# Peeking into the Future

MPC Resilient to Super-Rushing Adversaries

Gilad Asharov

Anirudh Chandramouli

Ran Cohen

Yuval Ishai





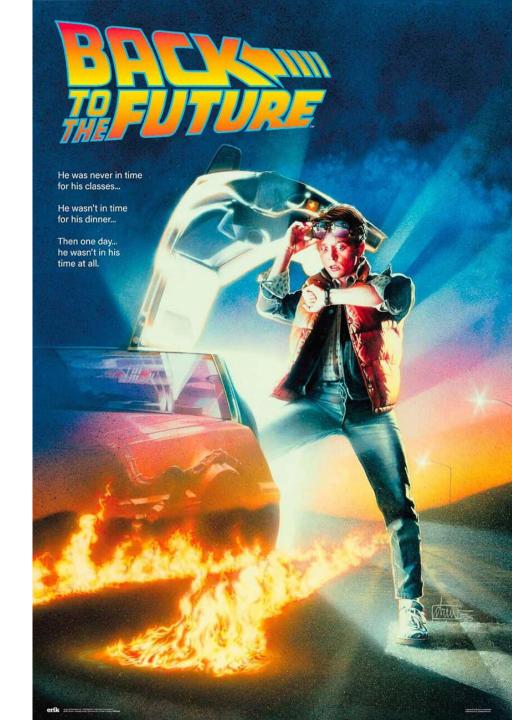




Eurocrypt 2025



"Well, hey, Doc, what's the harm in bringing back a little info on the future?
You know, maybe we could place a couple bets"



#### Biff's Attack on the Timeline

1955 2015



Gives it to his past self



Biff gets rich!



Biff steals the almanac

#### The Weather

The state of the same of same of the same

# Hill Halley Telegraph

Vol. XVII. No. 30

PUBLISHED DAILY

THE RESIDENCE AND ASSESSED.

PROCES IN CICYIN

#### BIFF TANNEN

# **Luckiest Man On Earth**

PRICE CREMING.

A supportion that public bearings on approxime to limited to see your Ma possible was taken ander ad-COMPARED BY SEC POSSISSALIVE.

Matter memorial fluor art three plage that some input within in Darthorning but it now buseeings, regressed it new budge, World Sheet, is pressure from the insole which your profession strengt the alpest of the case,

Of an law importance was the summer surgerises sheen of the fact that my mesonal freeze without to the person of our perilipsels over same, will not an until therefore. present to a relief for moectation and reogaration. This man extended in the inabromenia sabiptati by Macron Directors.

Thur at him condetence all per groundshelp freed their selfred the americanism agreem head requiring this technicaling Attangunation for Smooth with questions and deprive between the objection trens Norther studented.

A suggested that puttle American on applications in Smithed by one govern Ata months over behalf toward Ach. themself, by the presentation,

As indeed the remedigation NO ADDITION BALL DWINGS ON that some son right will be wheel on the principal to the near Tulipro, Alabable Belly



BIFF TANNEN

Frield Statement Wiles. Schiller, Was, Science of Files Consumpt State

#### State Likely to Start Payroll Tax in 1960

the customer verygoline. storm of the last that less mention of Totals, willflushed his filled policy of the continuety toppersonally be a subtilety that make

#### Nasser Accuses Reds of Plotting His Overthrow

new point Corvers, howard. what is housed will be a distanceproperties assumed every far more hard, become known. the distance of the consulties

The Safe InputStar Novell. MATCH STREETS WAS ADDRESS. POLICE the polysytten, Between conterrible like action have been a three broughed study with the Street The To Service Street,

it would appear that the professorie impaint total time inside his to fail oil abidsep of the telese differences princing from the advances, but retire him approximat the from of these perferences for Store book smolesment by Mar. Senta Co.

"Many painting had all this Shape St. Af. Acres 1995 Acres & Staffunnsting this it were be-Store in grouppe from the in-STEE NAME AND PROPERTY.

soline ortoin the roses, shale De performed Dately on In mostley of hore, notes at the combing the solding have been exchange. This defeats have prior a professional revealings. continues according the final time time to be feet that such has made for though been been a more decided obey will flu-

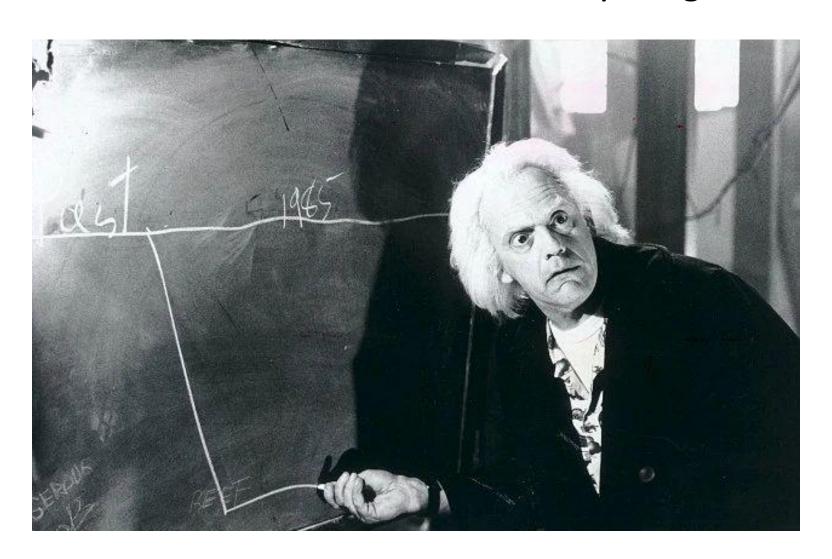
Thus an this rischessors all to alone the local supreme , and pronouncing local form. makes become the production of making the supplement paper. THREE PROPERTY THE RESIDENCE. ing. Attachgary exts. for chickey. will previous and dispute ferrous. Buy impaiding many Suffrer Depoints.

The Mason, preservising here. STATISTICAL PROPERTY AND ADDRESS. title on bright his popular, Balances HE CITY MIGH CONSUME THAT ADfor the course I had Dangbard, No. pulsate startings presenting the stone, that the Manur ted Sale as public stutched to Mining and other Real Day.

As Sadrock by Wearington for assessment made menticulations are digt some new right; and he Mad no the attactors in the NAME THROWN, AVAILABLE PARK hardle improc. hot. purpositions Sed Wat Can hill dipose wine decade of lumining hit at

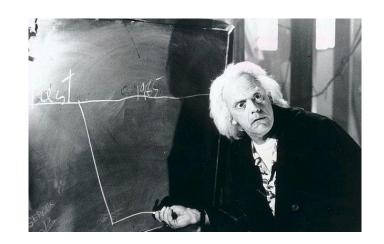
March parents had at \$500.

"No, Marty, we've already agreed that having information about the future could be extremely dangerous!"



