Guaranteed Output in $O(\sqrt{n})$ Rounds for Round-Robin Sampling Protocols

Ran Cohen — Reichman University

Jack Doerner — Northeastern University

Yashvanth Kondi — Northeastern University

abhi shelat — Northeastern University

AARHUS UNIVERSITY

The Powers of Tau

$$\{\tau^i \cdot G\}_{i \in [d]} : \tau \leftarrow \mathbb{Z}_q$$

where $(\mathbb{G}, G, q, +)$ is an EC group

The Powers of Tau

$$\{\tau^i \cdot G\}_{i \in [d]} : \tau \leftarrow \mathbb{Z}_q$$

where $(\mathbb{G}, G, q, +)$ is an EC group

Structured Reference String for Polynomial Commitments,
introduced by [Kate-Zaverucha-Goldberg '10]
Related to Strong Diffie-Hellman [Boneh-Boyen '08]

The Powers of Tau

Structured Reference String for Polynomial Commitments,
introduced by [Kate-Zaverucha-Goldberg ’10]
Related to Strong Diffie-Hellman [Boneh-Boyen ’08]

Related SRSes used in:

“Phase 1” of [Gro16]
[GKM+18]
“Sonic” [MBKM19]
“Plonk” [GWC19]
“AuroraLight” [Gab19]
“Marlin” [CHM+20]

The Powers of Tau

Structured Reference String for Polynomial Commitments,
introduced by [Kate-Zaverucha-Goldberg '10]
Related to Strong Diffie-Hellman [Boneh-Boyen '08]

Related SRSes used in:

"Phase 1" of [Gro16]
[GKM+18]
"Sonic" [MBKM19]
"Plonk" [GWC19]
"AuroraLight" [Gab19]
"Marlin" [CHM+20]

Currently deployed by:

FileCoin
Ethereum (Semaphore)
ZCash
Probably others...

SRS Sampling Problem

τ must remain secret
(for SNARK soundness)

AND

No public-coin sampling
(computing $\tau^2 G$ given τG is hard)

SRS Sampling Problem

τ must remain secret (for SNARK soundness) AND No public-coin sampling (computing $\tau^2 G$ given τG is hard)

Common Solution: Sample SRS via MPC

[Ben-Sasson-Chiesa-Green-Tromer-Virza '15]
[Bowe-Gabizon-Miers '17]
[Bowe-Gabizon-Green '18]
[Groth-Kohlweiss-Maller-Meiklejohn-Miers '18]
[Kohlweiss-Maller-Siim-Volkhov '21]

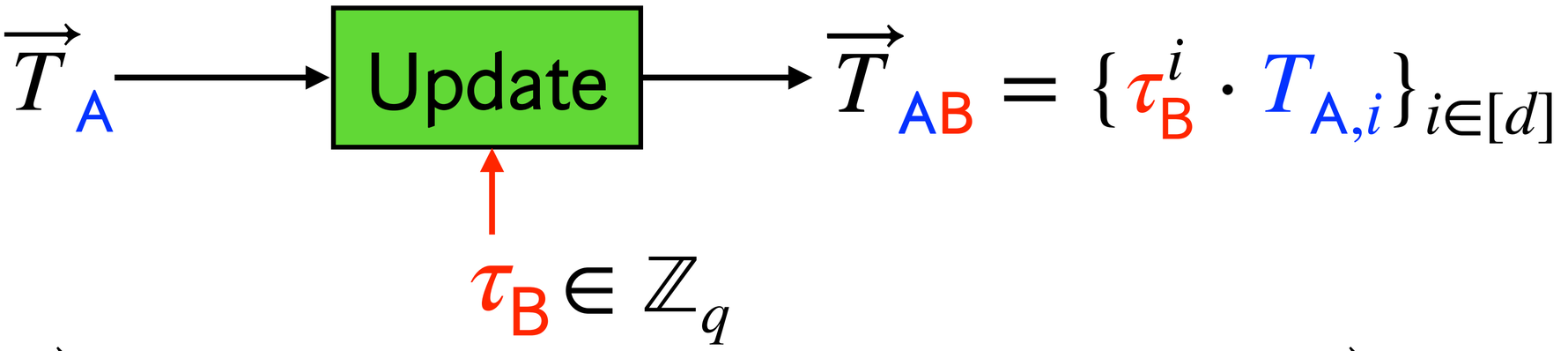
What do we want from this MPC?

- **Security against all-but-one active corruptions**:
  Trust (almost) nobody!
- **Scalability**:
  Support for hundreds/thousands of parties to contribute
- **Guaranteed Output Delivery**:
  The SRS does ultimately need to be available

Canonical Powers of Tau Sampling

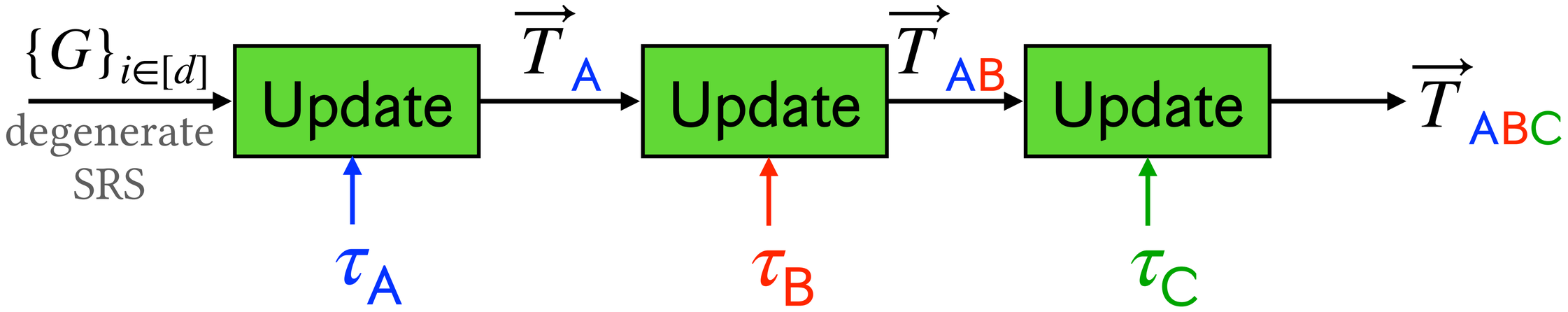
Building Block:

Given $\overrightarrow{T}_{\mathsf{A}} = \{T_{\mathsf{A},i} \in \mathbb{G}\}_{i \in [d]}$, define function

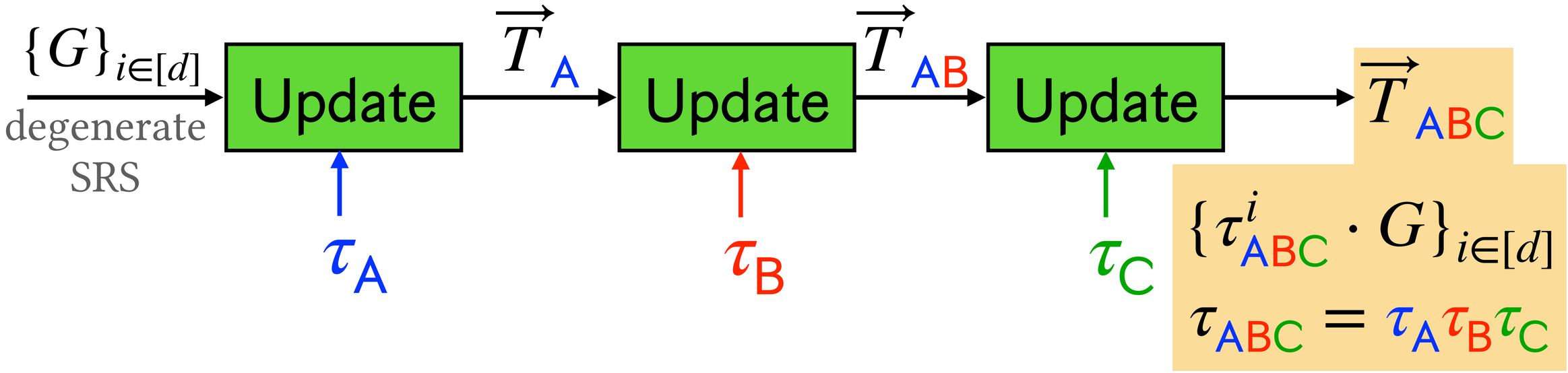


Observe: If $\overrightarrow{T}_{\mathsf{A}}$ is a Powers-of-Tau SRS, then so is $\overrightarrow{T}_{\mathsf{AB}}$

