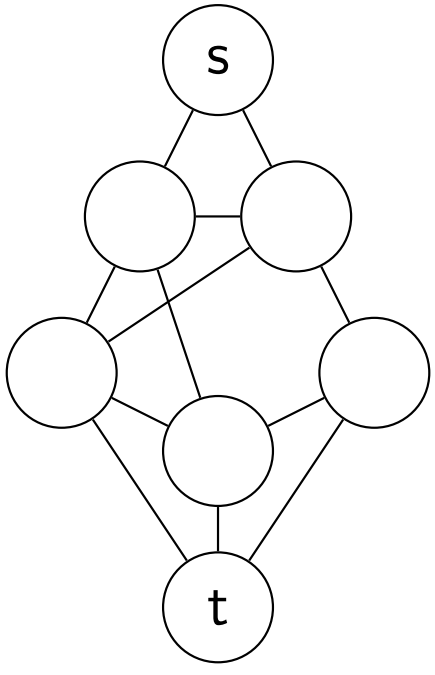
Roadmap

1. Primitive: SMT on a single 2-connected graph
   - Correctness and Message-Security
2. Run $|\mathcal{G}_{2\text{-}conn}|$ instances in parallel, one per graph in the class
   - For the correct guess, correctness and message-security
   - For an incorrect guess, the run is *censored* to destroy the message

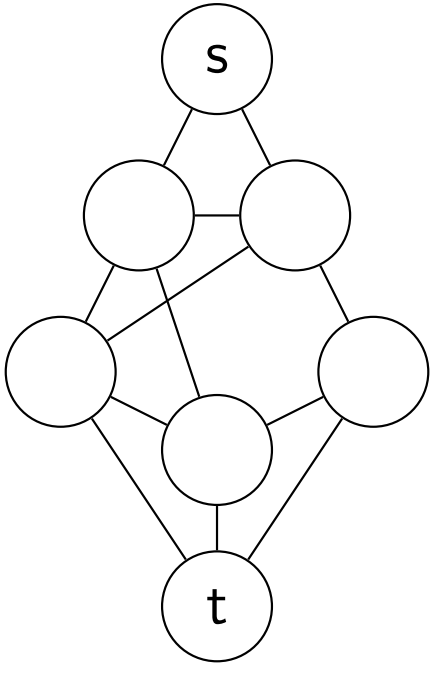
Roadmap

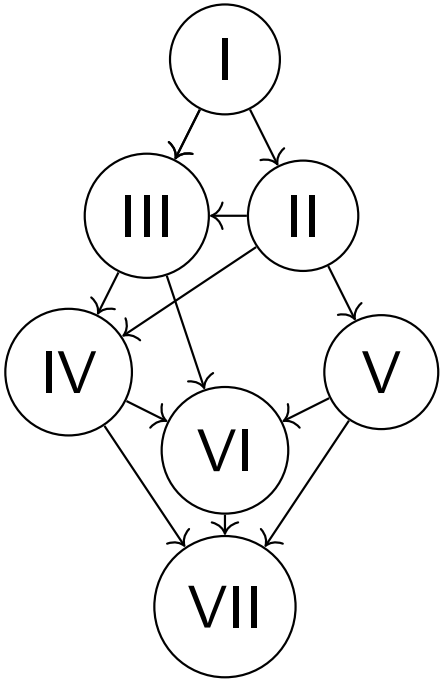
1. Primitive: SMT on a single 2-connected graph
   - Correctness and Message-Security
2. Run $|\mathcal{G}_{2\text{-}conn}|$ instances in parallel, one per graph in the class
   - For the correct guess, correctness and message-security
   - For an incorrect guess, the run is *censored* to destroy the message
3. Merge all runs to hide which guess was correct
   - Topology-hiding

Step 1: The "Single-Graph" SMT Primitive



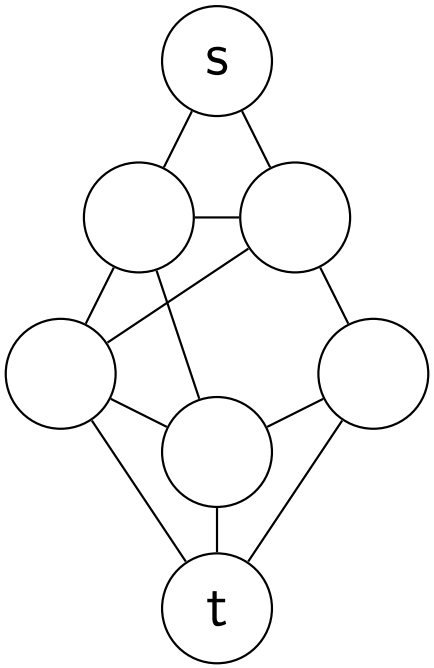
Step 1: The "Single-Graph" SMT Primitive

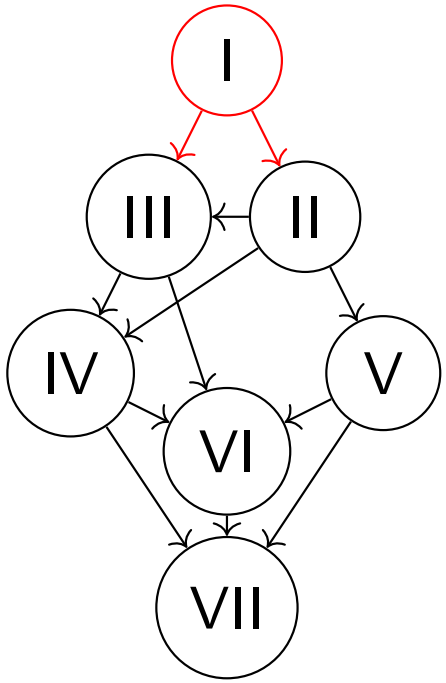




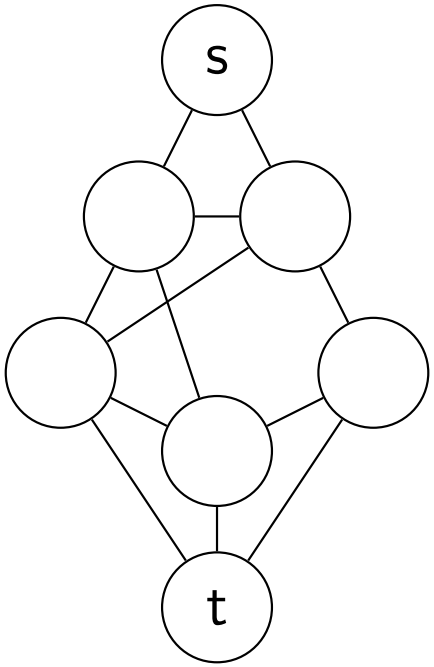
bipolar orientation from s to t:
orientation as D.A.G. with single source s and single sink t

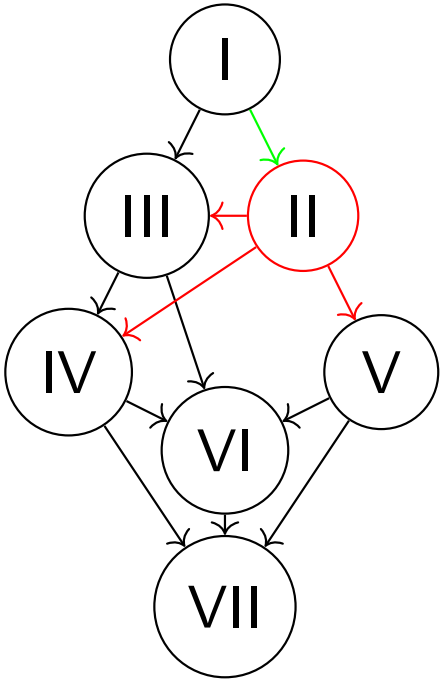
Step 1: The "Single-Graph" SMT Primitive