#### This Work

"Back to the Future" attacks on MPC



- Optimistic implementations of certain synchronous MPC protocols may be vulnerable
- Goal: understand what makes a protocol immuned to such attacks (enable optimistic implementations)

#### Communication Models for MPC

#### **Fully asynchronous**

Adversarial message delivery
 (can drop messages)

- Most UC secure MPC
- No guaranteed termination

# Asynchronous with eventual delivery

- Every message eventually arrives
- Guaranteed termination
- No "input completeness"
- Inherent t < n/3
- Same limitations for partial synchrony

#### **Synchronous**

- Round-by-round, potentially with broadcast
- Guaranteed termination
- Input completeness
- Guaranteed output delivery for t < n/2 (sometimes t < n)
- Vast majority of literature

#### Communication Models for MPC

#### Fully asynchronous

- Adversarial message delivery
   (can drop messages)
- Most UC secure MPC
- No guaranteed termination

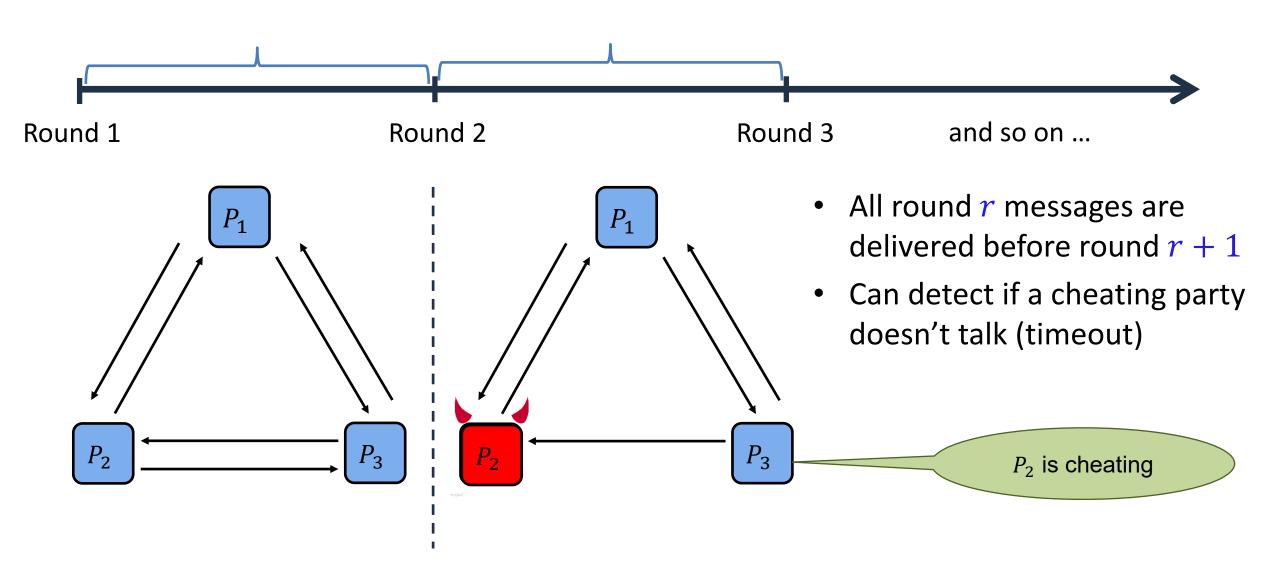
# Asynchronous with eventual delivery

- Every message eventually arrives
- Guaranteed termination
- No "input completeness"
- Inherent t < n/3
- Same limitations for partial synchrony

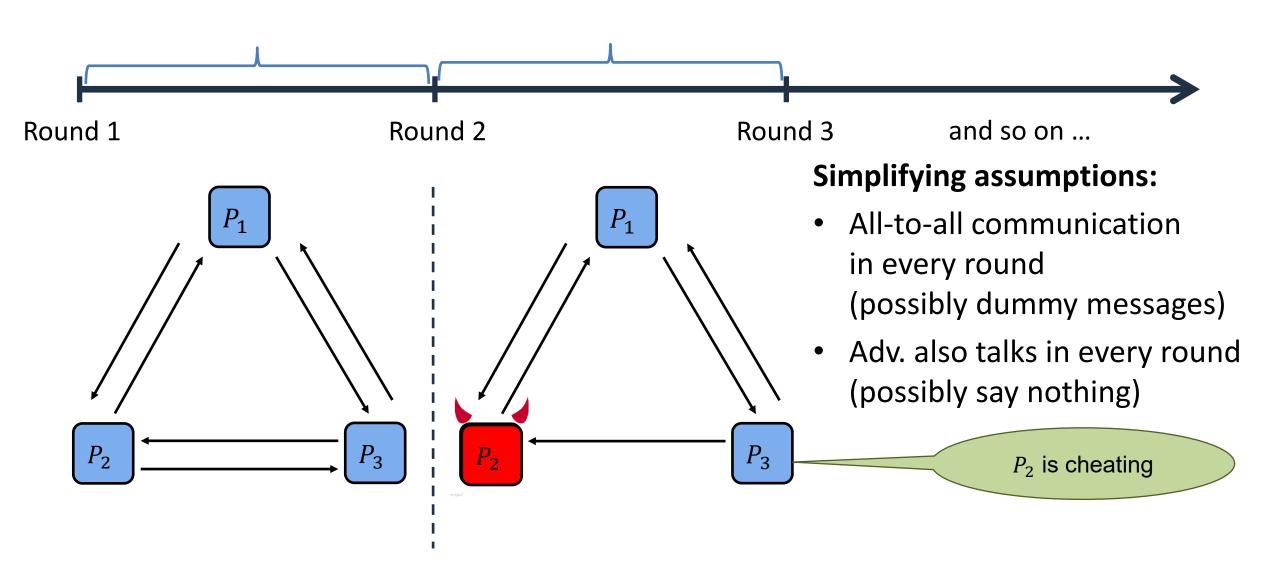
#### **Synchronous**

- Round-by-round, potentially with broadcast
- Guaranteed termination
- Input completeness
- Guaranteed output delivery for t < n/2 (sometimes t < n)
- Vast majority of literature

# Synchronous Protocols



# Synchronous Protocols

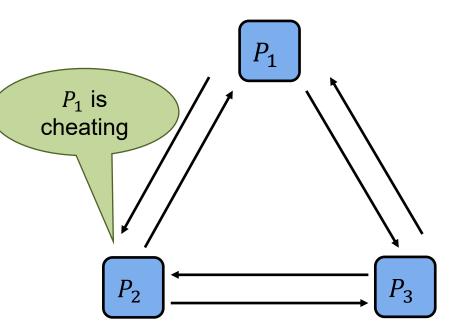


#### How much time should we wait?



Say the expected duration is 1 second



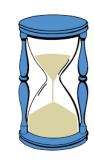


#### **Idea #1:**

- Set round duration to 2 seconds
- But...
   honest parties might be falsely detected as cheaters

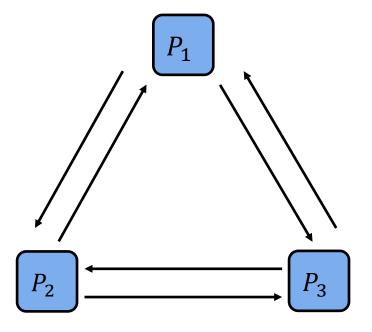


#### How much time should we wait?



Say the expected duration is 1 second





#### Idea #2:

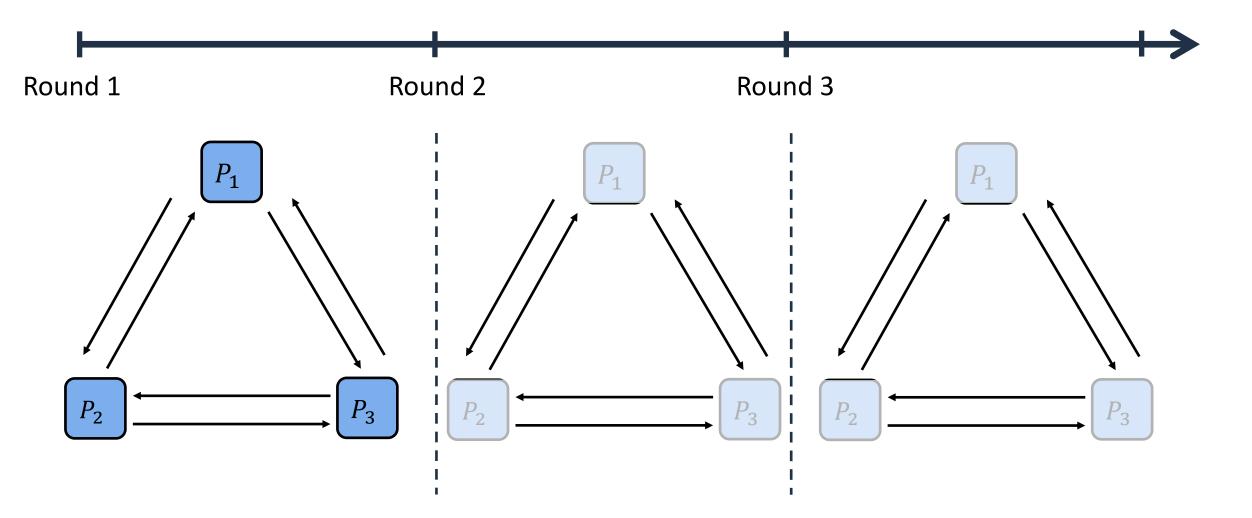
- Set round duration to 1 hour
- No party falsely accused
- But...
   who's gonna use my protocol