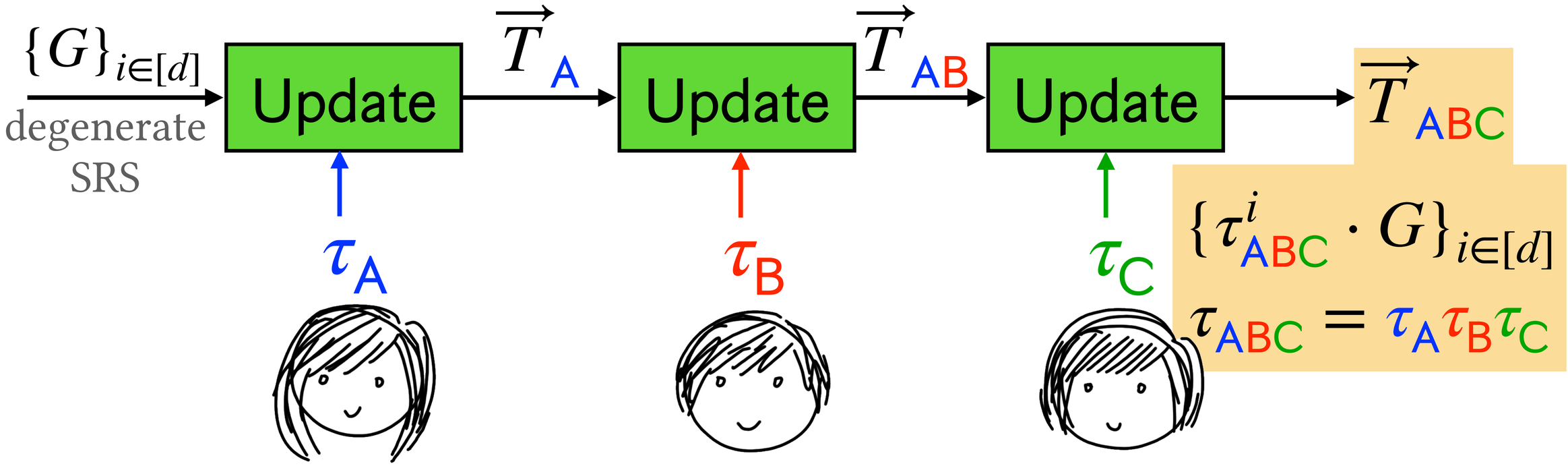
Canonical Powers of Tau Sampling



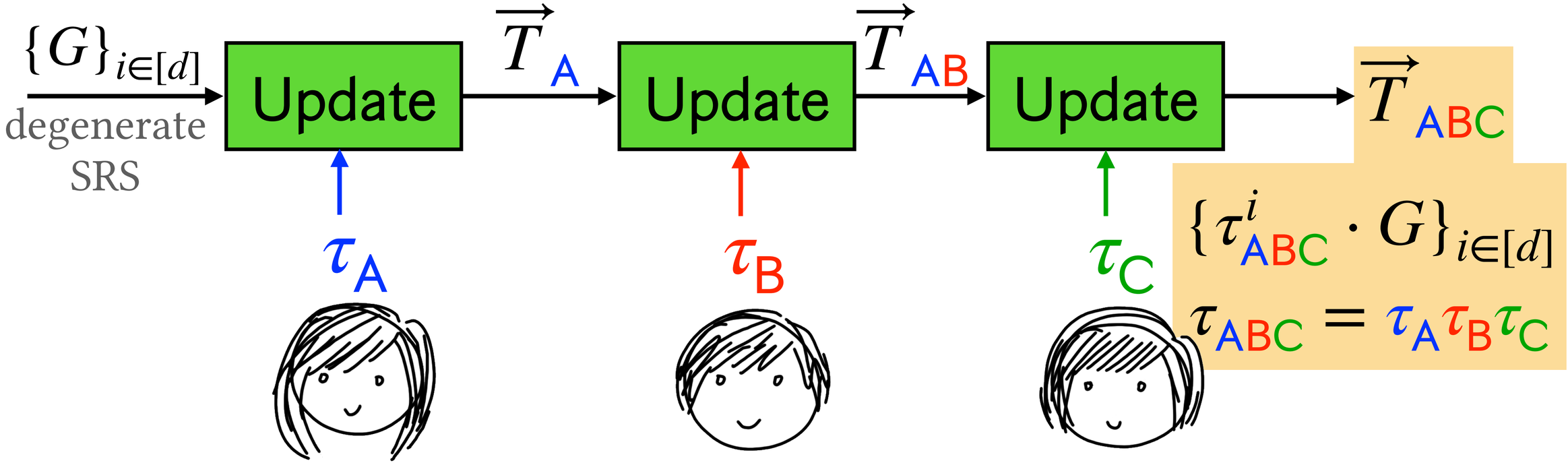
Canonical Powers of Tau Sampling



Canonical Powers of Tau Sampling

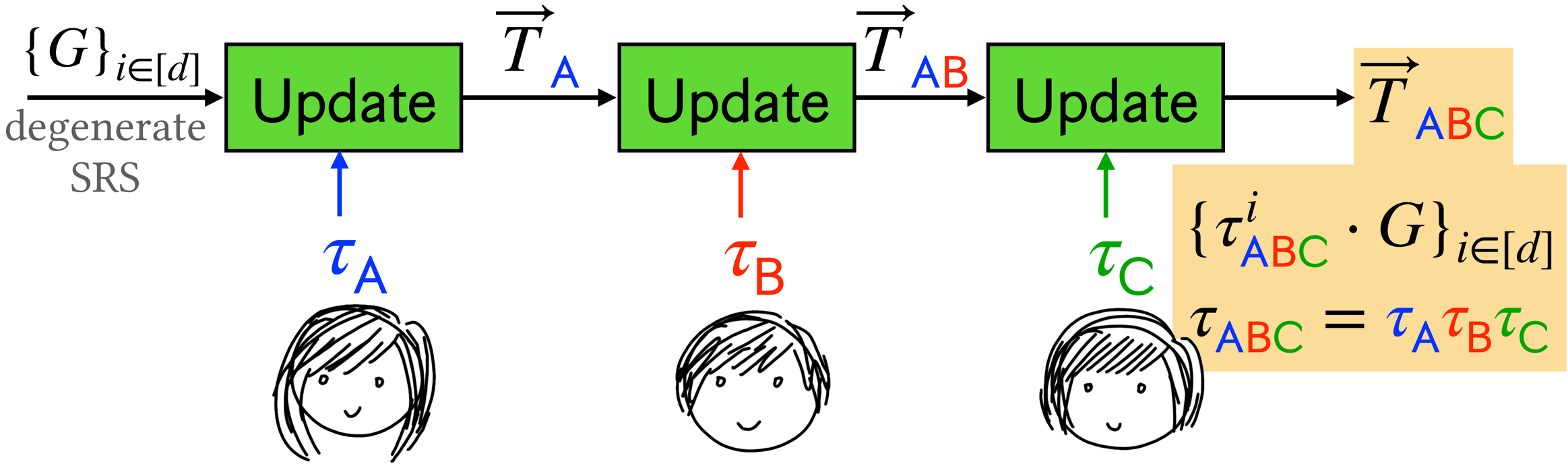


Canonical Powers of Tau Sampling



Security:
τ_{ABC} hidden unless all corrupt

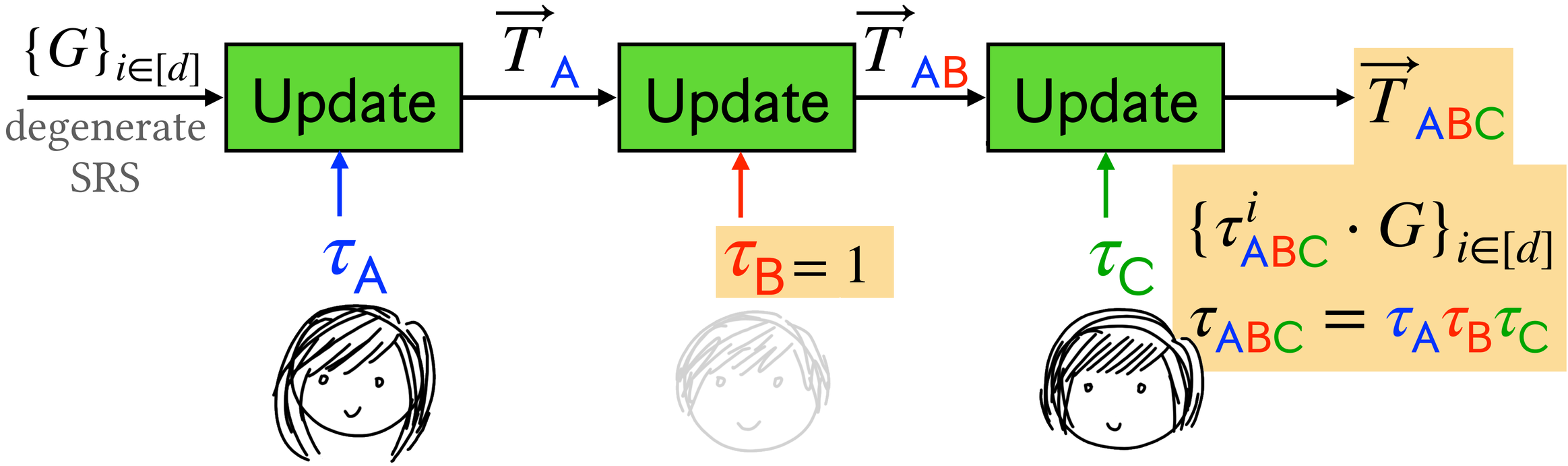
Canonical Powers of Tau Sampling



Security:
τ_{ABC} hidden unless all corrupt

GOD:
Skip non-responsive parties

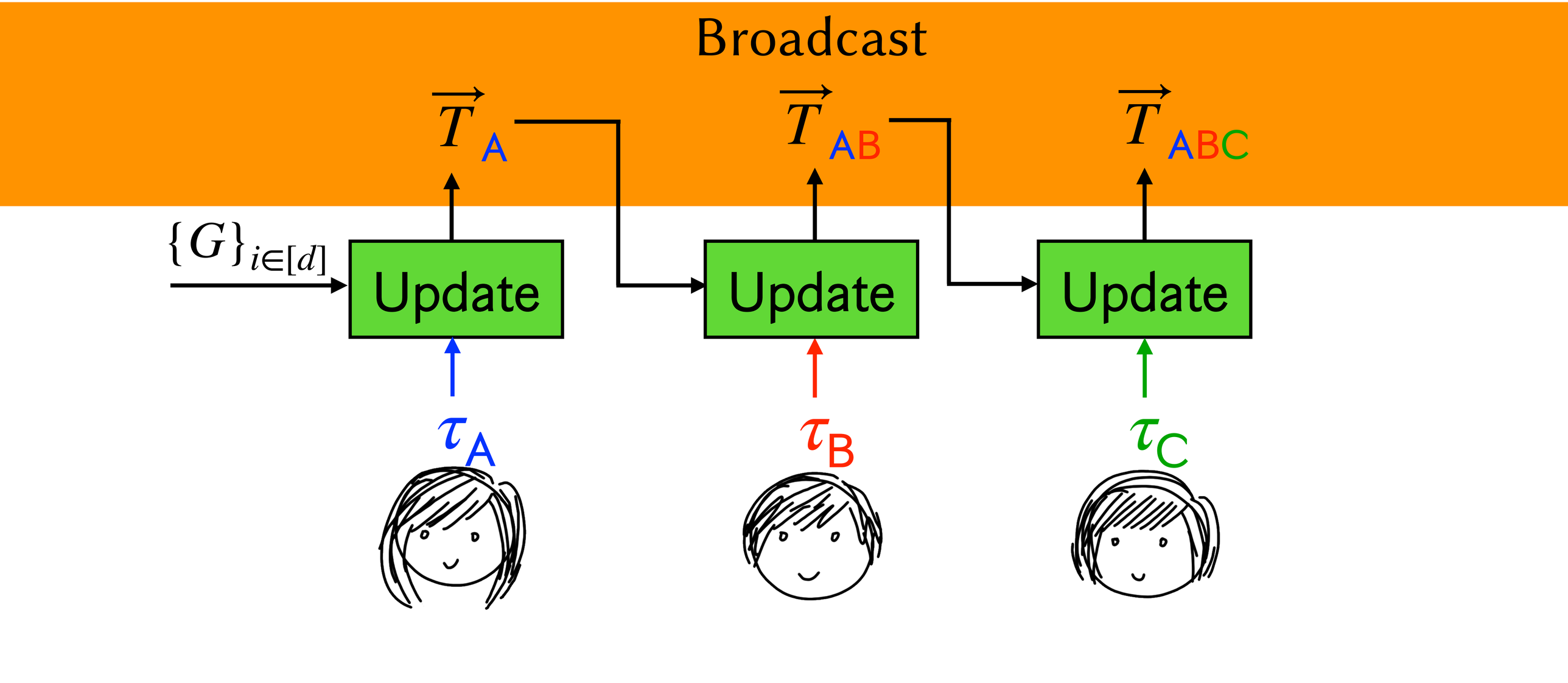
Canonical Powers of Tau Sampling



Security:
τ_{ABC} hidden unless all corrupt

GOD:
Skip non-responsive parties

Canonical Powers of Tau Sampling



Properties of the Protocol

1. Random Inputs
2. *Round Robin* Structure

   $\Longrightarrow$ n Broadcast Rounds
3. Guaranteed Output Delivery against $n - 1$ corruptions

Parties

Strongly Player Replaceable Round Robin Protocol

1. Random Inputs
2. *Round Robin* Structure

   $\Longrightarrow$ n Broadcast Rounds
3. Guaranteed Output Delivery against $n - 1$ corruptions

Parties

SPR³ Protocol

1. Random Inputs
2. *Round Robin* Structure

   $\Longrightarrow$ n Broadcast Rounds
3. Guaranteed Output Delivery against $n - 1$ corruptions

Parties

Our Main Result

A Protocol Compiler

Input: SPR^3 Protocol with GOD

Output:

1. $O(\sqrt{n})$ Broadcast Rounds
2. Guaranteed Output Delivery against $n - 1$ corruptions

A Corollary

Q: What other SPR^3 protocols exist?
A: Verifiable Mixnets

We give the first robust mixnet
in sublinear broadcast rounds

This Talk:

1. Powers of Tau/SPR^3 Protocols
2. A Brief History of GOD
3. Our Compiler
4. Bias
5. Future Directions

This Talk:

1. Powers of Tau/SPR³ Protocols
2. A Brief History of GOD
3. Our Compiler
4. Bias
5. Future Directions

In the beginning…

[Cleve ’86]: In the dishonest majority setting, some functionalities cannot be computed with GOD.

Any coin-tossing protocol with r rounds and a common output must have bias in $\Omega(1/r)$

In the beginning…

[Cleve ’86]: In the dishonest majority setting, some functionalities cannot be computed with GOD.