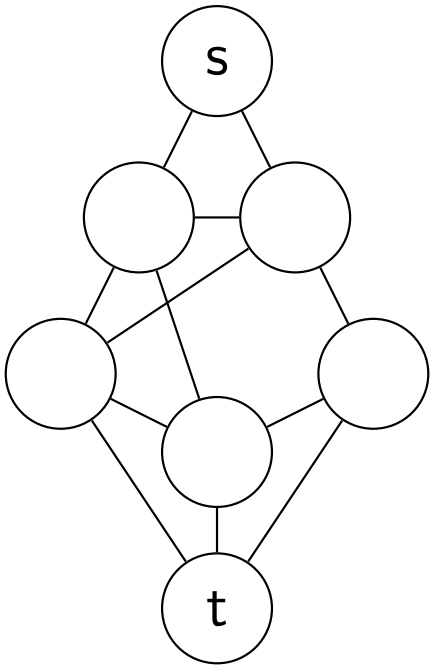


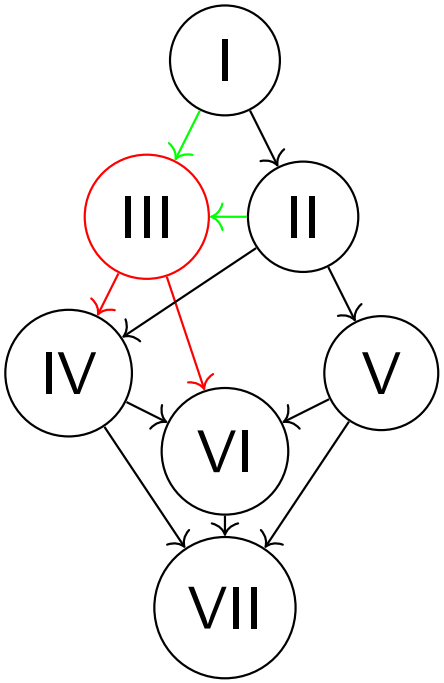
Step 1: The "Single-Graph" SMT Primitive



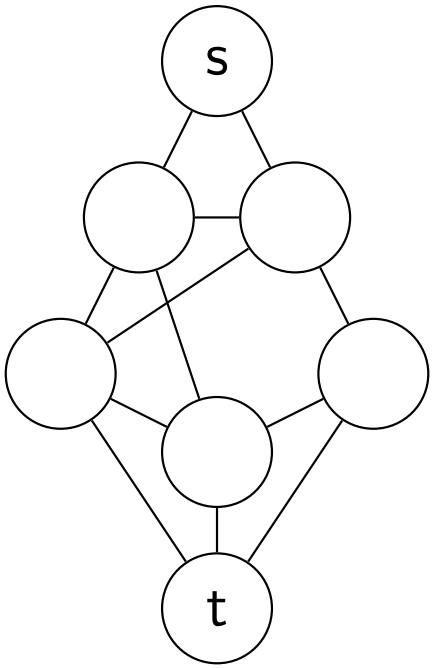


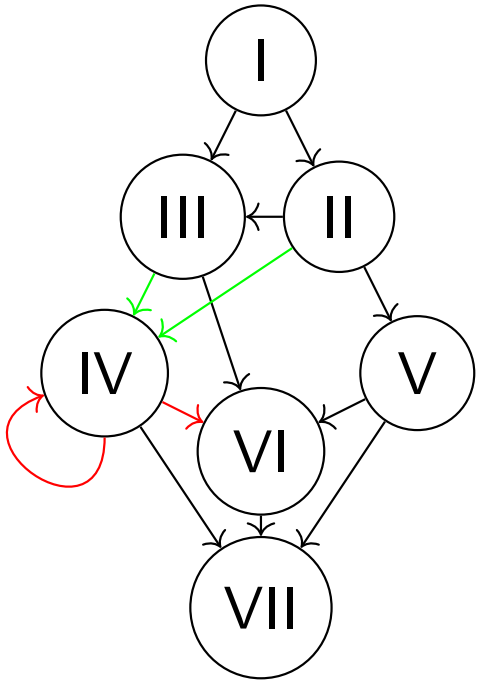
Step 1: The "Single-Graph" SMT Primitive



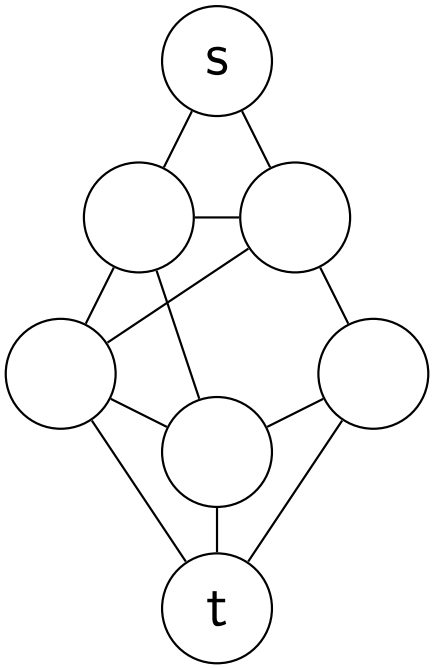


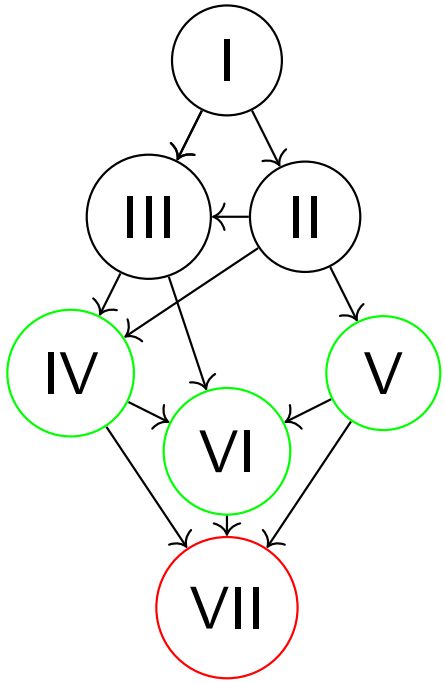
Step 1: The "Single-Graph" SMT Primitive





Step 1: The "Single-Graph" SMT Primitive





Step 2: Parallel and Censored Runs

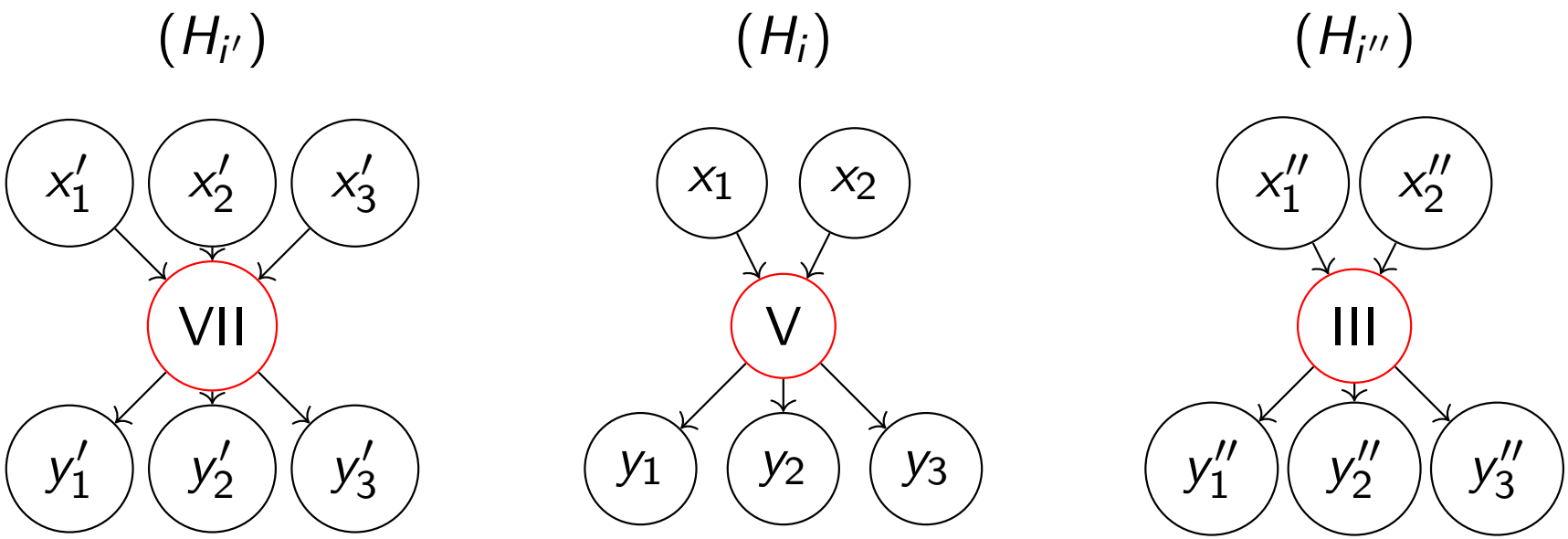
Public:

- $\mathcal{G} = \{G_1, G_2, \ldots, G_k\}$ on $V = [N]$
- st-orientations $s \to t : H_1, H_2, \ldots H_k$

Step 2: Parallel and Censored Runs

Public:

- $\mathcal{G} = \{G_1, G_2, \ldots, G_k\}$ on $V = [N]$
- st-orientations $s \to t : H_1, H_2, \ldots H_k$

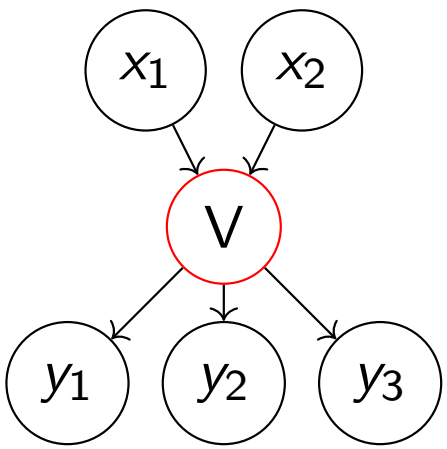


Step 2: Parallel and Censored Runs

Public:

- $\mathcal{G} = \{G_1, G_2, \ldots, G_k\}$ on $V = [N]$
- st-orientations $s \to t : H_1, H_2, \ldots H_k$

(H_i)



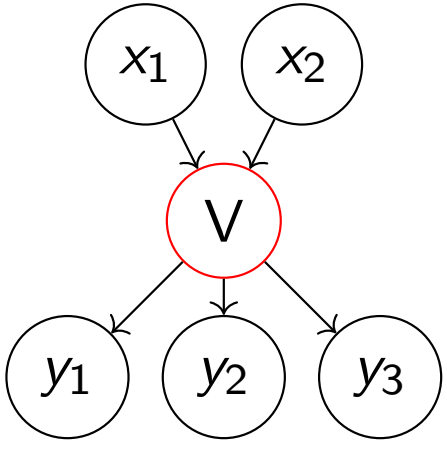
Step 2: Parallel and Censored Runs

Public:

- $\mathcal{G} = \{G_1, G_2, \ldots, G_k\}$ on $V = [N]$
- st-orientations $s \to t : H_1, H_2, \ldots H_k$

If $N_G = N_{G_i}$

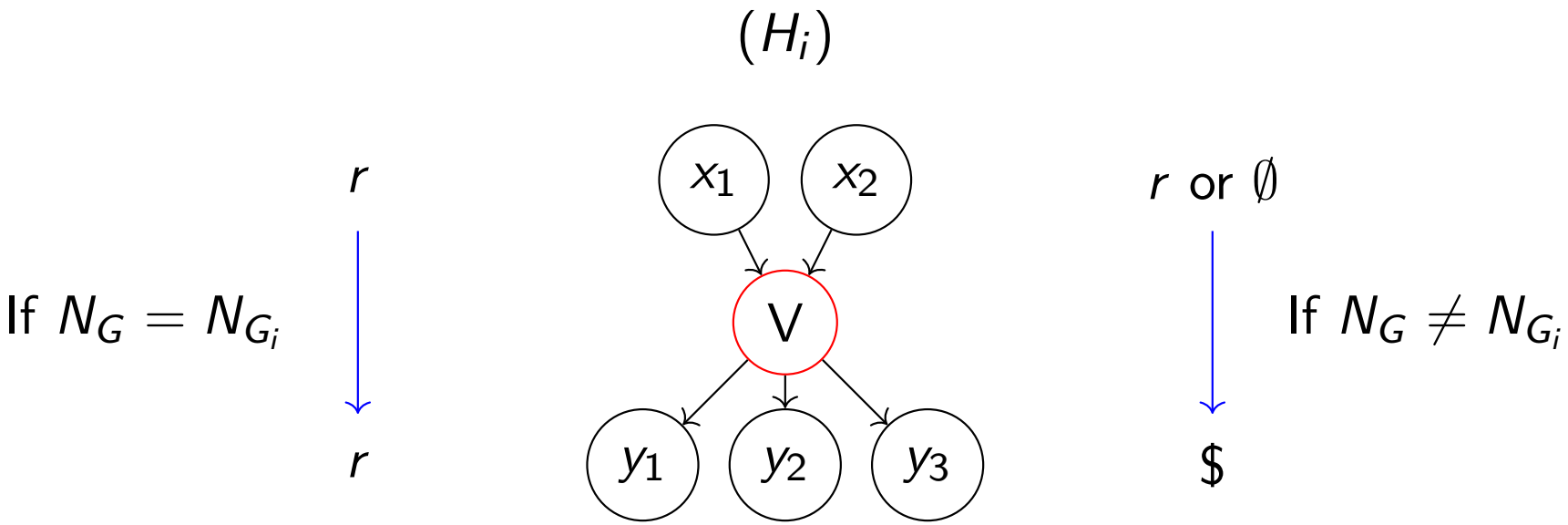
(H_i)



Step 2: Parallel and Censored Runs

Public:

- $\mathcal{G} = \{G_1, G_2, \ldots, G_k\}$ on $V = [N]$
- st-orientations $s \to t : H_1, H_2, \ldots H_k$



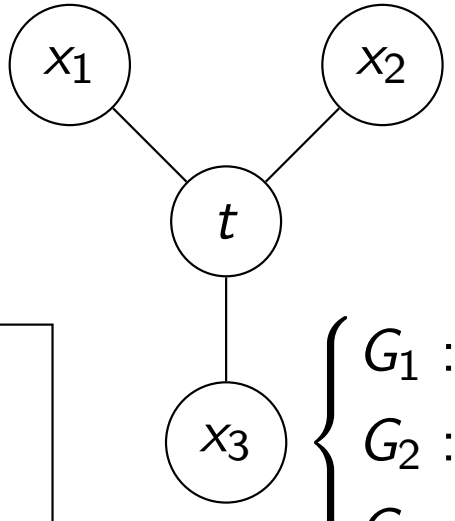
Step 3: Merging the Runs

Public:

- $\mathcal{G} = \{G_1, G_2, \ldots, G_k\}$
- st-orientations $s \to t : H_1, H_2, \ldots H_k$

(real graph G_2)

$$\begin{cases} G_1 : m_1 \\ G_2 : m_2 \\ G_3 : m_3 \end{cases}$$



$$\begin{cases} G_1 : m'_1 \\ G_2 : m'_2 \\ G_3 : m'_3 \end{cases}$$

$$\begin{cases} G_1 : m''_1 \\ G_2 : m''_2 \\ G_3 : m''_3 \end{cases}$$

$$\begin{cases} m_1 + m'_1 + m''_1 = \$ \\ m_2 + m'_2 + m''_2 = m_{BC} \\ m_3 + m'_3 + m''_3 = \$\$ \end{cases}$$