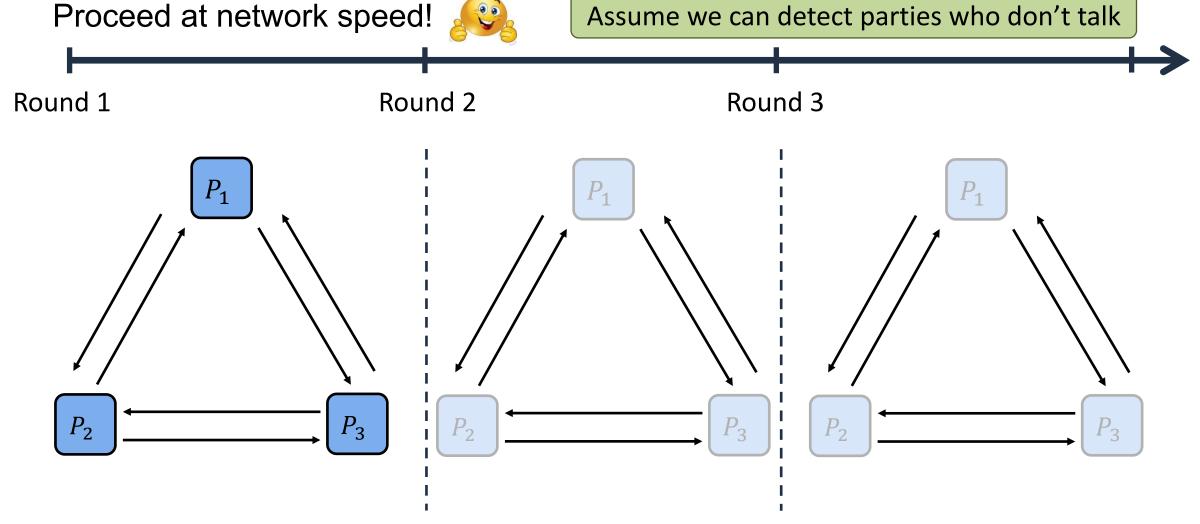
#### How much time should we wait?

Proceed optimistically: once all round *r* message arrive

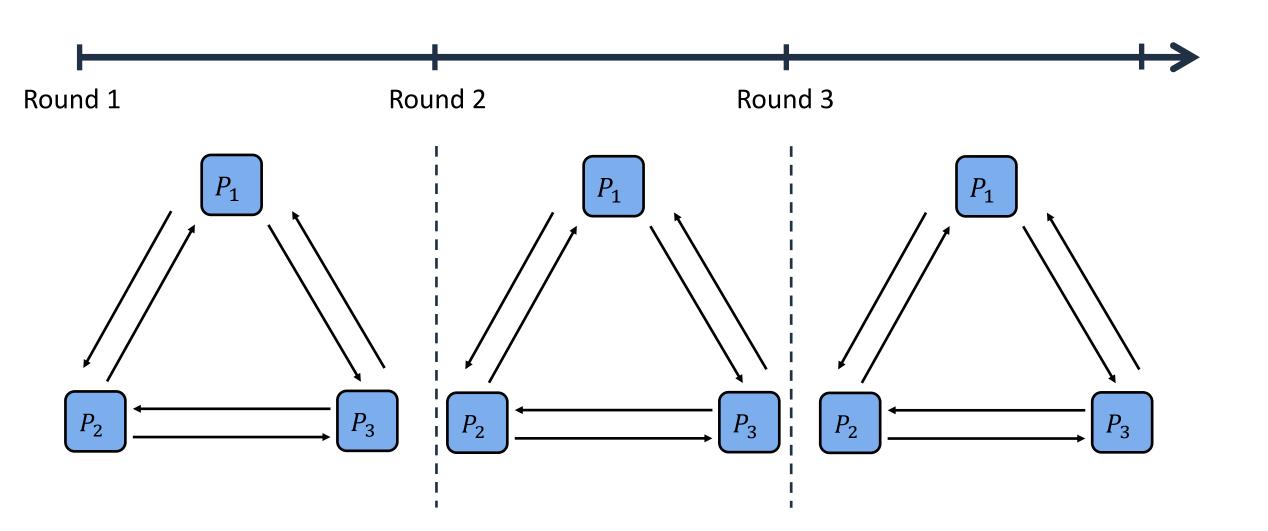


Proceed optimistically: once all round *r* message arrive

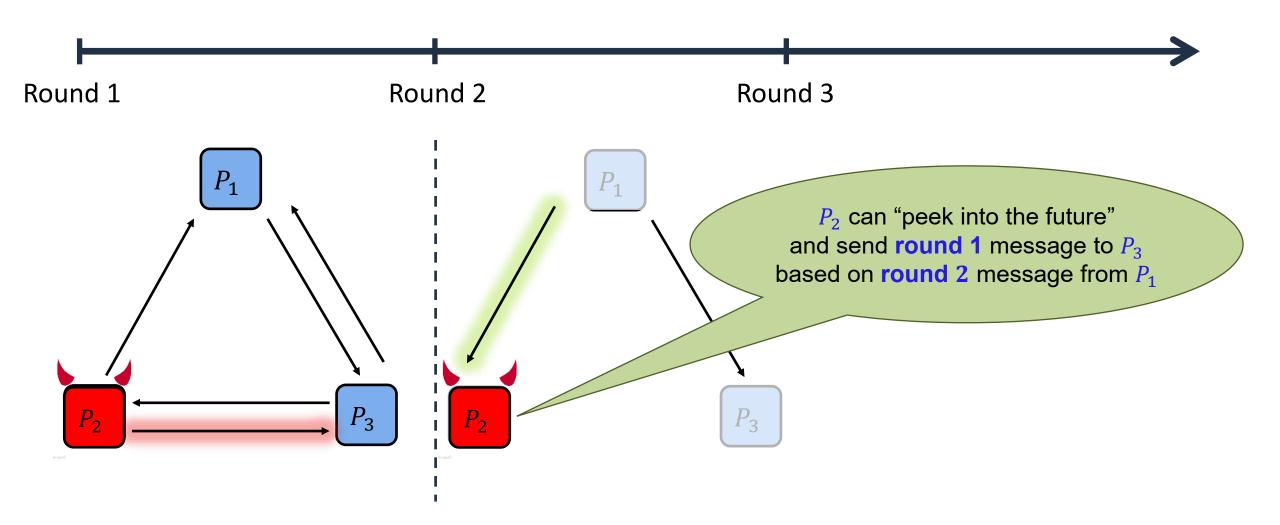




## Wait... What???



### Wait... What???



# Peeking ⇒ Super-Rushing

Non-Rushing

Adversary sends round-r messages **before** receiving the honest parties' round-r messages

Rushing

Adversary can send round-r messages **after** receiving the honest parties' round-r messages

Super-Rushing

Adversary can send round-r messages **after** receiving some round-(r + 1) messages

# A Gap in the Security Analysis

**Applied research** 

Theory (ideal synchrony)

**Practice** 

(optimistic implementations)



Rushing

Super-Rushing

Is it really a meaningful attack?