Any coin-tossing protocol with r rounds and a common output must have bias in $\Omega(1/r)$

Not a problem for:
— KZG10 Polynomial Commitments [This work]
— SNARKs (in the AGM) [GKM+18]

The Player Elimination Framework [GMW87]

1. Compute function via MPC with
   security against dishonest majority
2. Prove honest behavior via ZK
   over a broadcast channel
3. If anyone cheats, elminate them
   and start over

The Player Elimination Framework [GMW87]

1. Compute function via MPC with security against dishonest majority
2. Prove honest behavior via ZK over a broadcast channel
3. If anyone cheats, elminate them and start over

$\Longrightarrow$ $\Omega(n)$ rounds in dishonest majority setting

$\Longrightarrow$ the adversary can reject $\Omega(n)$ candidate outputs

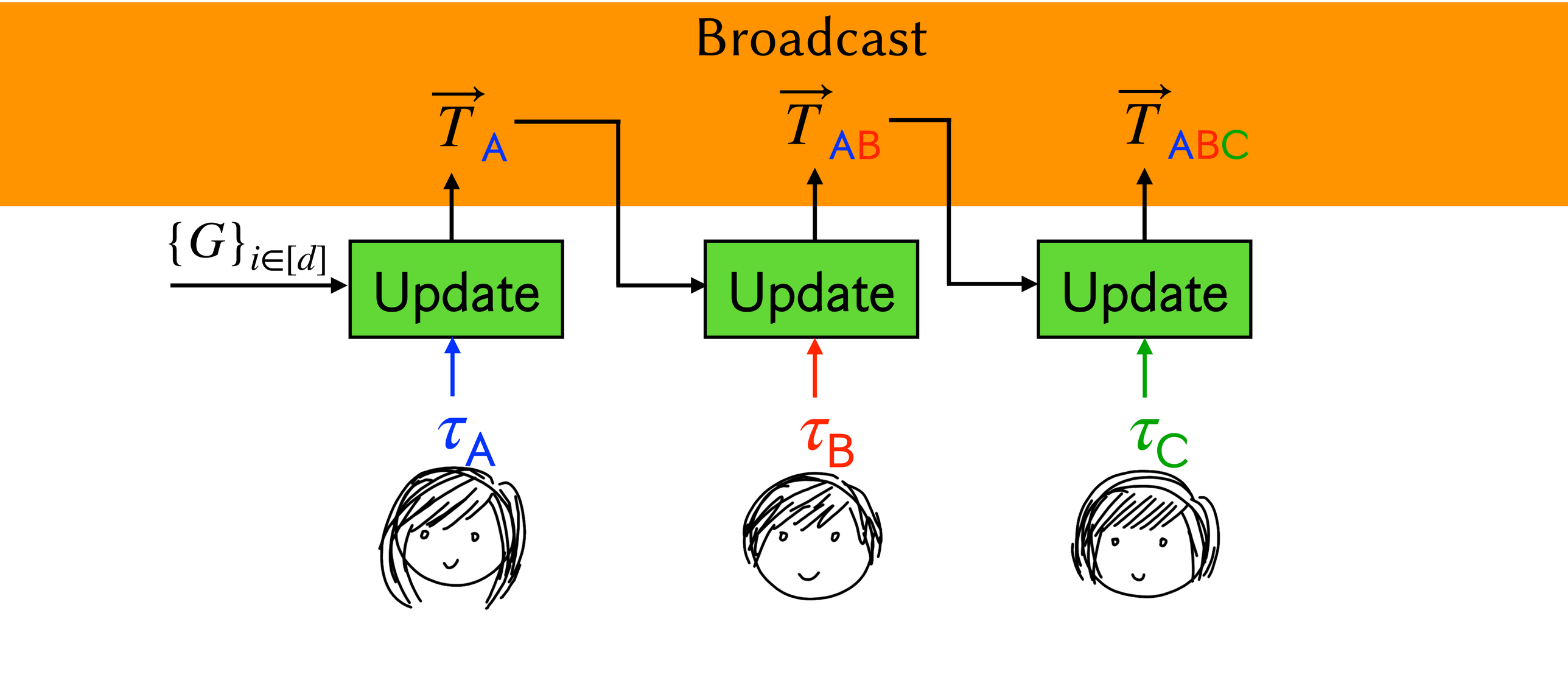
For most tasks, SotA
for GOD is player elimination
and $\Omega(n)$ broadcasts

We identify a class of
distributions that can be sampled
with GOD in $O(\sqrt{n})$ broadcast rounds

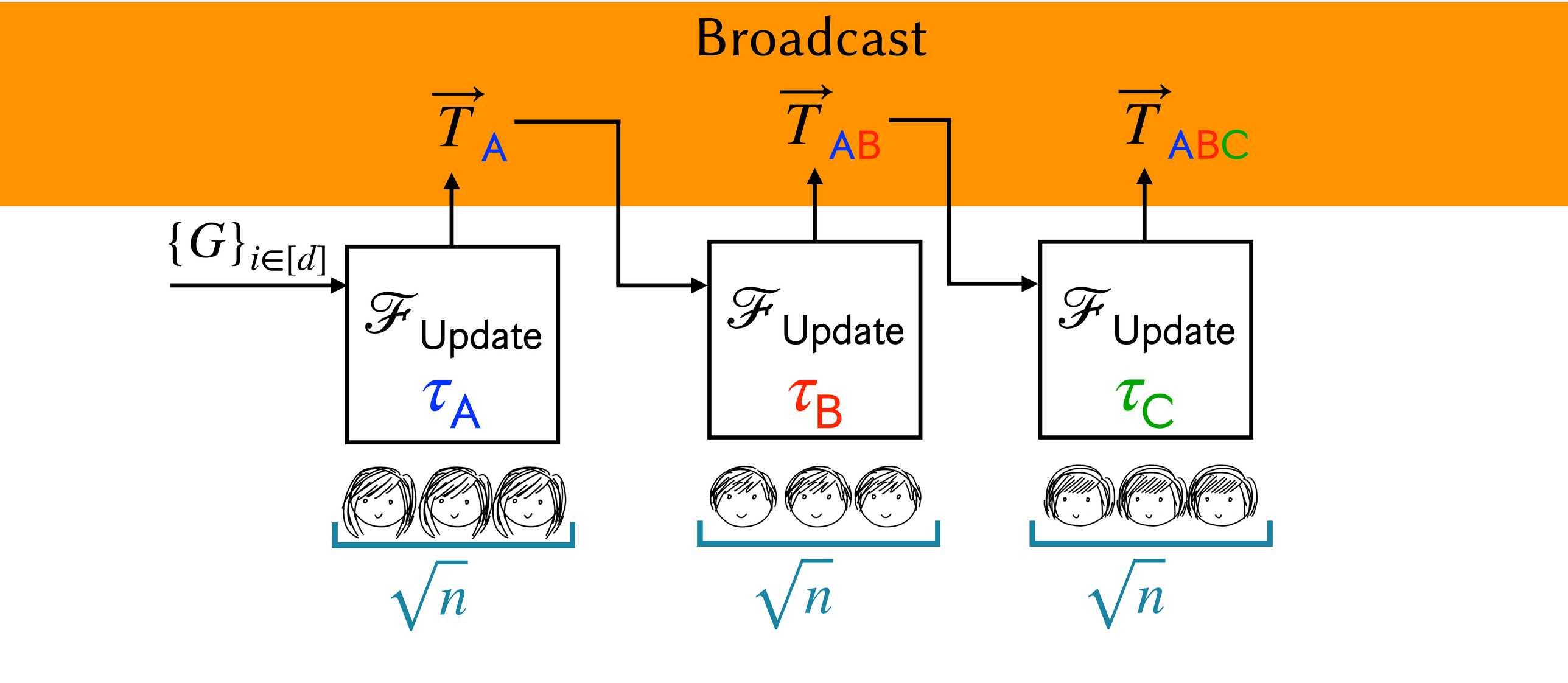
This Talk:

1. Powers of Tau/SPR³ Protocols
2. A Brief History of GOD
3. Our Compiler
4. Bias
5. Future Directions

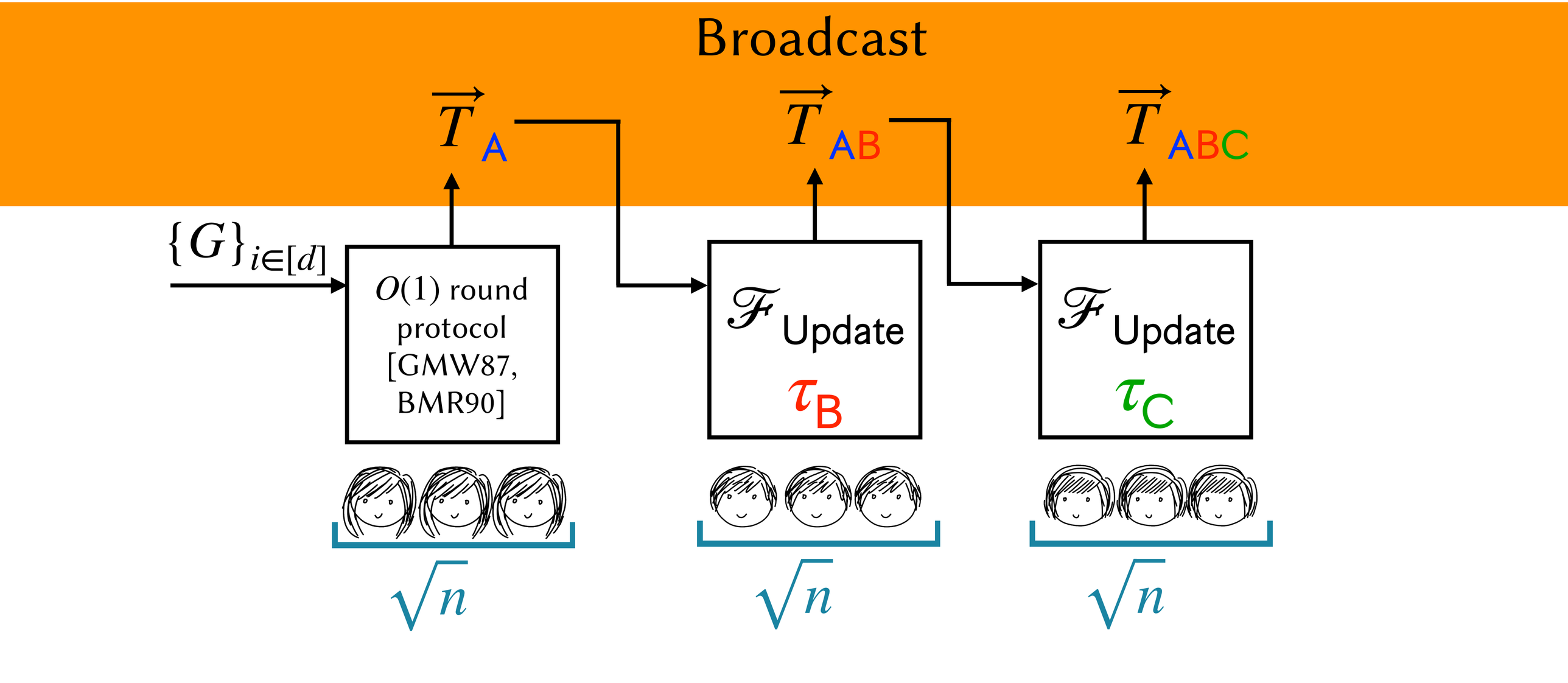
Canonical Powers of Tau Sampling



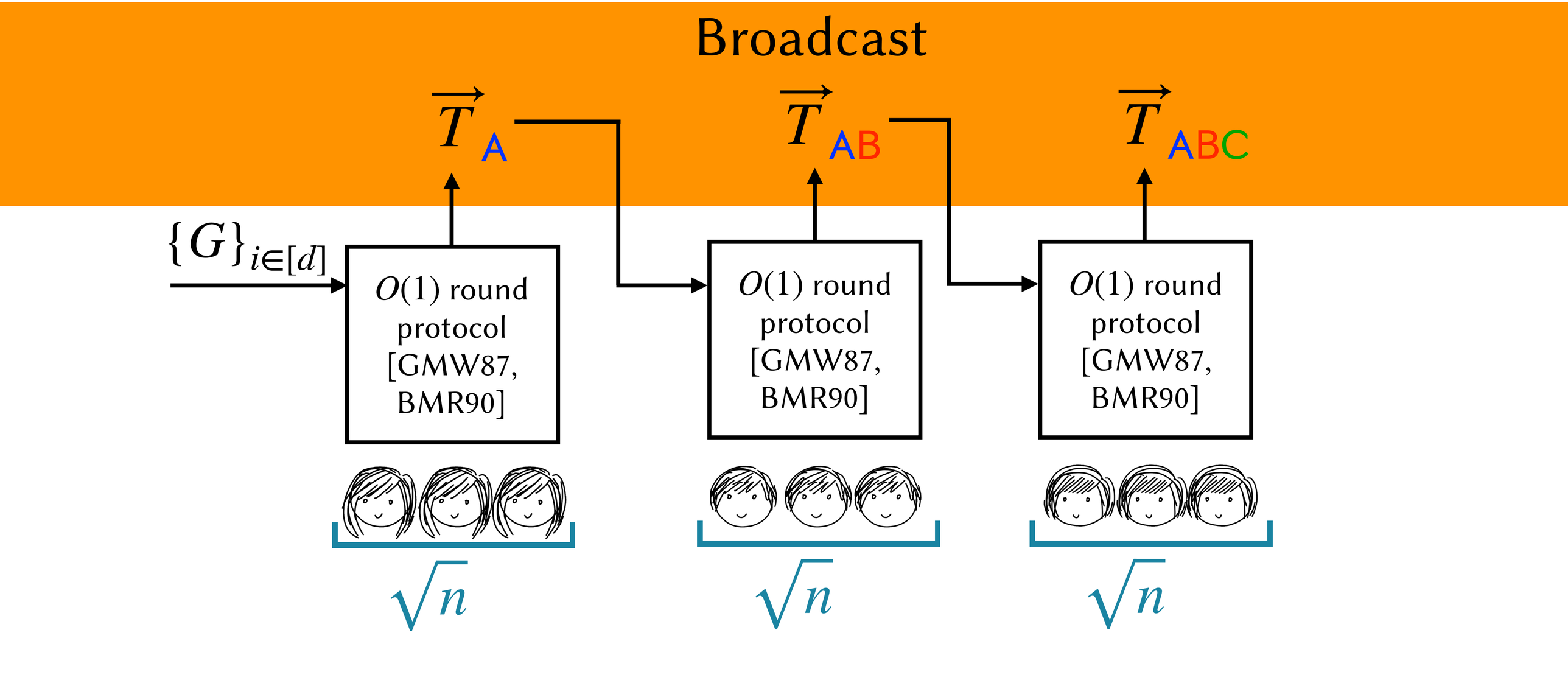
Powers of Tau Sampling++?