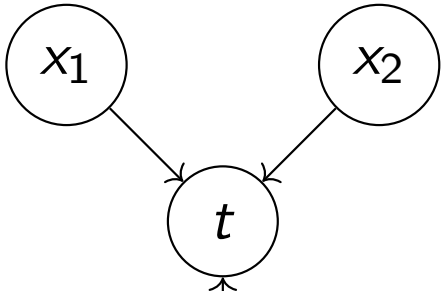
Step 3: Merging the Runs

Public:

- $\mathcal{G} = \{G_1, G_2, \ldots, G_k\}$
- st-orientations $s \to t : H_1, H_2, \ldots H_k$

(real graph G_2)

$$\begin{cases} G_1 : \overrightarrow{m_1} \\ G_2 : \overrightarrow{m_2} \\ G_3 : \overrightarrow{m_3} \end{cases}$$



$$\begin{cases} G_1 : \overrightarrow{m_1}' \\ G_2 : \overrightarrow{m_2}' \\ G_3 : \overrightarrow{m_3}' \end{cases}$$

$$\begin{cases} G_1 : \overrightarrow{m_1}'' \\ G_2 : \overrightarrow{m_2}'' \\ G_3 : \overrightarrow{m_3}'' \end{cases}$$

$$\begin{cases} \overrightarrow{m_1} + \overrightarrow{m_1}' + \overrightarrow{m_1}'' = \$ \\ \overrightarrow{m_2} + \overrightarrow{m_2}' + \overrightarrow{m_2}'' = 0^{\lambda} m_{BC} \\ \overrightarrow{m_3} + \overrightarrow{m_3}' + \overrightarrow{m_3}'' = \$\$ \end{cases}$$

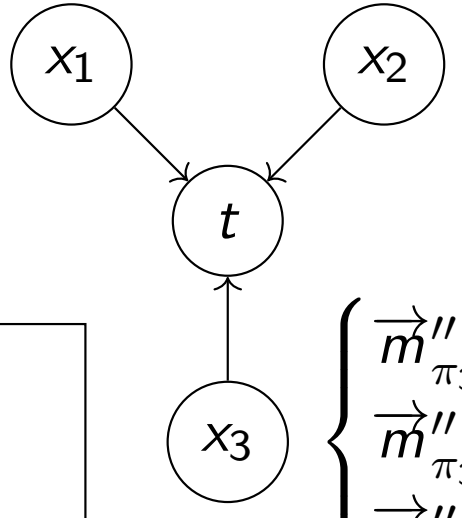
Step 3: Merging the Runs

Public:

- $\mathcal{G} = \{G_1, G_2, \ldots, G_k\}$
- st-orientations $s \to t$: $H_1, H_2, \ldots H_k$

(real graph G_2)

$$\begin{cases} \overrightarrow{m}_{\pi_1(1)} \\ \overrightarrow{m}_{\pi_1(2)} \\ \overrightarrow{m}_{\pi_1(3)} \end{cases}$$



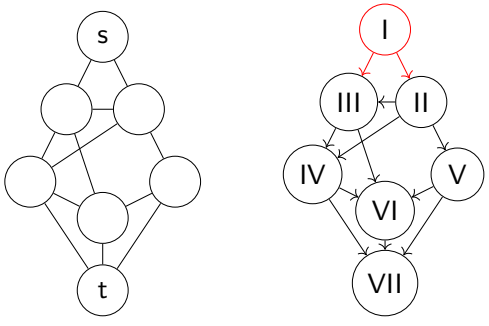
$$\begin{cases} \overrightarrow{m}'_{\pi_2(1)} \\ \overrightarrow{m}'_{\pi_2(2)} \\ \overrightarrow{m}'_{\pi_2(3)} \end{cases}$$

$$\begin{cases} \overrightarrow{m}''_{\pi_3(1)} \\ \overrightarrow{m}''_{\pi_3(2)} \\ \overrightarrow{m}''_{\pi_3(3)} \end{cases}$$

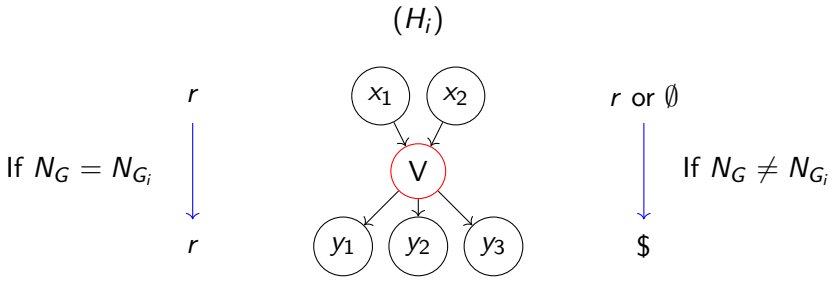
$$\begin{cases} \overrightarrow{m}_{\pi_1(1)} + \overrightarrow{m}'_{\pi_2(1)} + \overrightarrow{m}''_{\pi_3(1)} = \$ \\ \overrightarrow{m}_{\pi_1(2)} + \overrightarrow{m}'_{\pi_2(2)} + \overrightarrow{m}''_{\pi_3(2)} = 0^{\lambda} m_{BC} \\ \overrightarrow{m}_{\pi_1(3)} + \overrightarrow{m}'_{\pi_2(3)} + \overrightarrow{m}''_{\pi_3(3)} = \$\$ \end{cases}$$

Information-Theoretic Protocol – Summary

1: Single-Graph SMT

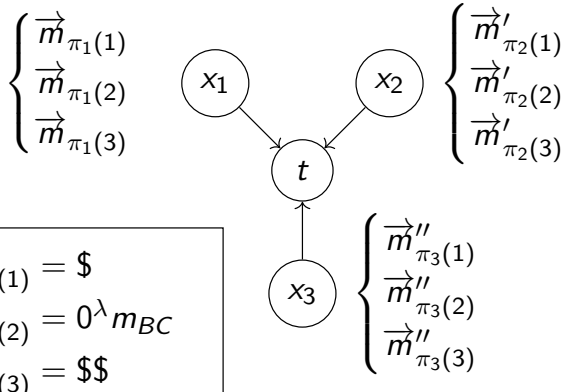


2: Censored Parallel Runs



3: Merging the Runs

(real graph G_2)



$$\begin{cases} \overrightarrow{m}_{\pi_1(1)} + \overrightarrow{m}'_{\pi_2(1)} + \overrightarrow{m}''_{\pi_3(1)} = \$ \\ \overrightarrow{m}_{\pi_1(2)} + \overrightarrow{m}'_{\pi_2(2)} + \overrightarrow{m}''_{\pi_3(2)} = 0^{\lambda} m_{BC} \\ \overrightarrow{m}_{\pi_1(3)} + \overrightarrow{m}'_{\pi_2(3)} + \overrightarrow{m}''_{\pi_3(3)} = \$\$ \end{cases}$$

1-THB

	IT	2-Connectivity	KA
	All 2-connected graphs + 2-paths		All graphs
Previously:	Cycles of fixed length		Paths of length four

1-THAB

	IT	2-Connectivity	KA	Honest Majority	OT
	All 2-connected graphs + 2-paths		All graphs with ≥ 3 nodes		All graphs
More Precisely:			Anything not 2-connected		Paths of length 2,3

Thank You!