Are existing synchronous MPC protocols vulnerable to super-rushing attacks?

## A Gap in the Security Analysis

Applied research

**Practice** 

(optimistic implementations)



Rushing

Theory

(ideal synchrony)

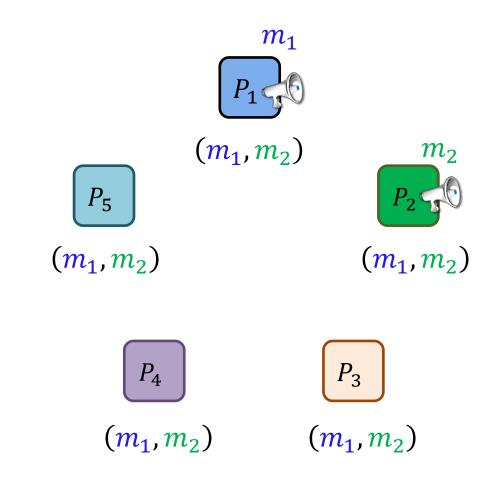
Super-Rushing

Yes!

Some protocols are insecure against super-rushing adversaries

#### Simultaneous Broadcast [CGMA85]

- 5 parties, 2 senders
- $P_1$  holds  $m_1$  and  $P_2$  holds  $m_2$
- Everyone outputs  $(m_1, m_2)$
- Security against 1 corruption
- $P_1$  cannot choose  $m_1$  as a function of  $m_2$  (and vice versa)

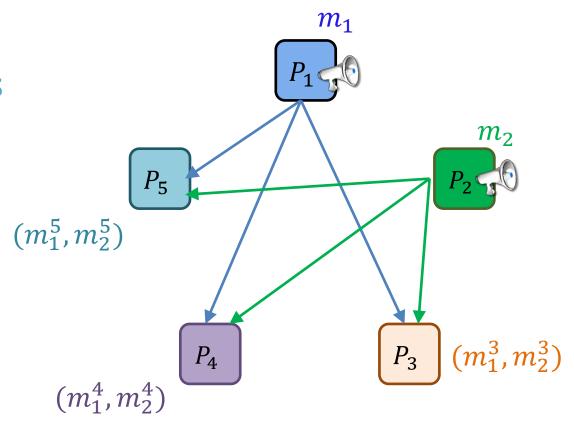


## A Simple Simultaneous Broadcast Protocol [GIKR02]

5 parties, 2 senders, 1 corruption

• Round 1:

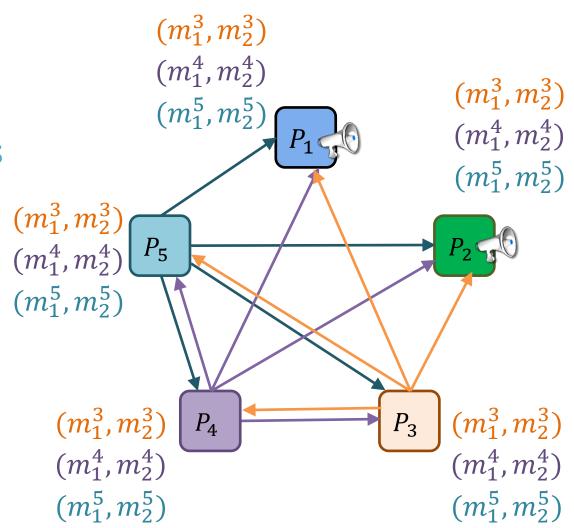
 $P_1$  and  $P_2$  send input message to  $P_3$ ,  $P_4$ ,  $P_5$ 



# A Simple Simultaneous Broadcast Protocol [GIKR02]

5 parties, 2 senders, 1 corruption

- Round 1:  $P_1$  and  $P_2$  send input message to  $P_3$ ,  $P_4$ ,  $P_5$
- Round 2:  $P_3, P_4, P_5$  echo message to everyone



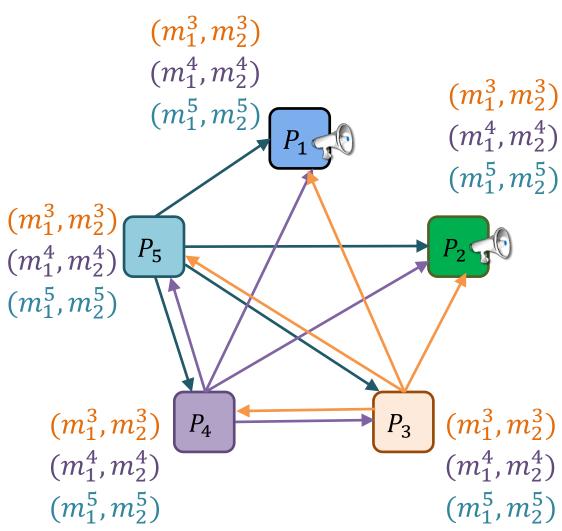
## A Simple Simultaneous Broadcast Protocol [GIKR02]

5 parties, 2 senders, 1 corruption

- Round 1:  $P_1$  and  $P_2$  send input message to  $P_3$ ,  $P_4$ ,  $P_5$
- Round 2:  $P_3$ ,  $P_4$ ,  $P_5$  echo message to everyone
- Output:  $(m'_1, m'_2)$  echoed by at least 2 parties

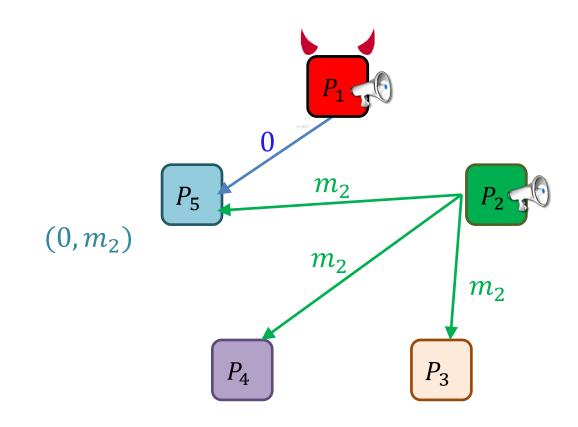
#### Security against rushing adversary

- Corrupt sender: independent message
- Corrupt non-sender: cannot affect majority



- Attack: corrupted P<sub>1</sub>
- Round 1:

```
P_2 sends m_2 to P_3, P_4, P_5
P_1 send 0 only to P_5
```

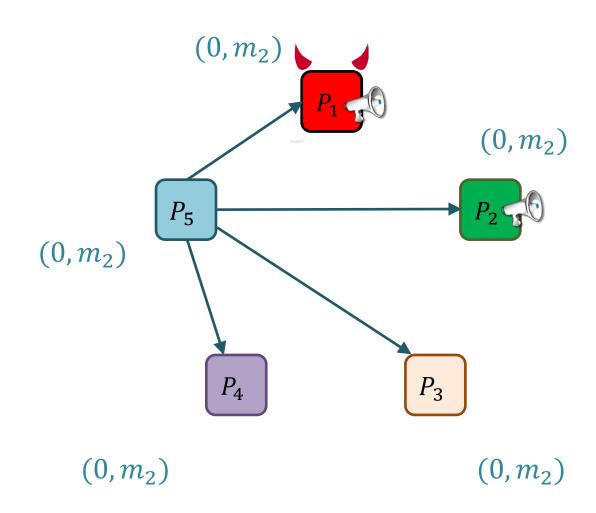


- Attack: corrupted P<sub>1</sub>
- Round 1:

```
P_2 sends m_2 to P_3, P_4, P_5
P_1 sends 0 only to P_5
```