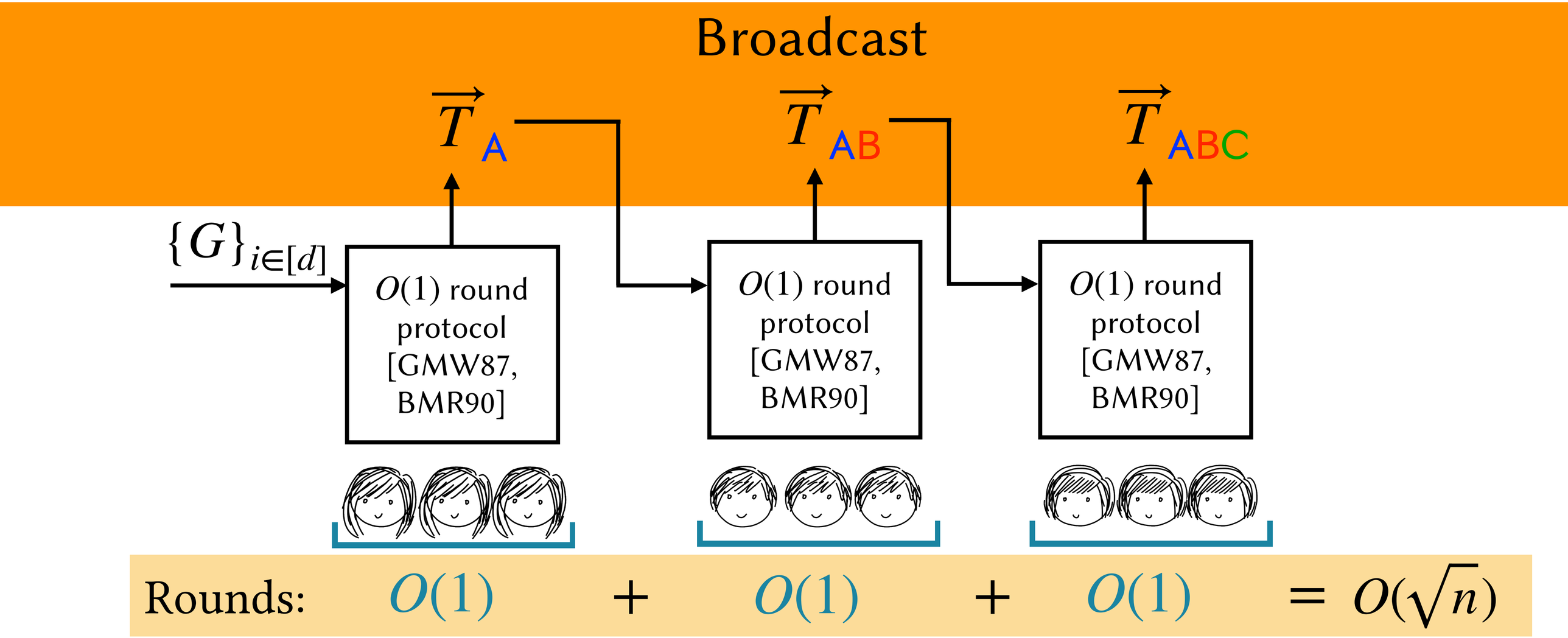
Powers of Tau Sampling++?



Powers of Tau Sampling++?

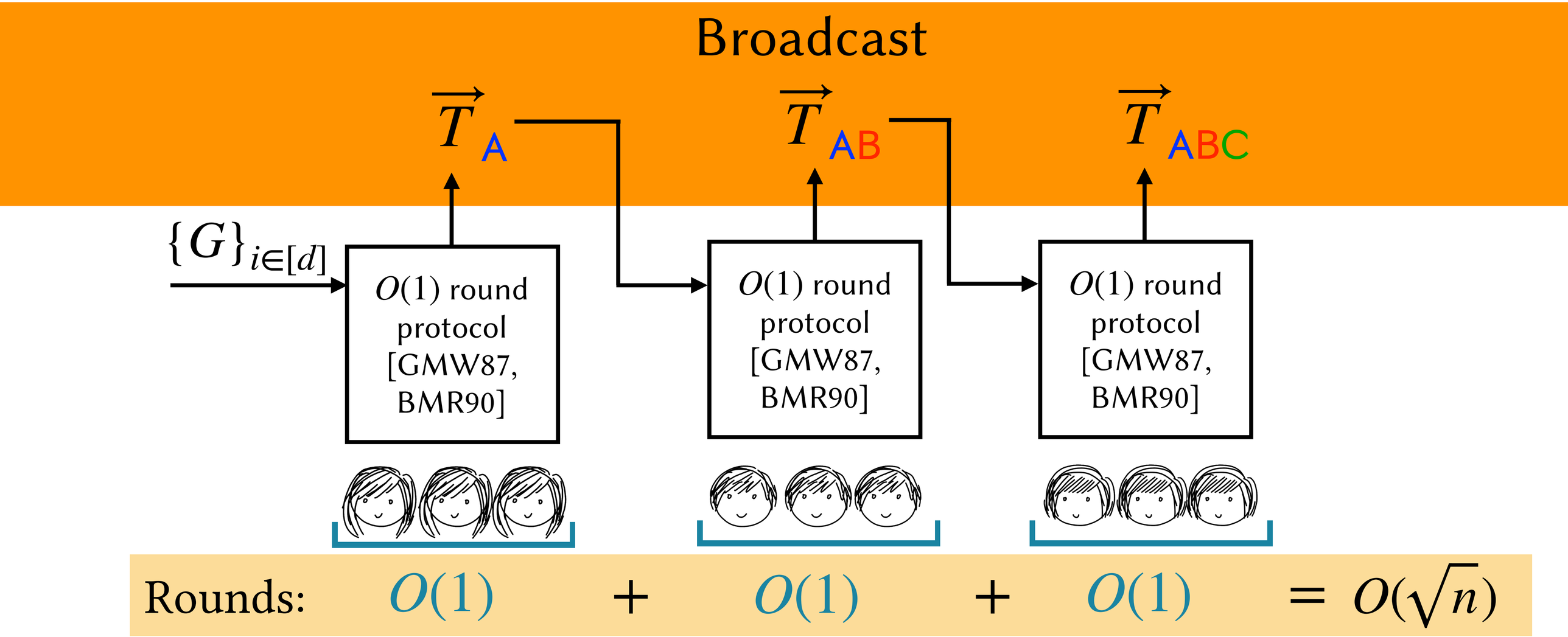


Powers of Tau Sampling++?

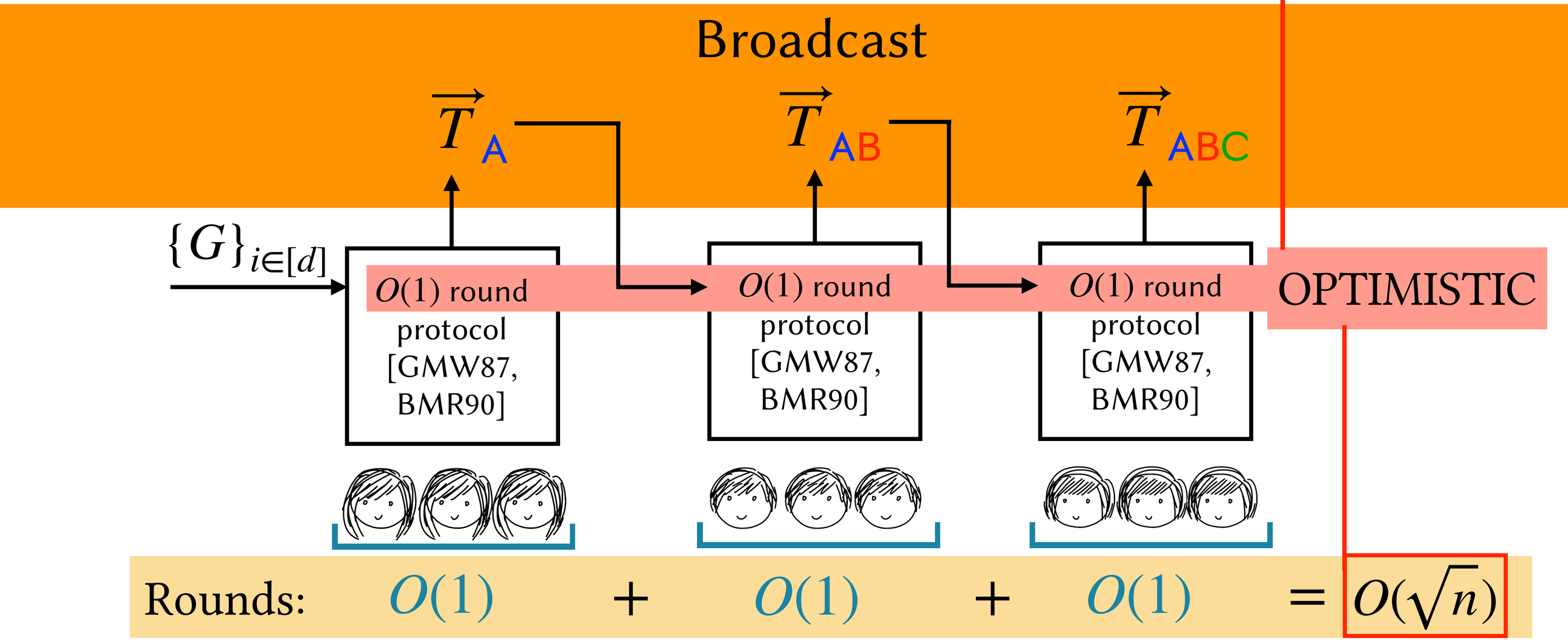


Rounds: $O(1) + O(1) + O(1) = O(\sqrt{n})$

Powers of Tau Sampling++?

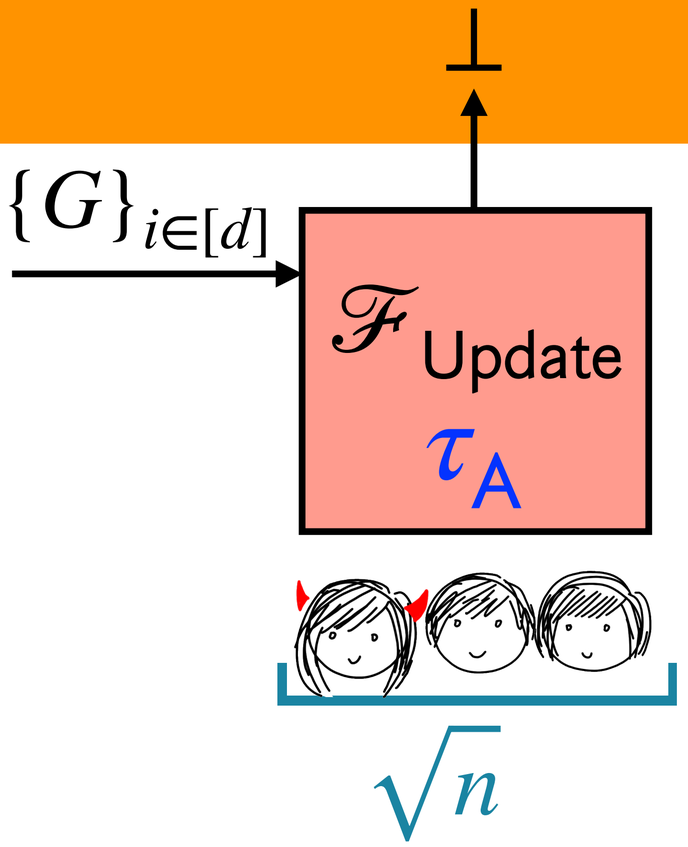


Powers of Tau Sampling++?