Previous Works

TCC'15 **Topology-Hiding Computation**
Moran, Orlov, Richelson

Crypto'16 **Network-Hiding Communication and Applications to Multi-party Protocols**
Hirt, Maurer, Tschudi, Zikas

Eurocrypt'17 **Topology-Hiding Computation Beyond Logarithmic Diameter**
Akavia, Moran

Crypto'17 **Topology-Hiding Computation on All Graphs**
Akavia, LaVigne, Moran

Eurocrypt'18 **Exploring the Boundaries of Topology-Hiding Computation**
Ball, Boyle, Malkin, Moran

TCC'18 **Topology-Hiding Computation Beyond Semi-Honest Adversaries**
LaVigne, Liu-Zhang, Maurer, Mularczyck, Tschudi

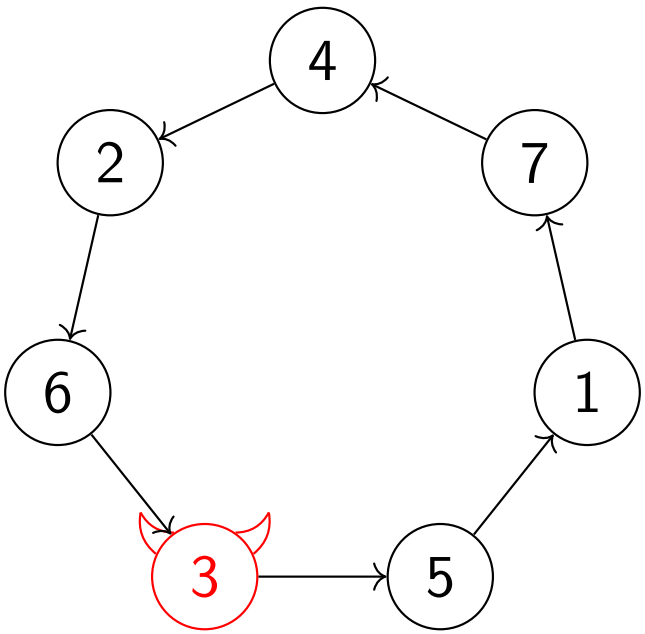
TCC'19 **Is Information-Theoretic Topology-Hiding Computation Possible?**
Ball, Boyle, Cohen, Malkin, Moran

PKC'20 **Topology-Hiding Computation for Networks with Unknown Delays**
LaVigne, Liu-Zhang, Maurer, Moran, Mularczyck, Tschudi

Simplest Protocol – Not Topology-Hiding

Secure Sum:

$$\begin{cases} \texttt{Inputs}: i \text{ holds } x_i \\ \texttt{Output}: \sum_{i=1}^{7} x_i \end{cases}$$

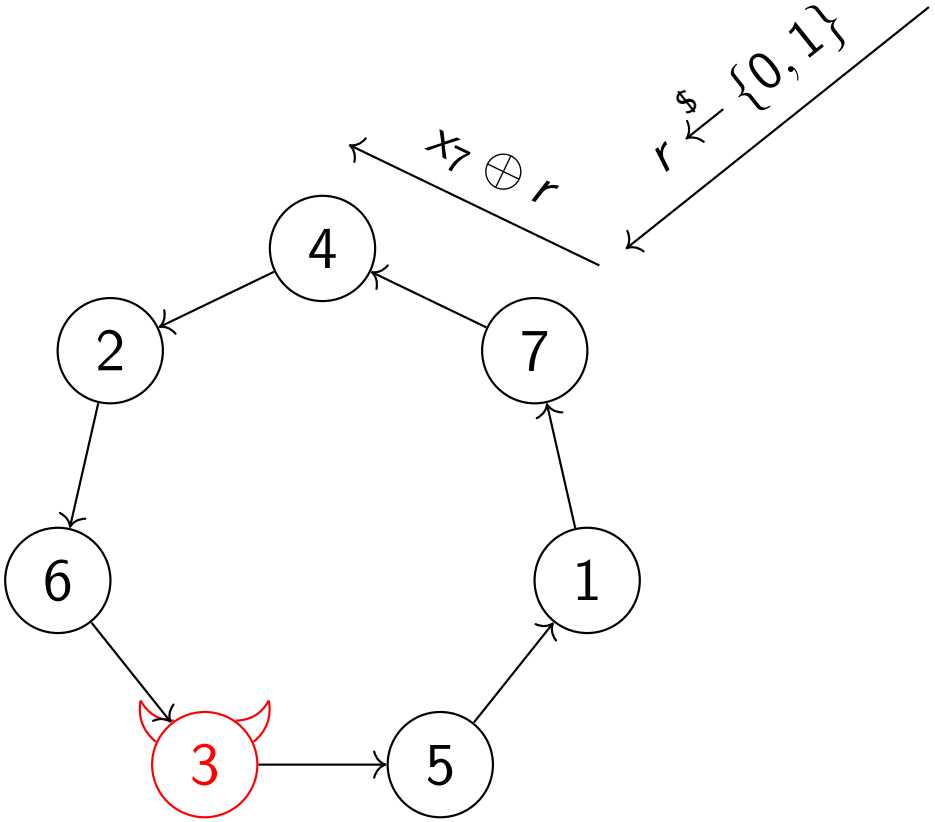


Simplest Protocol – Not Topology-Hiding

Round 1

Secure Sum:

$$\begin{cases} \texttt{Inputs}: i \text{ holds } x_i \\ \texttt{Output}: \sum_{i=1}^{7} x_i \end{cases}$$

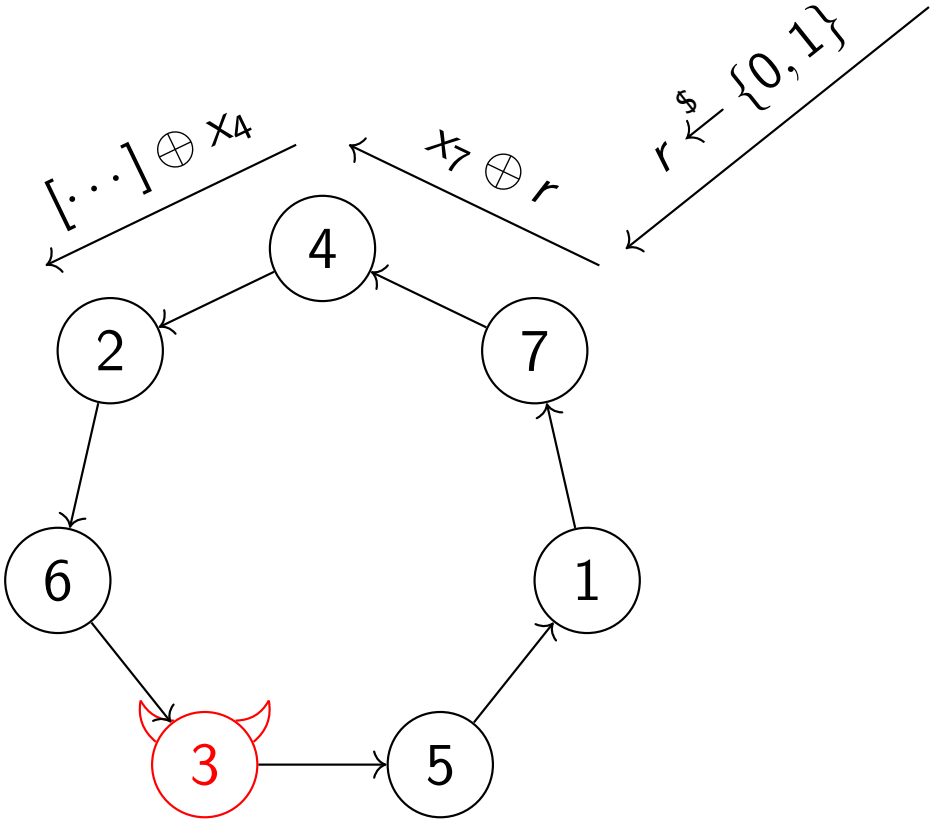


Simplest Protocol – Not Topology-Hiding

Round 2

Secure Sum:

$$\begin{cases} \texttt{Inputs}: i \text{ holds } x_i \\ \texttt{Output}: \sum_{i=1}^{7} x_i \end{cases}$$