• Round 2:

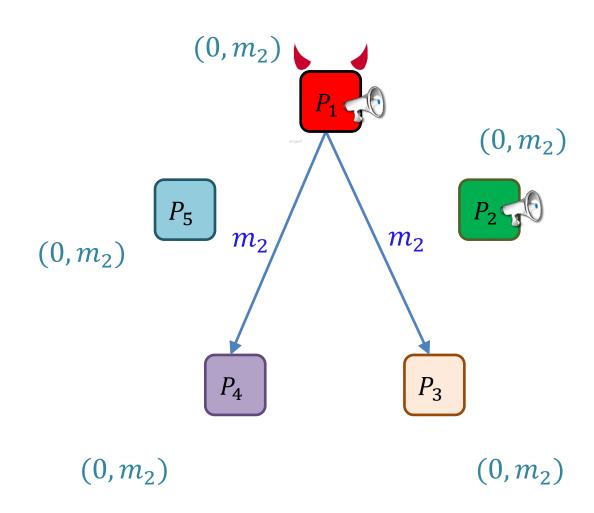
```
P_5 echos (0, m_2) to P_1, P_2, P_3, P_4
```



- Attack: corrupted P<sub>1</sub>
- Round 1:

```
P_2 sends m_2 to P_3, P_4, P_5
P_1 sends 0 only to P_5
```

- Round 2:
  - $P_5$  echos  $(0, m_2)$  to  $P_1, P_2, P_3, P_4$
- Round 1:
  - $P_1$  sends  $m_2$  to  $P_3$ ,  $P_4$



- Attack: corrupted P<sub>1</sub>
- Round 1:

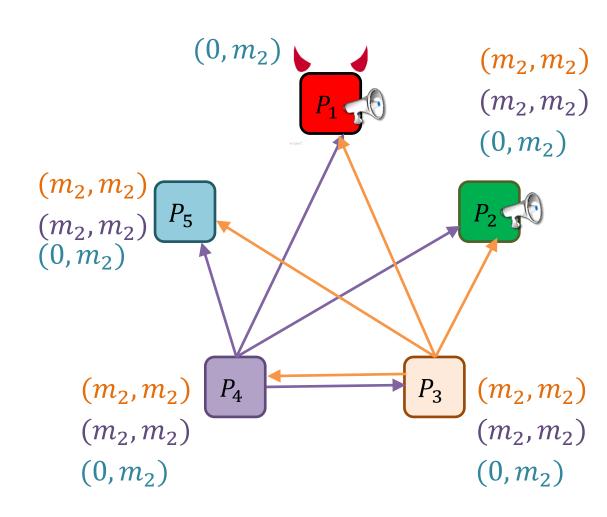
```
P_2 sends m_2 to P_3, P_4, P_5
```

- $P_1$  sends 0 only to  $P_5$
- Round 2:

```
P_5 echos (0, m_2) to P_1, P_2, P_3, P_4
```

- Round 1:
  - $P_1$  sends  $m_2$  to  $P_3$ ,  $P_4$
- Round 2:

```
P_3, P_4 echo (m_2, m_2) to everyone
```



 $(0, m_2)$ 

- Attack: corrupted P<sub>1</sub>
- Round 1:

```
P_2 sends m_2 to P_3, P_4, P_5
```

- $P_1$  sends 0 only to  $P_5$
- Round 2:

```
P_5 echos (0, m_2) to P_1, P_2, P_3, P_4
```

- Round 1:
  - $P_1$  sends  $m_2$  to  $P_3$ ,  $P_4$
- Round 2:

```
P_3, P_4 echo (m_2, m_2) to everyone
```

• Output: everyone outputs  $(m_2, m_2)$ 

```
(0, m_2) (m_2, m_2) (m_2, m_2) (0, m_2) (m_2, m_2) (0, m_2)
```

```
(m_2, m_2) P_4 P_3 (m_2, m_2) (m_2, m_2) (0, m_2) (0, m_2)
```

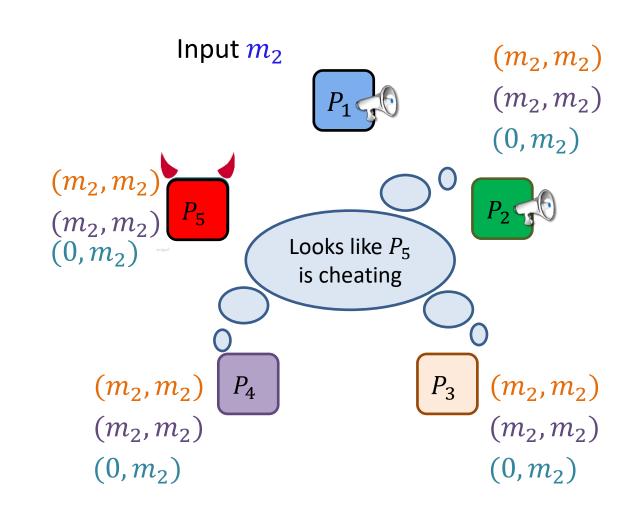
- Attack: corrupted P<sub>1</sub>
- Round 1:

```
P_2 sends m_2 to P_3, P_4, P_5
```

- $P_1$  sends 0 only to  $P_5$
- Round 2:

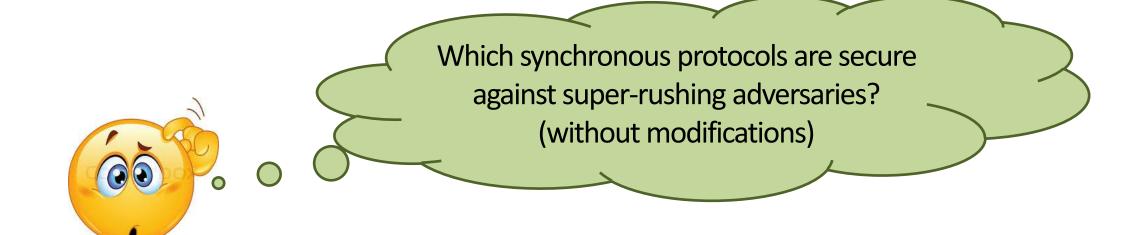
$$P_5$$
 echos  $(0, m_2)$  to  $P_1, P_2, P_3, P_4$ 

- Round 1:
  - $P_1$  sends  $m_2$  to  $P_3$ ,  $P_4$
- Round 2:
  - $P_3$ ,  $P_4$  echo  $(m_2, m_2)$  to everyone
- Output: everyone outputs  $(m_2, m_2)$



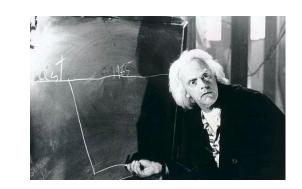
#### Our Results #1

**Theorem:** There exists a protocol (with two input providers) that is **perfectly secure** against **rushing** adversaries but is **insecure** against **super-rushing** adversaries



#### What happened in this "Back to the Future" Attack

 $P_1$  and  $P_2$  provide inputs  $P_3, P_4, P_5$  learn the output  $P_3, P_4, P_5$  reveal the output Round 1



- 1)  $P_1$  advances  $P_5$  to round 2
- $P_1$  chooses input message as a function of  $P_2$ 's input message

P<sub>1</sub> peeks into round-2 ( $P_5$ 's round-2 message) & learns  $P_2$ 's input message

Super-rushing breaks input independence

What if only one party provides input? (Broadcast, VSS, etc.)