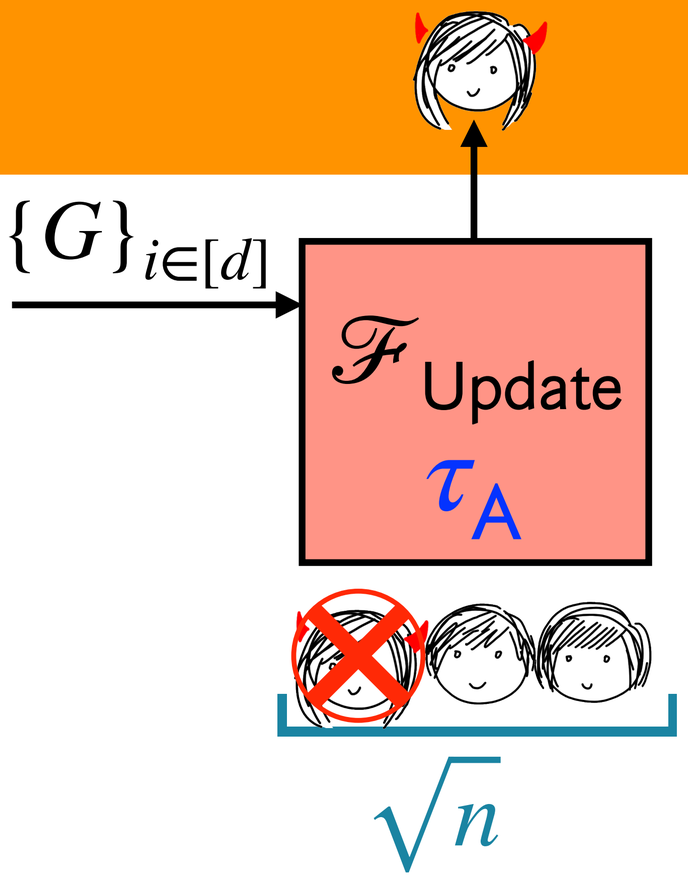
Deflating Optimism

Broadcast



Deflating Optimism

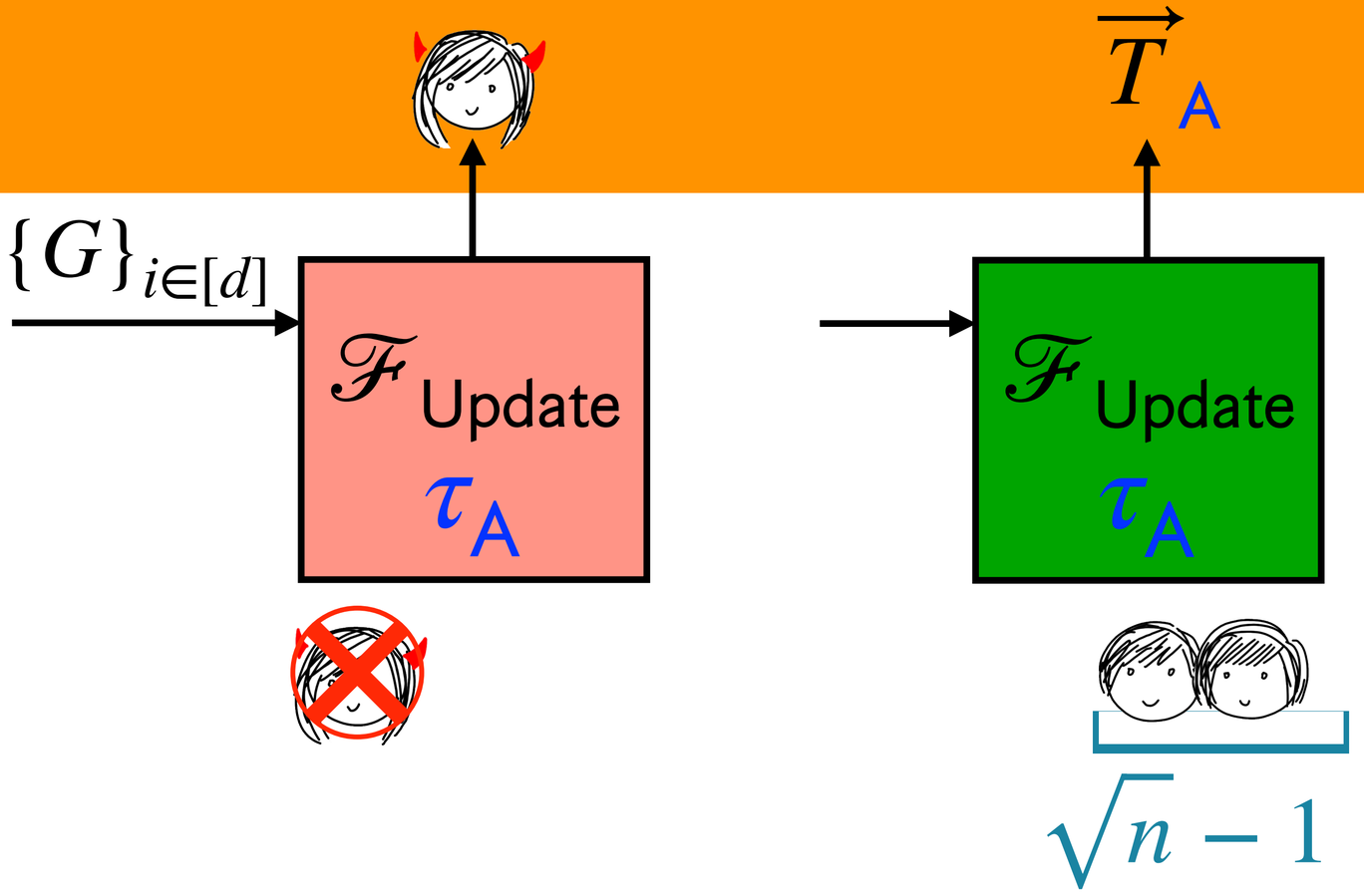
Broadcast



Tool:
Publicly Verifiable
Identifiable Abort
[GMW87, BMR90]

Deflating Optimism

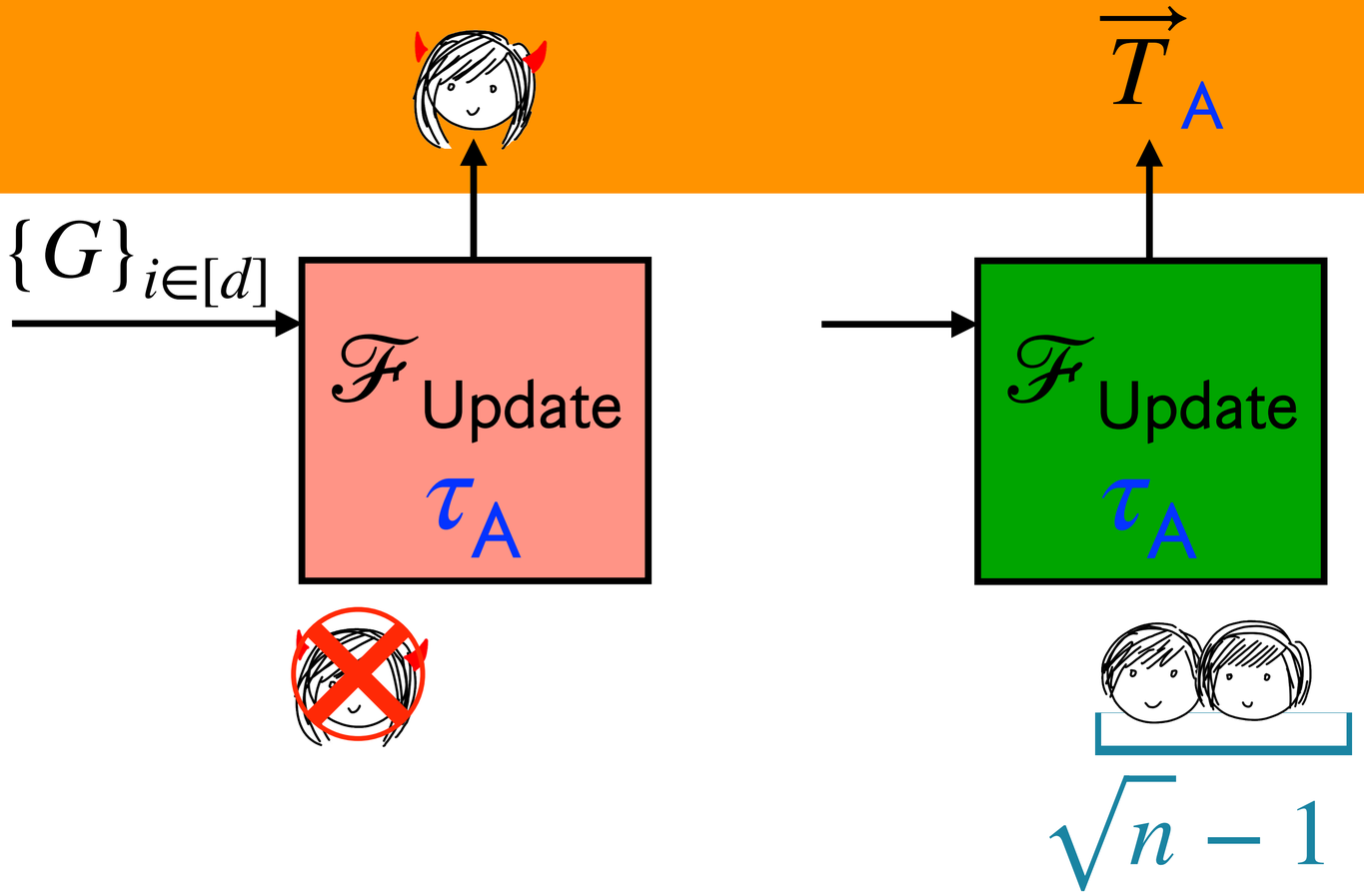
Broadcast



Tool:
Publicly Verifiable
Identifiable Abort
[GMW87, BMR90]

Deflating Optimism

Broadcast



Worst case:

Each committee needs

$\sqrt{n}$ restarts

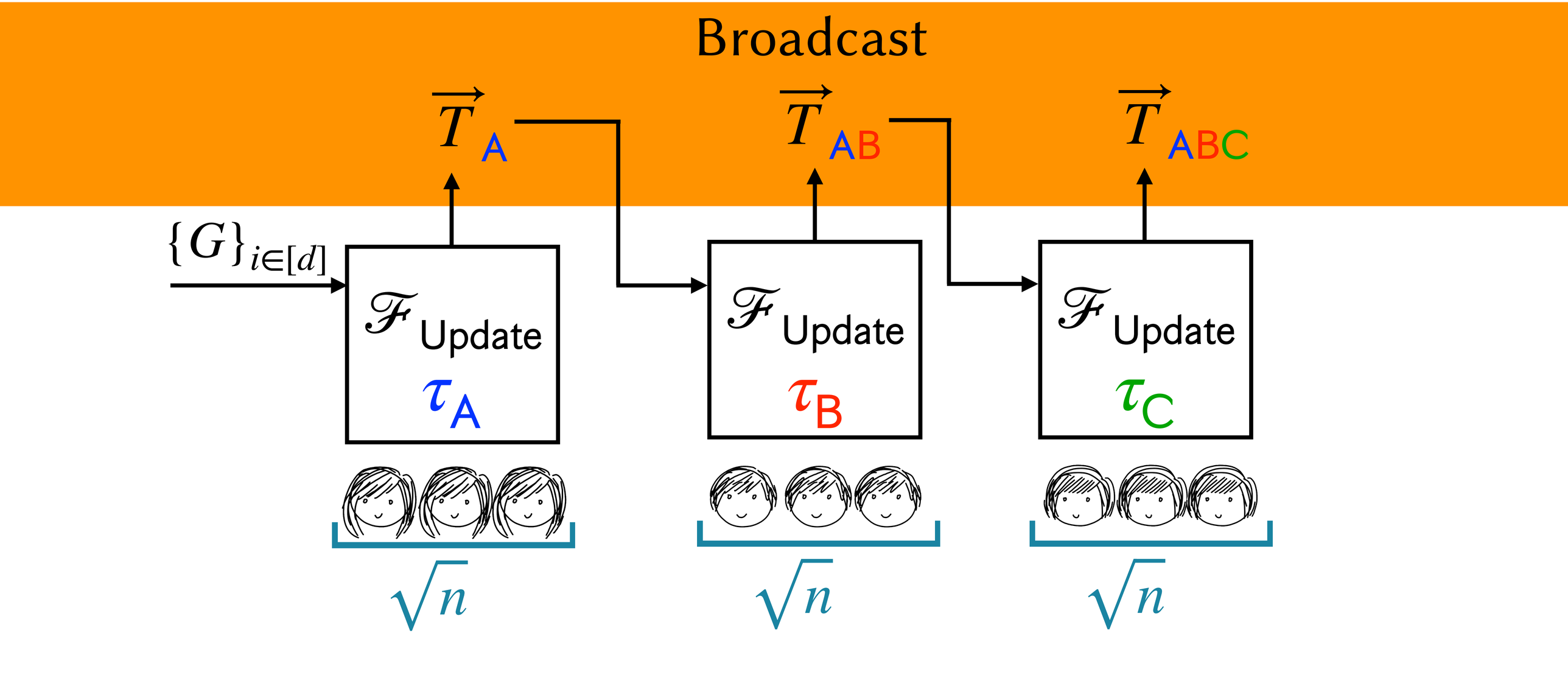
$\times$

$\sqrt{n}$ committees

$= O(n)$ rounds

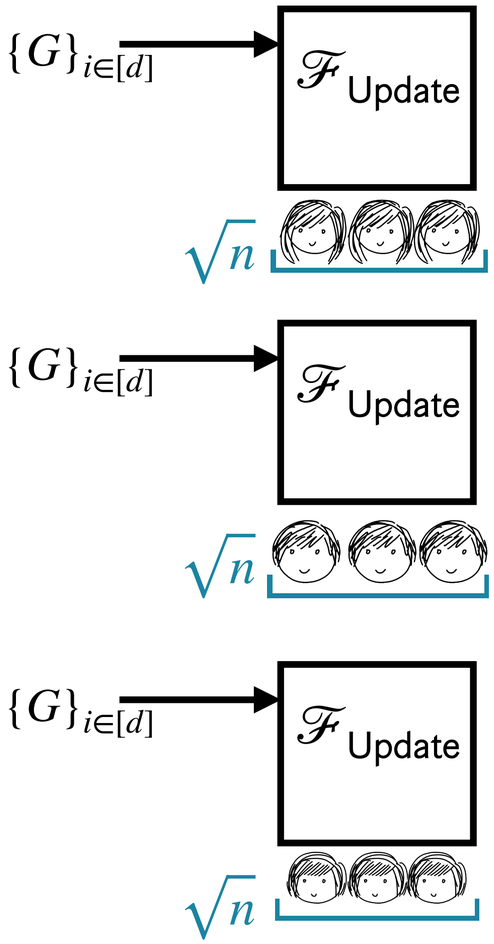
Achieved nothing (yet)

Powers of Tau Sampling++?



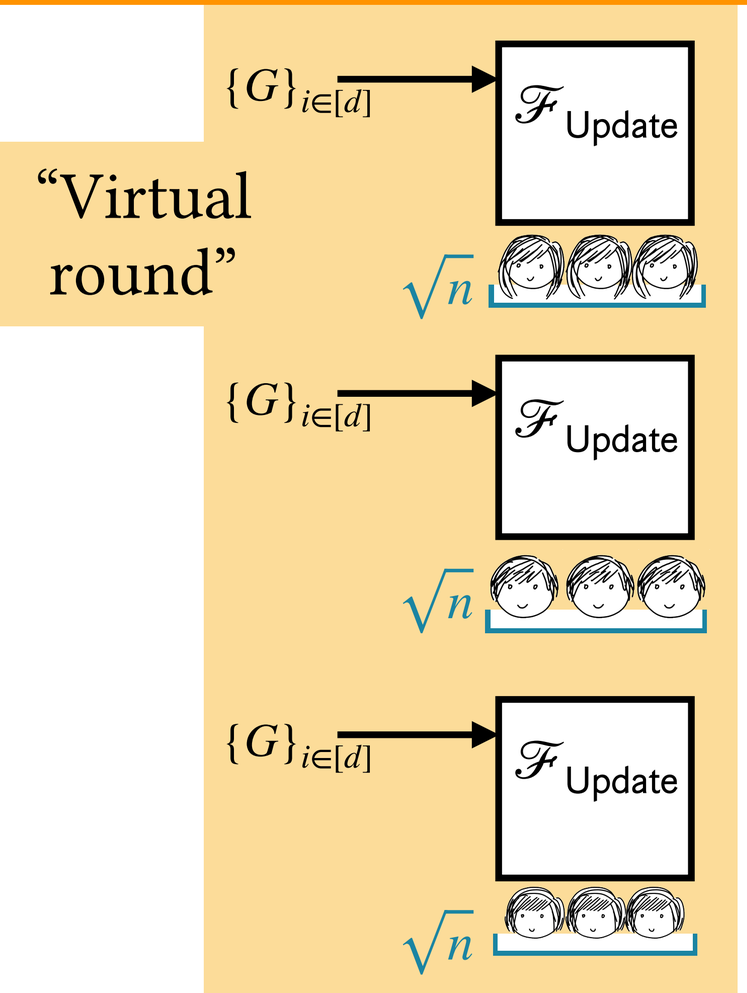
Powers of Tau Sampling++?

Broadcast



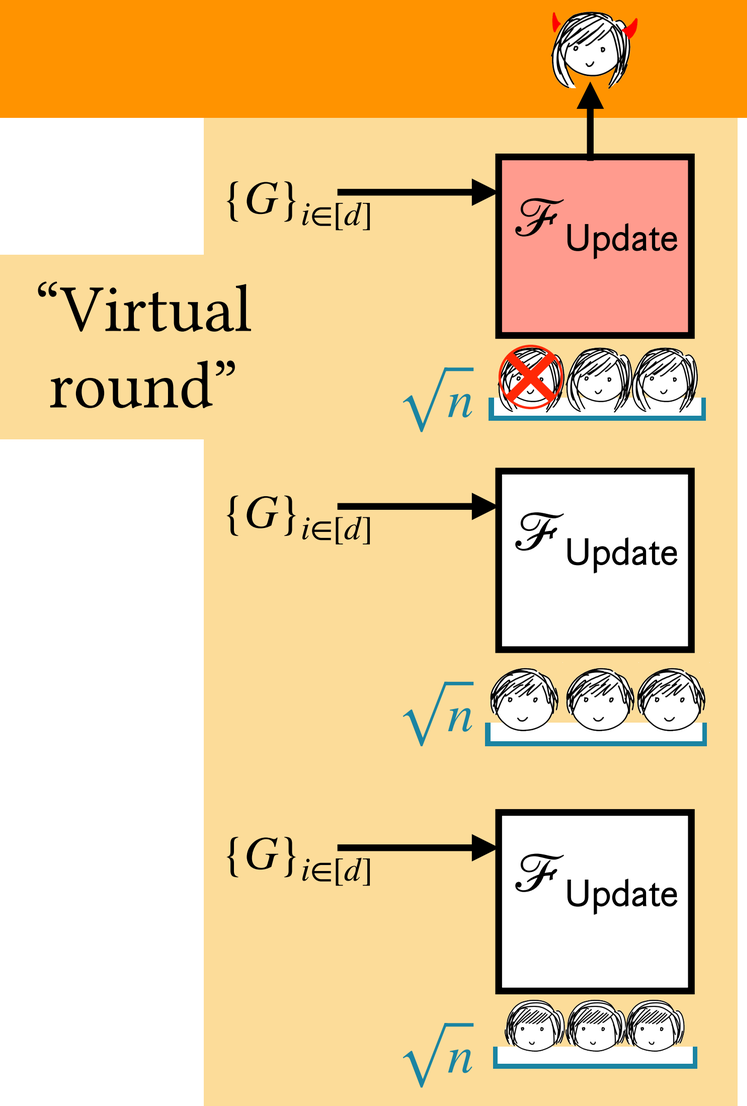
Powers of Tau Sampling++?

Broadcast



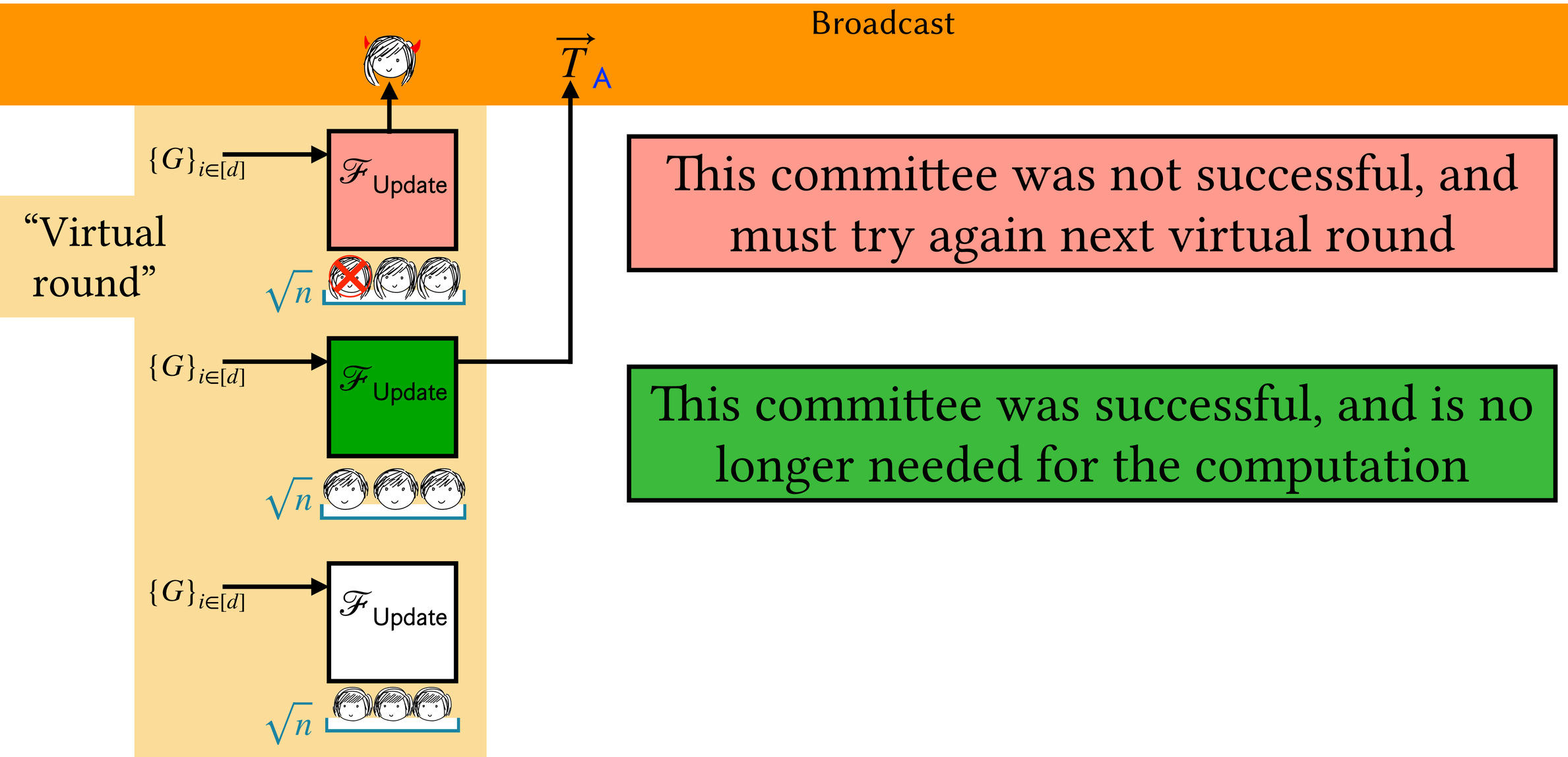
Powers of Tau Sampling++?