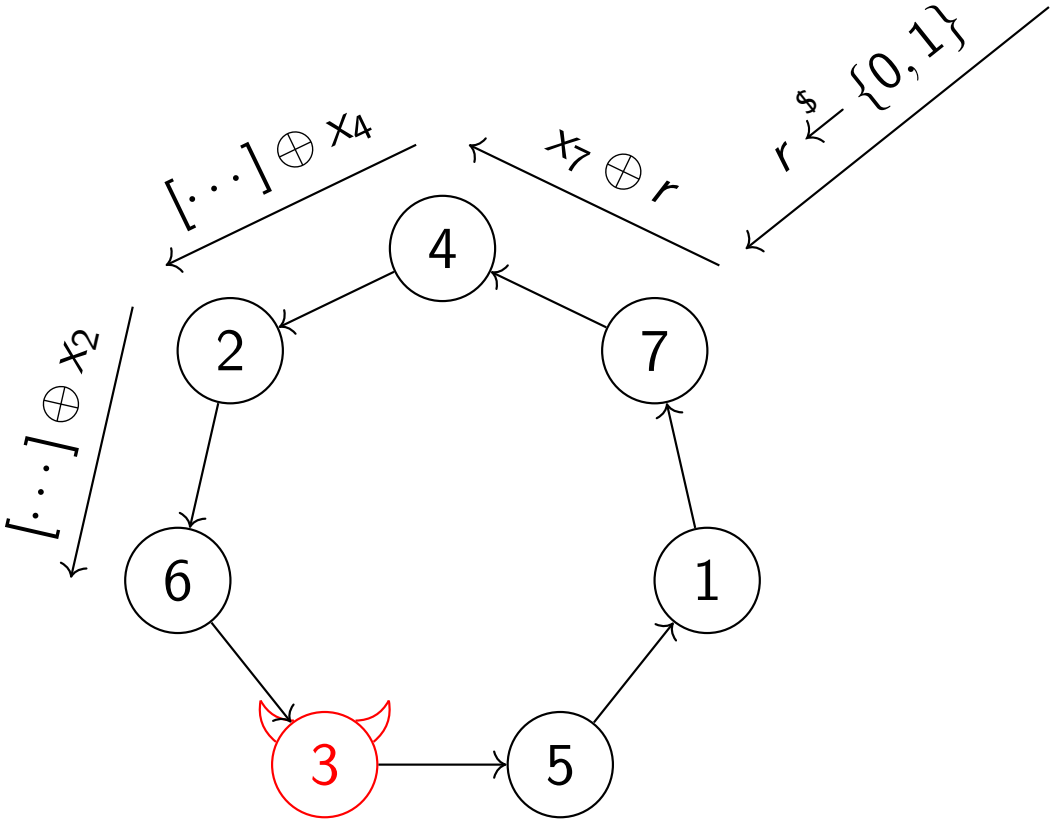


Simplest Protocol – Not Topology-Hiding

Round 3

Secure Sum:

$$\begin{cases} \texttt{Inputs}: i \text{ holds } x_i \\ \texttt{Output}: \sum_{i=1}^{7} x_i \end{cases}$$

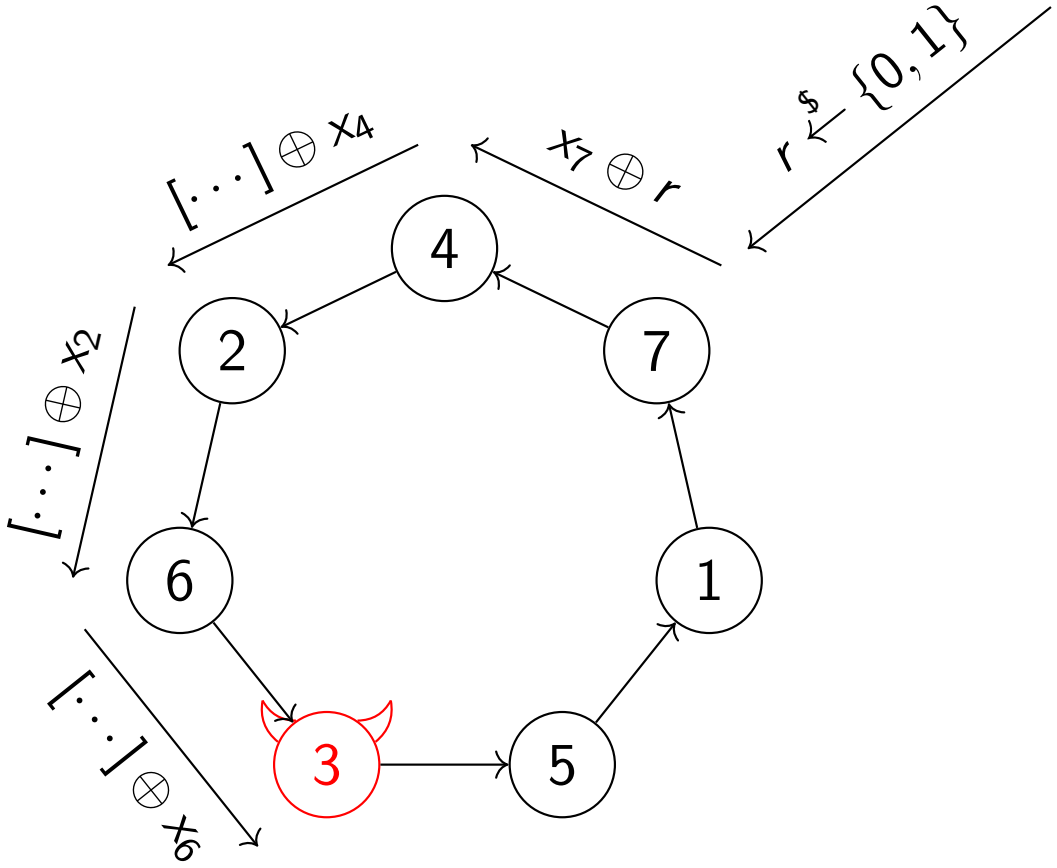


Simplest Protocol – Not Topology-Hiding

Round 4

Secure Sum:

$$\begin{cases} \texttt{Inputs}: i \text{ holds } x_i \\ \texttt{Output}: \sum_{i=1}^{7} x_i \end{cases}$$