#### Our Results #2

For perfectly secure MPC with one input provider

**Super-Rushing** ≡ Rushing ≡ Non-Rushing

**Theorem:** every protocol with a single input provider that is perfectly secure against **non-rushing** adversaries is also perfectly secure against **super-rushing** adversaries

Till now we worked too hard to show too little!!



## The Story So Far (Perfect Security)

- ✓ Single Input Provider: Super-Rushing ≡ Rushing ≡ Non-Rushing
- **Two Input Providers:** ∃ a protocol for simultaneous broadcast that is secure against rushing but not against super-rushing

The protocol feels different from MPC protocols: no privacy in the first round

Parties commit to inputs nothing learned about output

Adv cannot change inputs & output is revealed

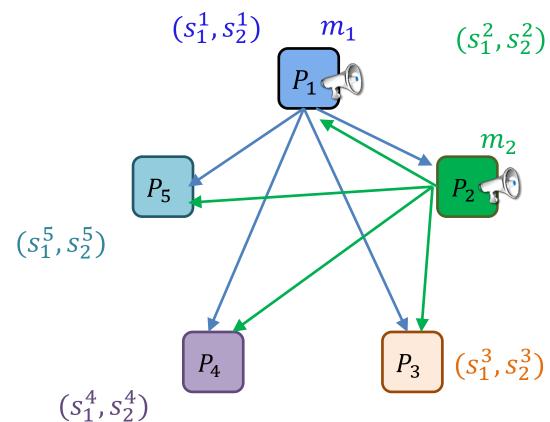
Committal Round CR

Maybe a CR prevents super-rushing attacks?

#### Simultaneous Broadcast with CR

Uses 5-party, 1-secure, 1-round VSS [GIKR01] (2 shares suffice to reconstruct)

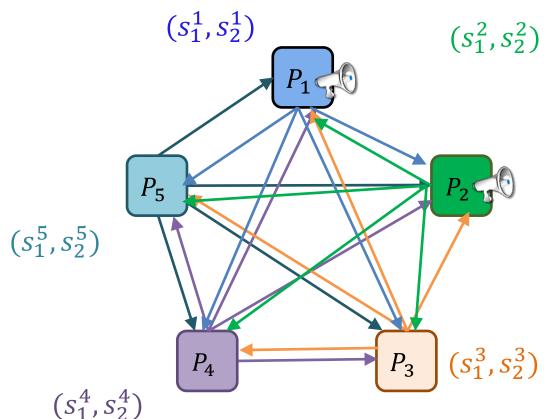
• Round 1:  $P_1$  and  $P_2$  VSS their input message



#### Simultaneous Broadcast with CR

Uses 5-party, 1-secure, 1-round VSS [GIKR01] (2 shares suffice to reconstruct)

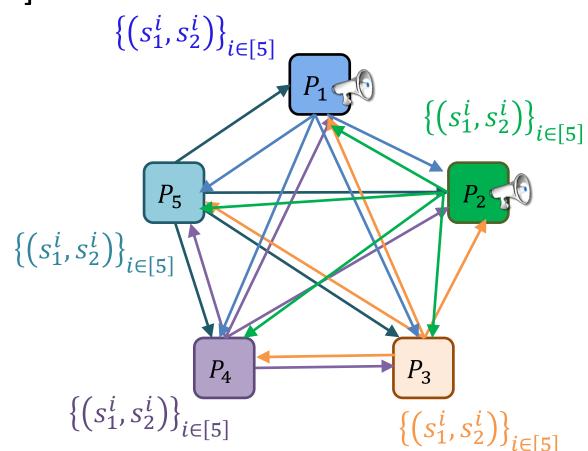
- Round 1:  $P_1$  and  $P_2$  VSS their input message
- Round 2: everyone echo their shares



#### Simultaneous Broadcast with CR

Uses 5-party, 1-secure, 1-round VSS [GIKR01] (2 shares suffice to reconstruct)

- Round 1:  $P_1$  and  $P_2$  VSS their input message
- Round 2: everyone echo their shares



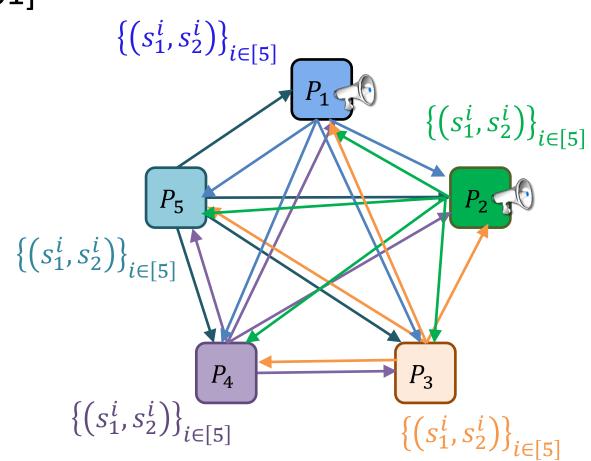
### Simultaneous Broadcast with CR

Uses 5-party, 1-secure, 1-round VSS [GIKR01] (2 shares suffice to reconstruct)

- Round 1:  $P_1$  and  $P_2$  VSS their input message
- Round 2: everyone echo their shares
- Output: reconstruct  $(m'_1, m'_2)$

#### Security against rushing adversary

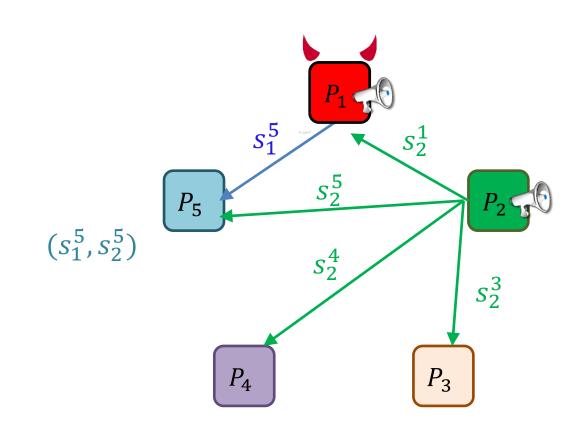
- Round 1: committal round (CR)
- Round 2: output revealing round (ORR)



- Attack: corrupted P<sub>1</sub>
- Round 1:

 $P_2$  VSS  $m_2$ 

 $P_1$  sends a random share only to  $P_5$ 



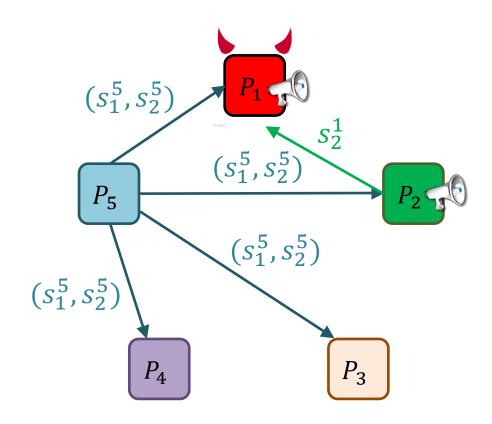
- Attack: corrupted P<sub>1</sub>
- Round 1:

 $P_2$  VSS  $m_2$ 

 $P_1$  sends a random share only to  $P_5$ 

• Round 2:

 $P_5$  echos  $(s_1^5, s_2^5)$  to  $P_1, P_2, P_3, P_4$ 



 $m_2 = \operatorname{Recon}(s_2^1, s_5^1) -$ 

- Attack: corrupted P<sub>1</sub>
- Round 1:

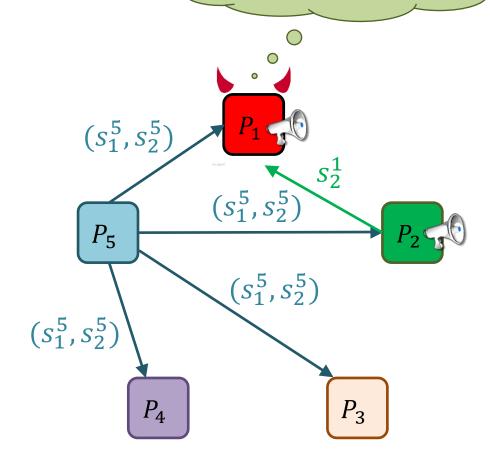
 $P_2$  VSS  $m_2$ 

 $P_1$  sends a random share only to  $P_5$ 

• Round 2:

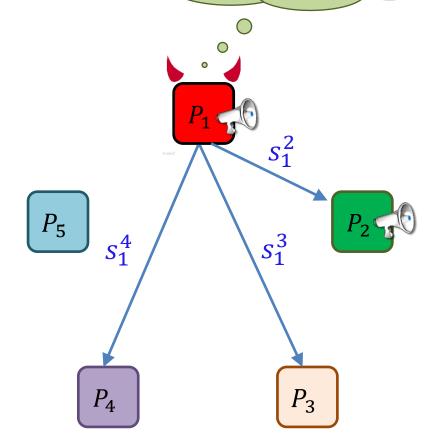
$$P_5$$
 echos  $(s_1^5, s_2^5)$  to  $P_1, P_2, P_3, P_4$ 

 $\triangleright P_1$  reconstructs  $m_2$  from  $s_2^1$  and  $s_2^5$ 

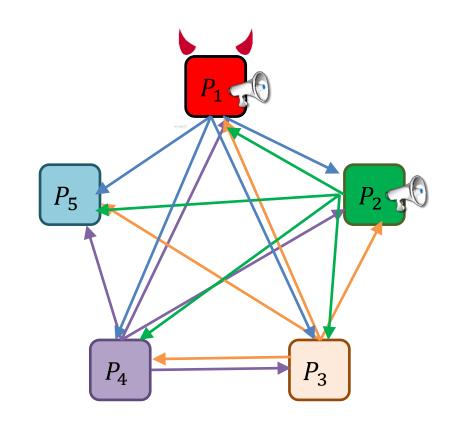


 $m_2 = \operatorname{Recon}(s_2^1, s_5^1)$ 

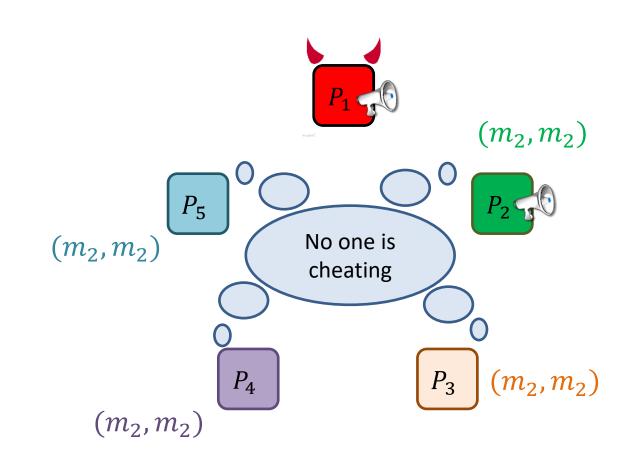
- Attack: corrupted P<sub>1</sub>
- Round 1:
  - $P_2$  VSS  $m_2$
  - $P_1$  sends a random share only to  $P_5$
- Round 2:
  - $P_5$  echos  $(s_1^5, s_2^5)$  to  $P_1, P_2, P_3, P_4$
- $ightharpoonup P_1$  reconstructs  $m_2$  from  $s_2^1$  and  $s_2^5$
- Round 1:
  - $P_1$  VSS  $m_2$  to  $P_2$ ,  $P_3$ ,  $P_4$

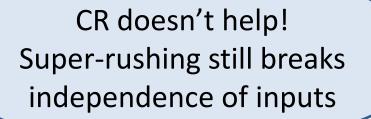


- Attack: corrupted P<sub>1</sub>
- Round 1:
  - $P_2$  VSS  $m_2$
  - $P_1$  sends a random share only to  $P_5$
- Round 2:
  - $P_5$  echos  $(s_1^5, s_2^5)$  to  $P_1, P_2, P_3, P_4$
- $ightharpoonup P_1$  reconstructs  $m_2$  from  $s_2^1$  and  $s_2^5$
- Round 1:
  - $P_1$  VSS  $m_2$  to  $P_2$ ,  $P_3$ ,  $P_4$
- Round 2:
  - $P_1$ ,  $P_2$ ,  $P_3$ ,  $P_4$  echo their shares



- Attack: corrupted P<sub>1</sub>
- Round 1:
  - $P_2$  VSS  $m_2$
  - $P_1$  sends a random share only to  $P_5$
- Round 2:
  - $P_5$  echos  $(s_1^5, s_2^5)$  to  $P_1, P_2, P_3, P_4$
- $ightharpoonup P_1$  reconstructs  $m_2$  from  $s_2^1$  and  $s_2^5$
- Round 1:
  - $P_1$  VSS  $m_2$  to  $P_2$ ,  $P_3$ ,  $P_4$
- Round 2:
  - $P_1, P_2, P_3, P_4$  echo their shares
- Output: everyone outputs  $(m_2, m_2)$



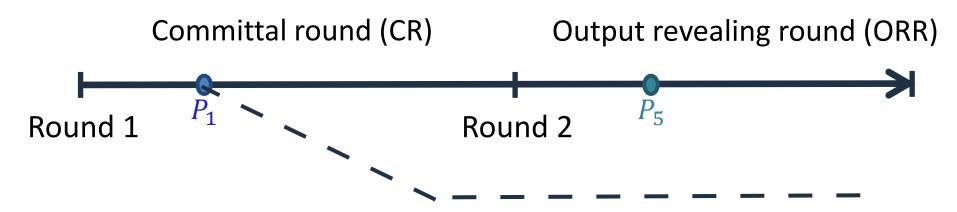




So, what are the sufficient conditions for tolerating super-rushing attacks?



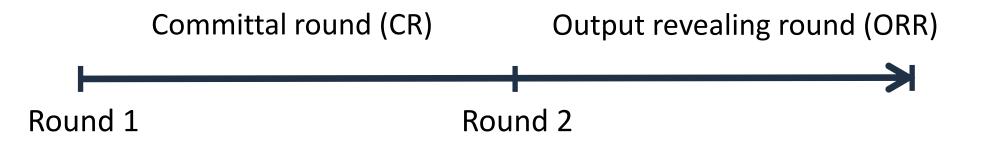
## What happened in this "Back to the Future" Attack



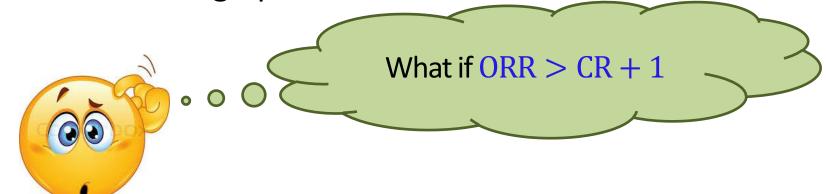
1  $P_1$  advances  $P_5$  to round 2

- 2  $P_1$  peeks into round-2 ( $P_5$ 's round-2 message) & learns  $P_2$ 's input message
- $P_1$  chooses input message as a function of  $P_2$ 's input message