Broadcast



This committee was not successful, and must try again next virtual round

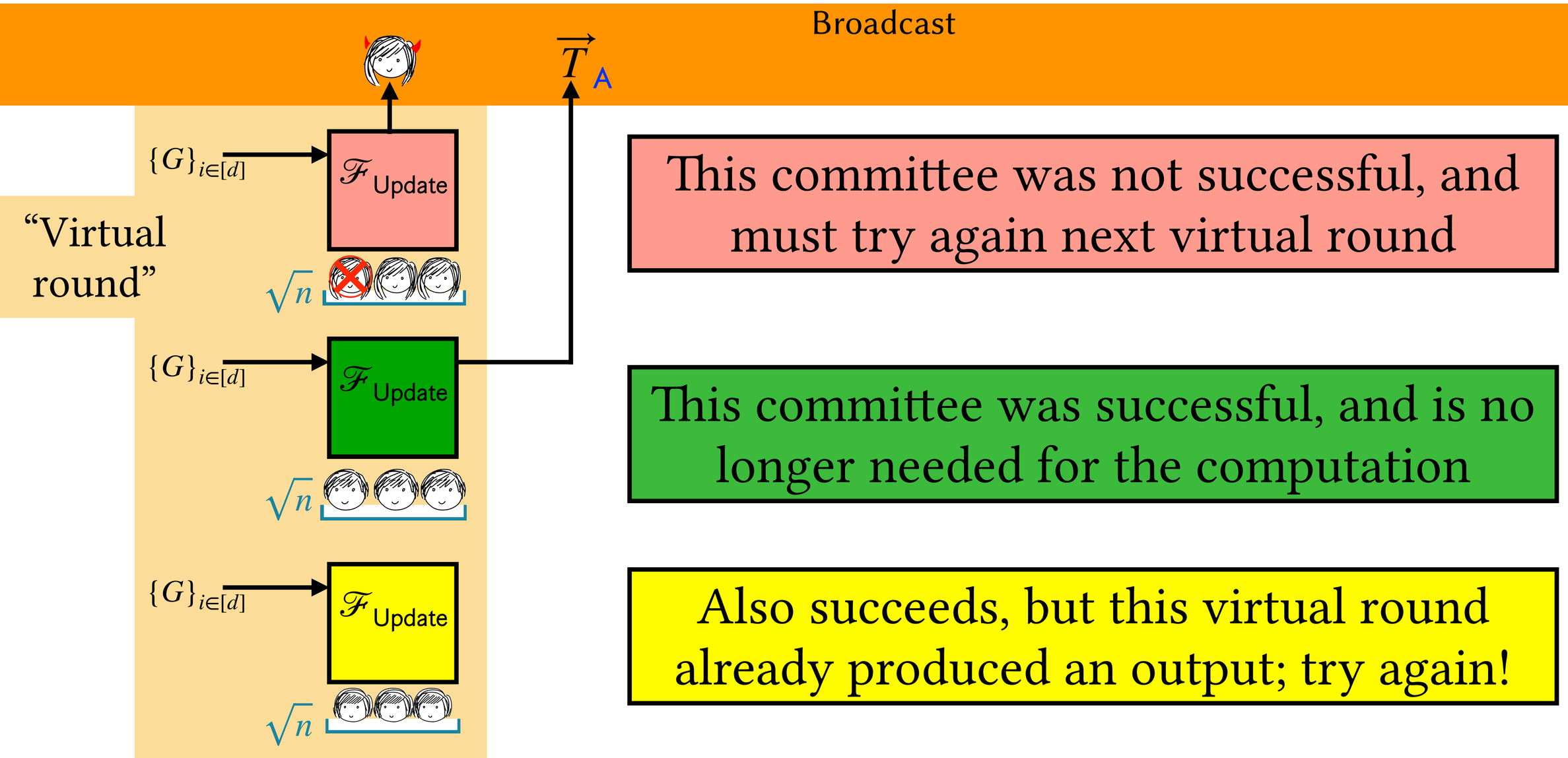
Powers of Tau Sampling++?



This committee was not successful, and must try again next virtual round

This committee was successful, and is no longer needed for the computation

Powers of Tau Sampling++?

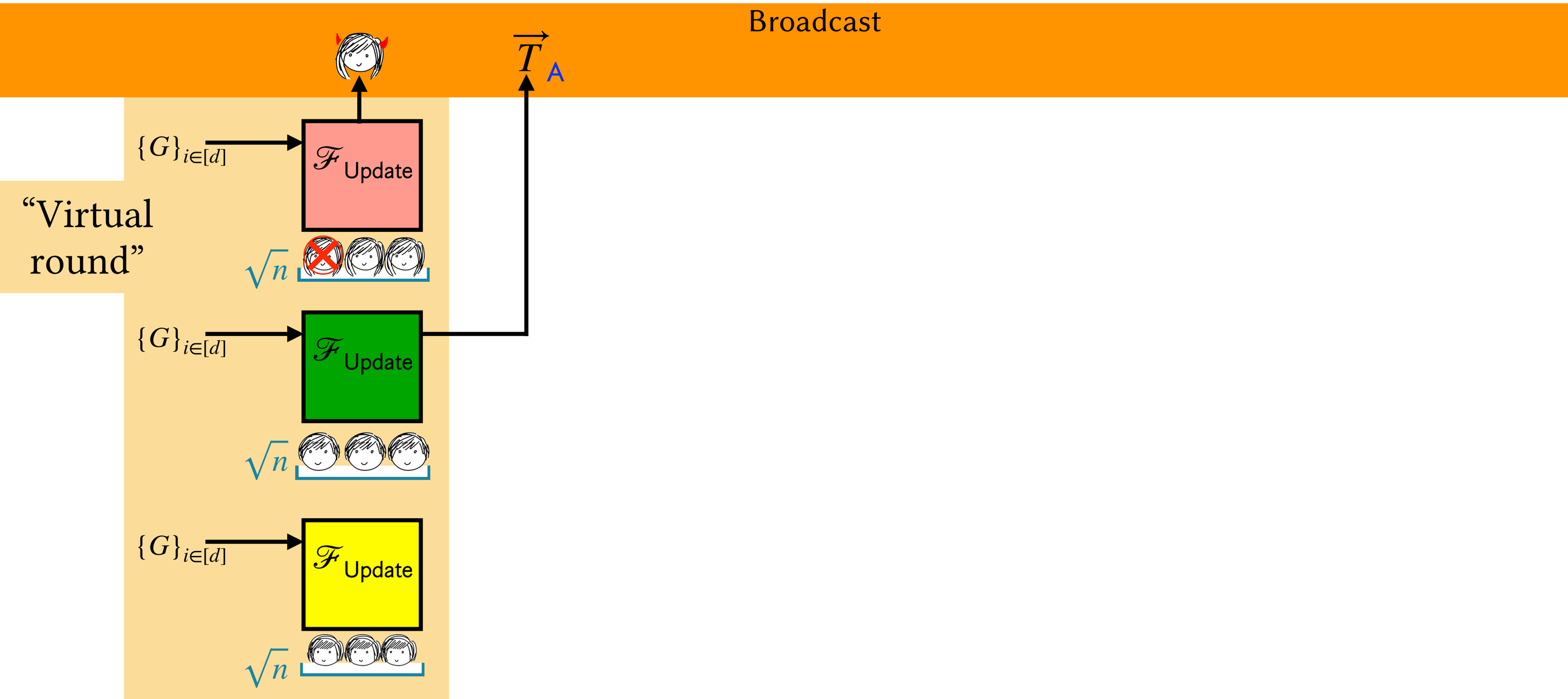


This committee was not successful, and must try again next virtual round

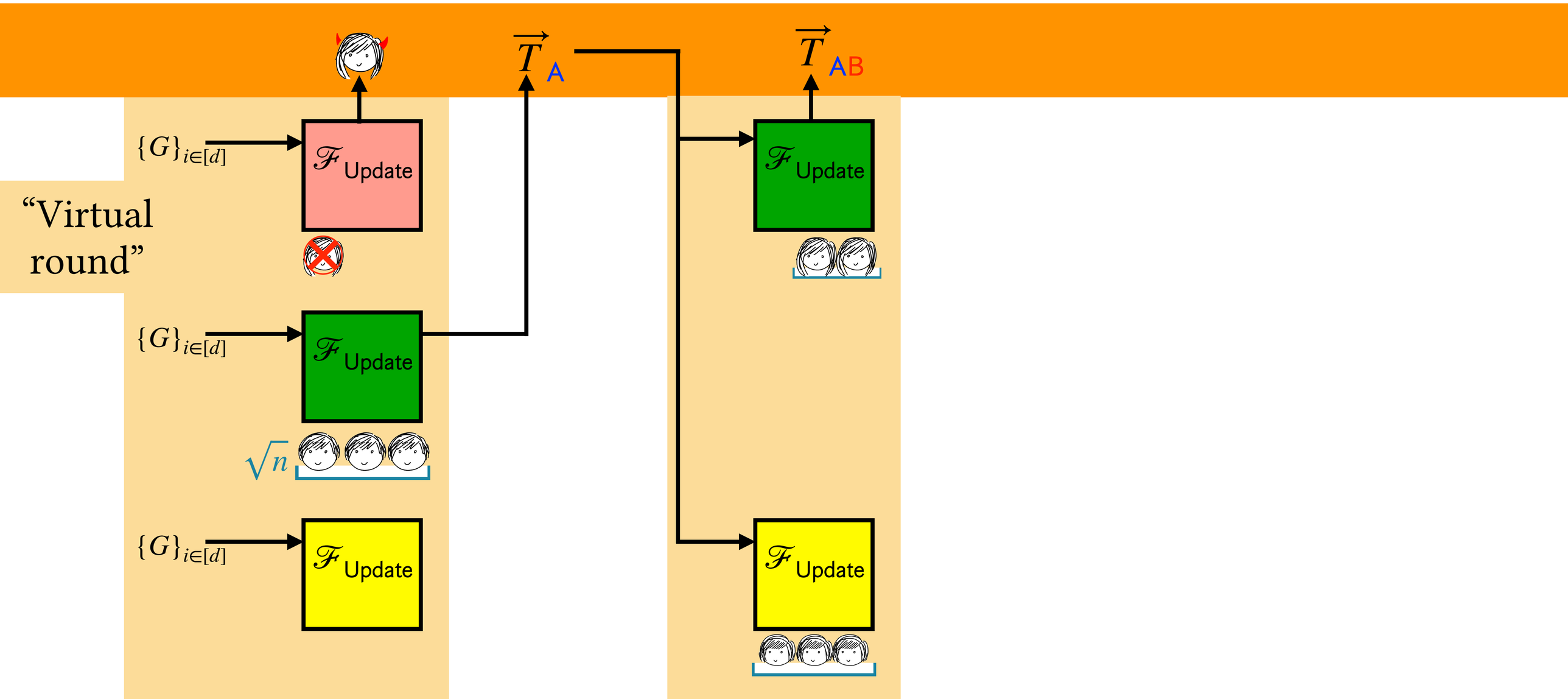
This committee was successful, and is no longer needed for the computation

Also succeeds, but this virtual round already produced an output; try again!