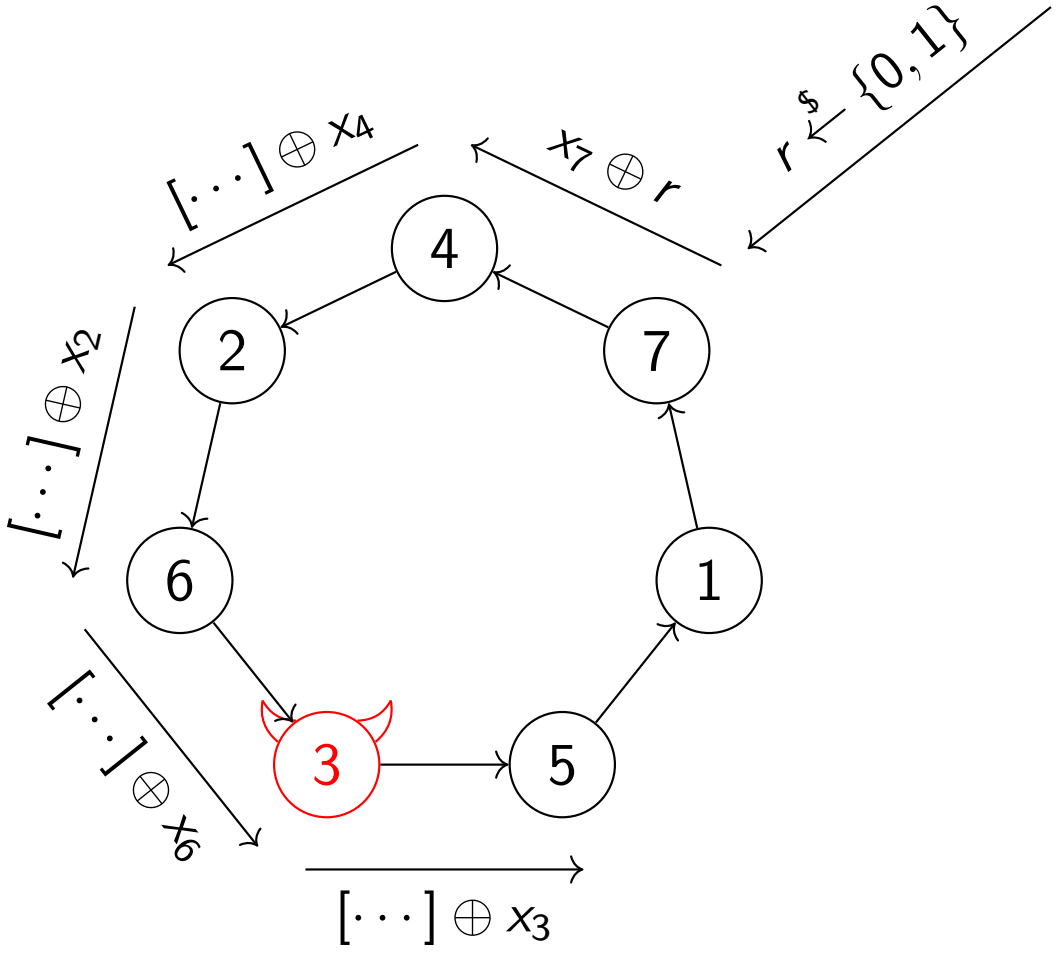


Simplest Protocol – Not Topology-Hiding

Round 5

Secure Sum:

$$\begin{cases} \texttt{Inputs}: i \text{ holds } x_i \\ \texttt{Output}: \sum_{i=1}^{7} x_i \end{cases}$$

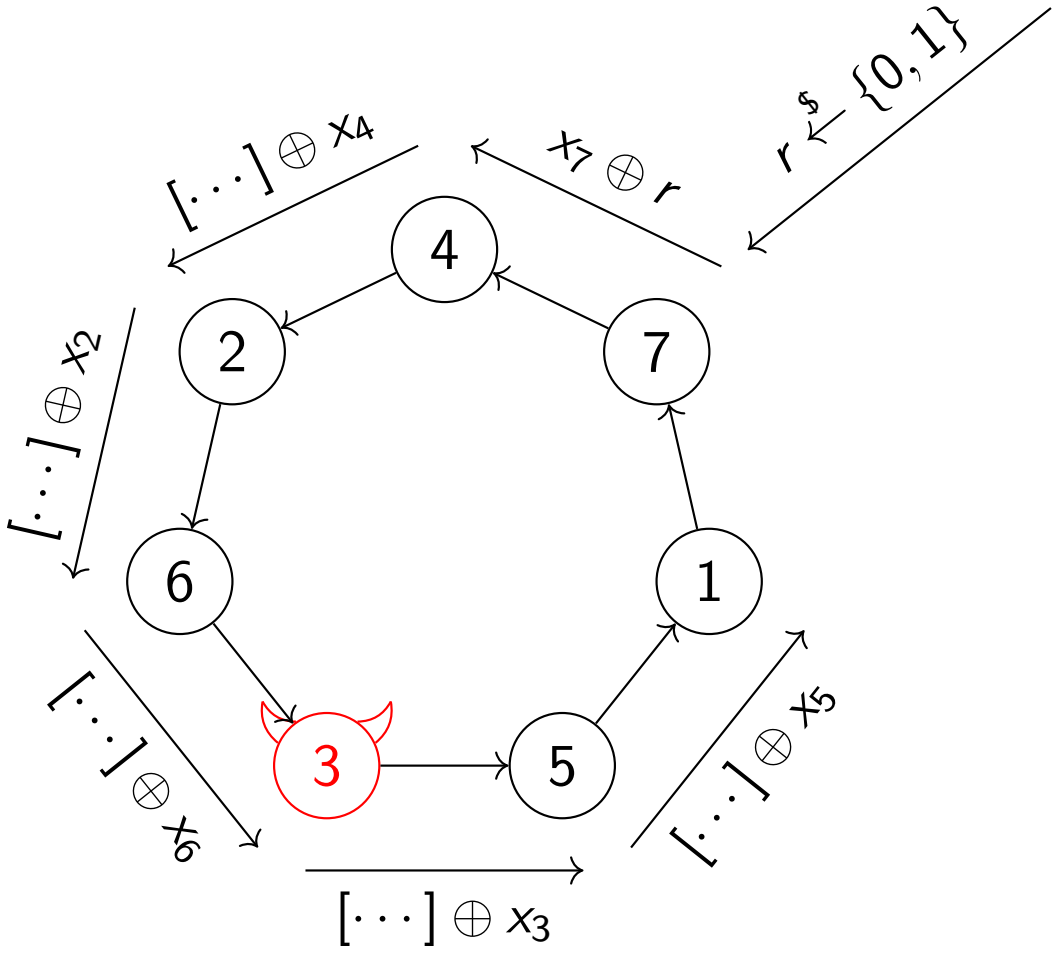


Simplest Protocol – Not Topology-Hiding

Round 6

Secure Sum:

$$\begin{cases} \texttt{Inputs}: i \text{ holds } x_i \\ \texttt{Output}: \sum_{i=1}^{7} x_i \end{cases}$$

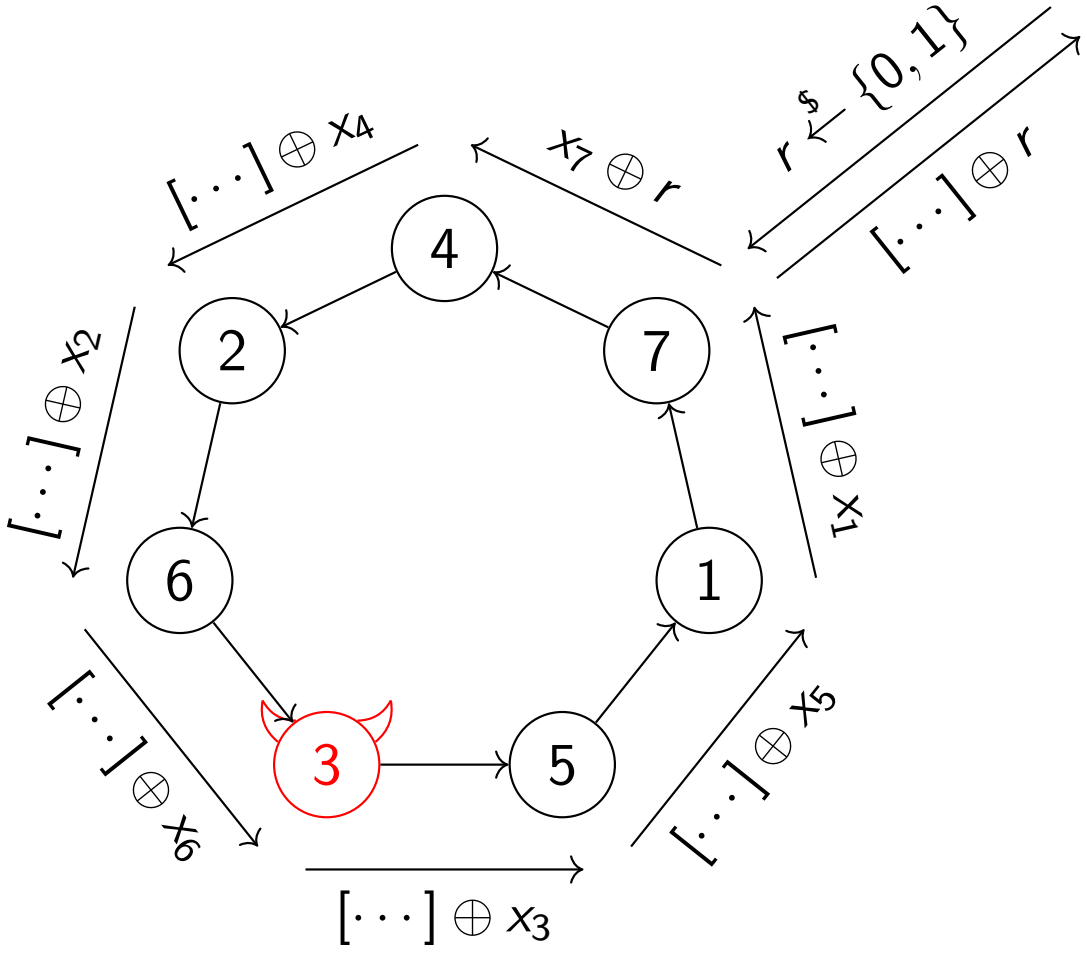


Simplest Protocol – Not Topology-Hiding

Round 7

Secure Sum:

$$\begin{cases} \texttt{Inputs}: i \text{ holds } x_i \\ \texttt{Output}: \sum_{i=1}^{7} x_i \end{cases}$$