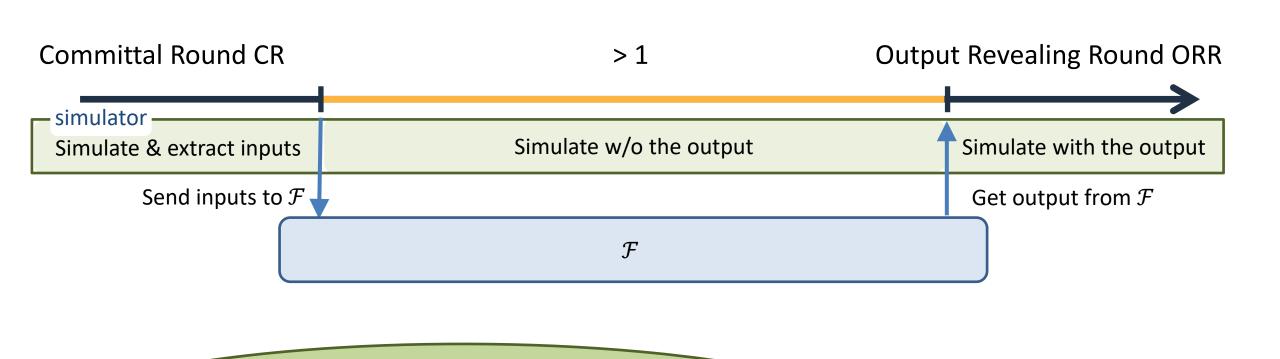
### What happened in this "Back to the Future" Attack



- Here CR = 1 and ORR = 2
- That is, ORR = CR + 1
- All-to-all communication ⇒ Peeking up to 1 round



#### **Our Sufficient Condition**



Mitigates Super-Rushing: can peek into ORR only after everyone have completed CR



#### Our Results #3

**Theorem:** every protocol that is

- 1) Perfectly secure against **rushing** adversaries \*
- 2) Has all-to-all communication
- 3) ORR > CR + 1

is also perfectly secure against super-rushing adversaries

Can we still support

ORR = CR + 1?

o simulation" (see the paper)

<sup>\*</sup> security is via "compatible simulation" (see the paper)

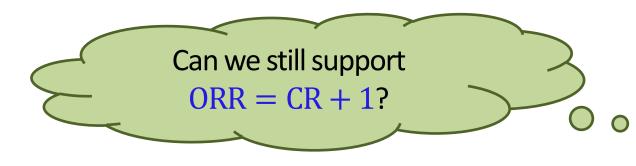
#### Our Results #3.5

**Theorem:** every protocol that is

- 1) Perfectly secure against rushing adversaries \*
- 2) Has all-to-all communication
- 3) ORR = CR + 1, but CR is over broadcast

is also perfectly secure against super-rushing adversaries

Adv cannot change its message



<sup>\*</sup> security is via "compatible simulation" (see the paper)

# Corollary

BGW is secure against super-rushing attacks!



• We have all-to-all & ORR > CR + 1

# Corollary

BGW is secure against super-rushing attacks!



- We have all-to-all & ORR > CR + 1
- What about linear functions with ORR = CR + 1?
- The VSS ends with a broadcast round
- Same for round-efficient variants [ABT19,AKP20]



#### Our Main Result

Corollary: BGW is secure against super-rushing attacks!

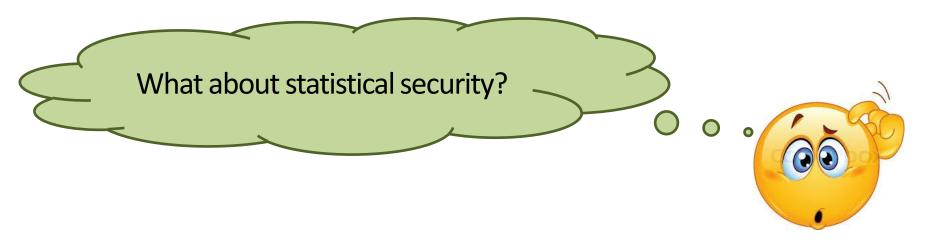
BGW be executed optimistically:

- Parties advance upon receiving messages
- Everyone talk ⇒ no need for continuous synchronization & long delays
- Timeouts only needed to detect parties who don't talk

Stronger security for free!



#### Our Results #4



**Theorem:** ∃ a protocol that is

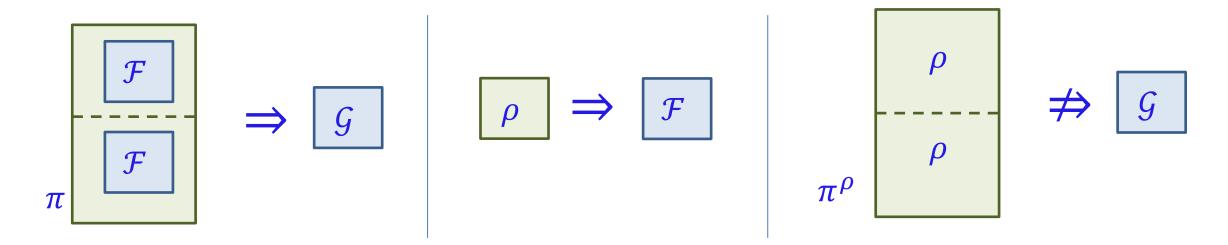
- 1) Statistically secure against rushing adversaries
- 2) Has all-to-all communication
- 3) ORR > CR + 1

But is **not** statistically secure against **super-rushing** adversaries

#### Our Results #5

Theorem: super-rushing security is not sequentially composable

- $\exists$  functionalities  $\mathcal{F}$  and  $\mathcal{G}$
- $\exists$  a protocol  $\pi$  that realizes G against super-rushing in the F-hybrid model
- $\exists$  a protocol  $\rho$  that realizes  $\mathcal{F}$  against super-rushing
- But  $\pi^{\rho}$  does not realize G against super-rushing



# The Story So Far (Perfect Security)

- ✓ Single Input Provider: Super-Rushing ≡ Rushing ≡ Non-Rushing
- **X** Two Input Providers: Super-Rushing ≠ Rushing ≠ Non-Rushing
- Committal round does not help (on its own)
- Modular analysis is tricky (no sequential composition)
- **Sufficient conditions:** Rushing ⇒ super-rushing if
  - All-to-all communication
  - ORR > CR + 1, or ORR = CR + 1 and CR over broadcast
- X This result doesn't extend to statistical security

# An Alternate Strategy

- Kushilevitz, Lindell, and Rabin [STOC '06]
  - > A generic compiler of synchronous MPC to asynchronous UC
  - In each round:
    - 1) Each party waits for all messages
    - 2) Sends OK to all
    - 3) Once receiving OK from all, advances to the next round
  - > Can be used for optimistically execute synchronous MPC
  - $\triangleright$  But  $\times$  2 round complexity and +  $O(n^2)$  communication
- This work: analyze unmodified synchronous protocols

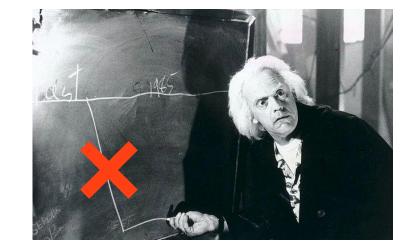
# Coming soon

- New sufficient conditions for perfect MPC
   with ORR = CR + 1 (capture [IKP10] and alike)
- Sequential composition theorem
- Capture protocols w/o all-to-all communication
  - Where communication pattern is fixed and known before each round
  - > À la [DN07, GLS19]



#### Conclusion

- Optimistic implementations may be vulnerable to "Back to the Future" attacks
- All-to-all & ORR > CR + 1 sufficient for Rushing  $\Rightarrow$  Super-Rushing



**Conjecture:** most (if not all) general purpose MPC remain secure



Thank you