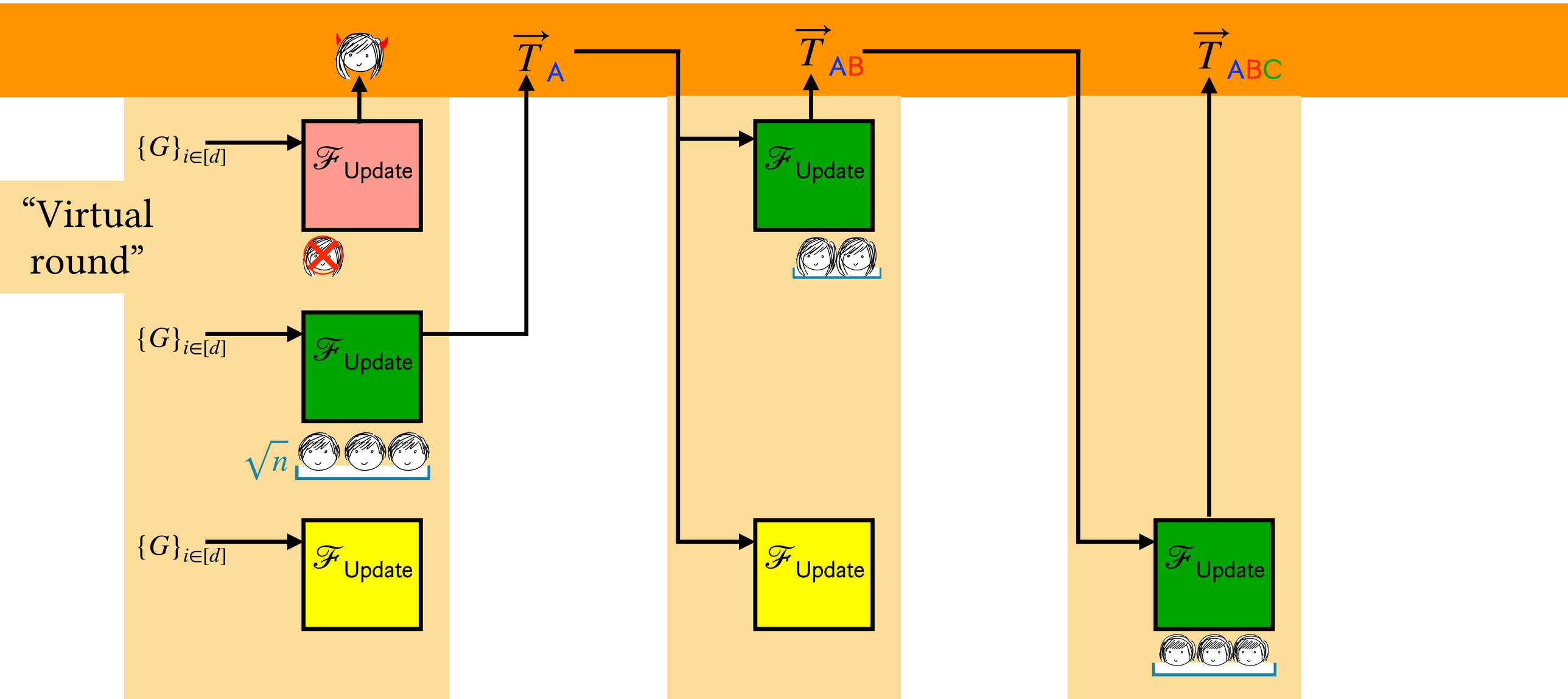
Powers of Tau Sampling++?



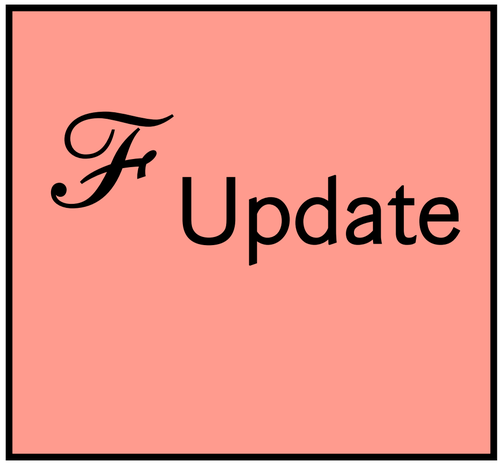
Powers of Tau Sampling++?



Powers of Tau Sampling++?



Each Committee Can Be …



… at most $\sqrt{n}$ times:

Each time a party is eliminated,

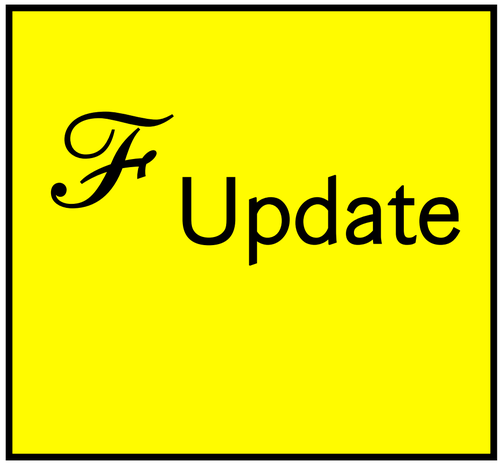
and there are $\sqrt{n}$ parties.

Each Committee Can Be …

… at most once:

The whole committee retires after this event.

Each Committee Can Be …



… at most $\sqrt{n} - 1$ times:

This only occurs if another committee is green.

Thus, the maximal number virtual rounds is

$$\max(\mathcal{F}_{\text{Update}}) + \max(\mathcal{F}_{\text{Update}}) + \max(\mathcal{F}_{\text{Update}})$$

$$\leq 2\sqrt{n}$$

Each virtual round corresponds to $O(1)$ “actual” broadcast rounds [GMW87,BMR90]

$$\Longrightarrow \quad O(\sqrt{n}) \text{ real broadcast rounds}$$

This Talk:

1. Powers of Tau/SPR^3 Protocols
2. A Brief History of GOD
3. Our Compiler
4. Bias
5. Future Directions

Does This Affect Output Distribution?

- Our ideal functionality permits a rejection sampling interface which was not available originally
- However for both of our applications (Powers of Tau and Mixnets) we prove unconditionally that **this extra interface is useless**
- Intuition: Output of the functionality is perfectly rerandomizable

This Talk:

1. Powers of Tau/SPR³ Protocols
2. A Brief History of GOD
3. Our Compiler
4. Bias
5. Future Directions

Future Directions

TPMPC 2022

14:00-15:15 Theory 4
Session chair: Sophia Yakoubov

14:00-14:25: Guaranteed Output in O(sqrt(n)) Rounds for Round-Robin Sampling Protocols

Ran Cohen, Jack Doerner, Yashvanth Kondi (Speaker) and abhi shelat

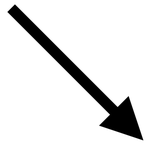
Future Directions

TPMPC 2022

14:00-15:15 Theory 4
Session chair: Sophia Yakoubov

14:00-14:25: Guaranteed Output in O(sqrt(n)) Rounds for Round-Robin Sampling Protocols

Ran Cohen, Jack Doerner, Yashvanth Kondi (Speaker) and abhi shelat



TPMPC 2023

9:30-10:45: Applications & Implementations 3
Session chair: Peter Scholl

Future Directions

TPMPC 2022

14:00-15:15 Theory 4
Session chair: Sophia Yakoubov

14:00-14:25: Guaranteed Output in O(sqrt(n)) Rounds for Round-Robin Sampling Protocols

Ran Cohen, Jack Doerner, Yashvanth Kondi (Speaker) and abhi shelat

TPMPC 2023

9:30-10:45: Applications & Implementations 3
Session chair: Peter Scholl

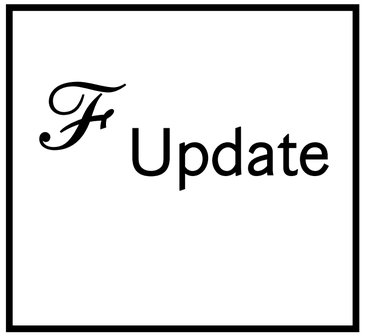
TPMPC 20??

21:00-22:00 Dessert and Networking

The Bottleneck

How to instantiate $\mathcal{F}$ Update ?

The Bottleneck

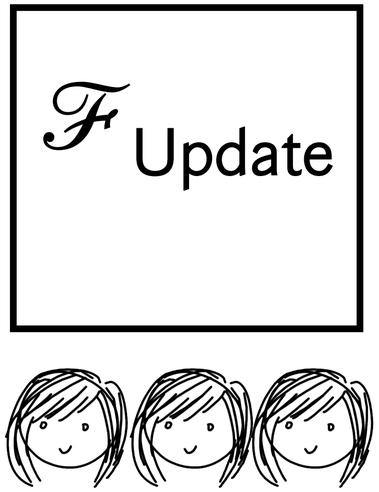
How to instantiate 

?

Feasibility:

- Express Update as Boolean circuit
- $O(1)$ round MPC [BMR90]
- Prove everything in ZK

The Bottleneck

How to instantiate 

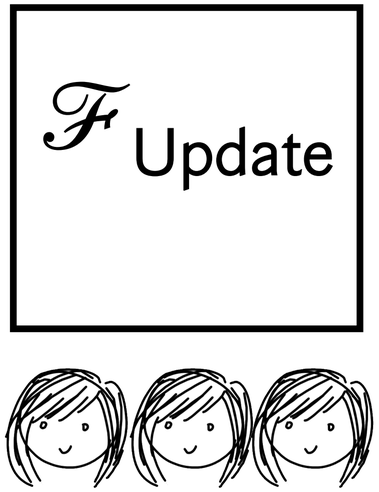
?

Feasibility:

- Express Update as Boolean circuit
- $O(1)$ round MPC [BMR90]
- Prove everything in ZK

Inefficient: $\mathbb{Z}_q$ multiplications and scalar multiplications in $\mathbb{G}$

The Bottleneck

How to instantiate 

?

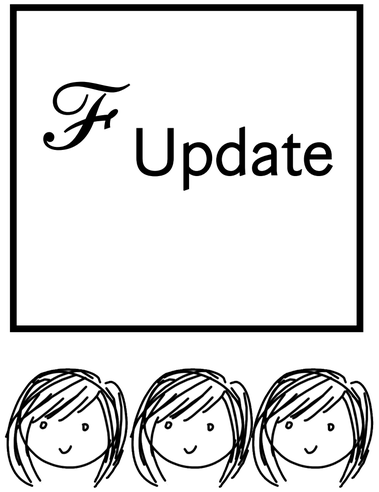
Maybe an arithmetic MPC system will work better?

Feasibility:

- Express Update as Boolean circuit
- $O(1)$ round MPC [BMR90]
- Prove everything in ZK

Inefficient: $\mathbb{Z}_q$ multiplications and scalar multiplications in $\mathbb{G}$

The Bottleneck

How to instantiate 

?

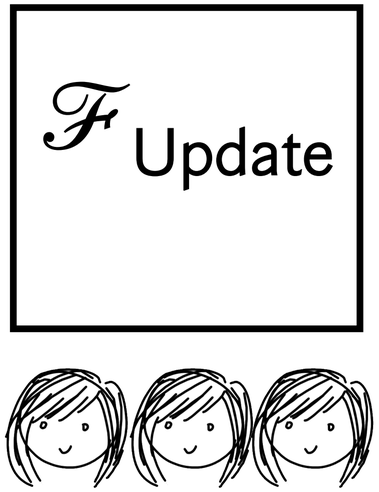
Maybe an arithmetic MPC system will work better?

Feasibility:

- Express Update as Boolean circuit
- $O(1)$ round MPC [BMR90]
- Prove everything in ZK

- Arithmetic circuit + exponentiation gates
- $O(\log d)$ round MPC
- Tailored consistency checks

The Bottleneck

How to instantiate 

?

Maybe an arithmetic MPC system will work better?

Feasibility:

- Express Update as Boolean circuit
- $O(1)$ round MPC [BMR90]
- Prove everythin

Overall complexity
$O(\sqrt{n}\log(d))$

- Arithmetic circuit + exponentiation gates
- $O(\log d)$ round MPC
- Tailored consistency checks

What's the Relevant Efficiency Metric?

- We have focused on reducing rounds in this talk
- Computation and bandwidth also plausibly the constraints in several settings
- In settings where bandwidth is not the bottleneck, we identify "*latency due to exponentiations*" as the relevant efficiency metric
- We show how our technique can be used to **outperform the canonical protocol** per this metric

“Fully Secure” MPC with PVIA is Unnecessary

- A relaxed instantiation of $\mathcal{F}_{\mathsf{Update}}$ is sufficient
- Privacy of inputs is unnecessary when tracing who caused an abort—the output will be thrown away and inputs never reused
- **Work in progress**: instantiating $\mathcal{F}_{\mathsf{Update}}$ within a small constant factor of standard (sec w. abort) arithmetic MPC for $\approx d$ gate

Thanks!

eprint.iacr.org/2022/257

Thanks to Eysa Lee for