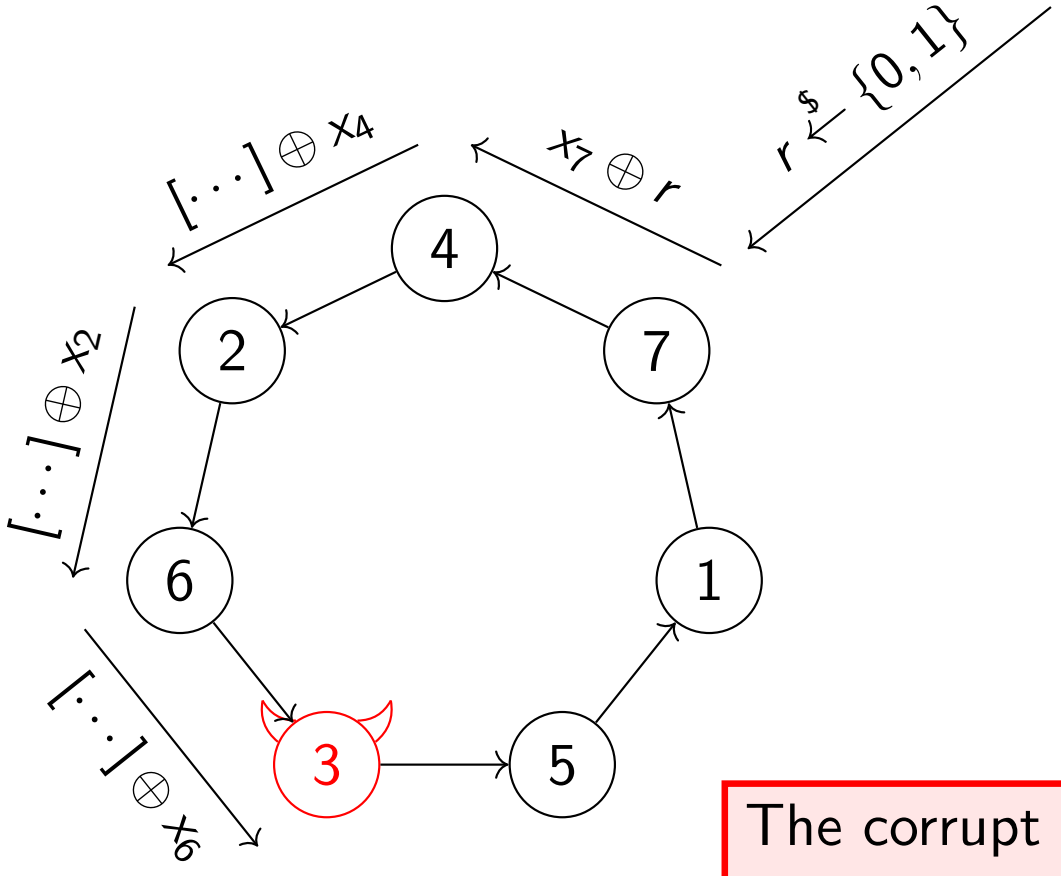


Simplest Protocol – Not Topology-Hiding

Round 4

Secure Sum:

$$\begin{cases} \texttt{Inputs}: i \text{ holds } x_i \\ \texttt{Output}: \sum_{i=1}^{7} x_i \end{cases}$$

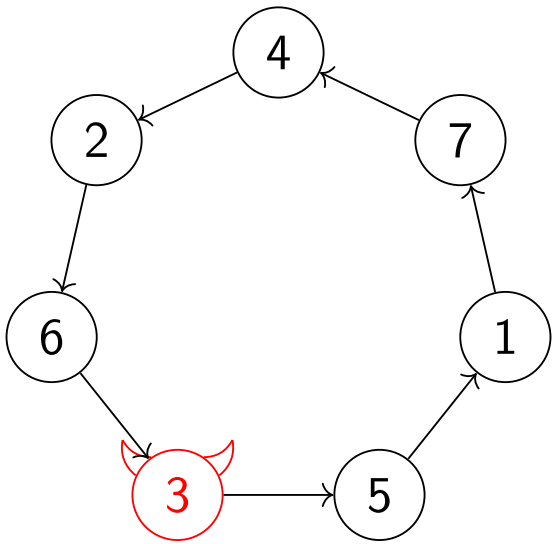


The corrupt node learns distance to initiator (7)

Simplest Protocol – Topology-Hiding

Secure Sum:

$$\begin{cases} \texttt{Inputs}: i \text{ holds } x_i \\ \texttt{Output}: \sum_{i=1}^{7} x_i \end{cases}$$

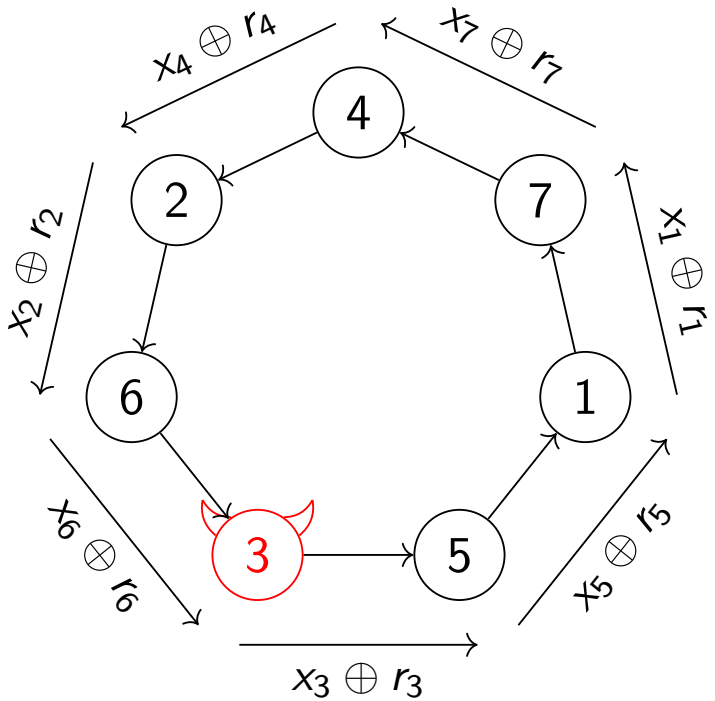


Simplest Protocol – Topology-Hiding

Round 1

Secure Sum:

$$\begin{cases} \texttt{Inputs}: i \text{ holds } x_i \\ \texttt{Output}: \sum_{i=1}^{7} x_i \end{cases}$$



Simplest Protocol – Topology-Hiding

Round 2-7

Secure Sum:

$$\begin{cases} \texttt{Inputs}: i \text{ holds } x_i \\ \texttt{Output}: \sum_{i=1}^{7} x_i \end{cases}$$