From Nakamoto to YOSO --

A New Model for MPC

Tal Rabin (UPenn and Algorand Foundation)

May 31, 2022

Based on Joint works with: Fabrice Benhamouda, Craig Gentry, Sergey Gorbunov, Shai Halevi, Hugo Krawczyk, Chengyu Lin, Bernardo Magri, Alex Miao, Jesper Buus Nielsen, Leonid Reyzin, Sophia Yakoubov

Multi-party computations

- Parties $P_1, P_2, \ldots, P_n$
- Holding inputs $x_1, x_2, \ldots, x_n$
- Want to compute a function, $f(x_1, x_2, \ldots, x_n)$
- While preserving the privacy of the inputs

Long history and many models

- Yao, GMW, BGW, CCD, **R**B (the 80's)
- Adversary: malicious, semi-honest, static, adaptive, mobile
- Computational, information theoretic
- Many beautiful results

New era

- Mega MPC, i.e. many many party computations

$N \approx$ millions

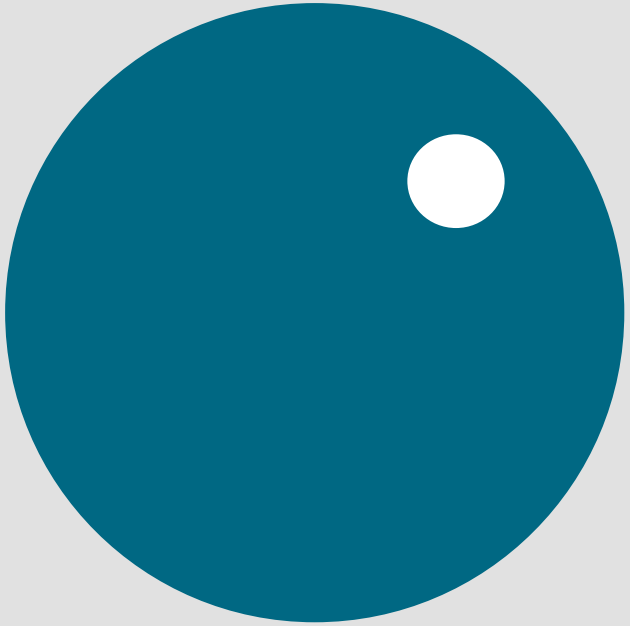
Presents a (Mega) problem

- Computation in most existing solutions is quadratic in the number of parties (at best)

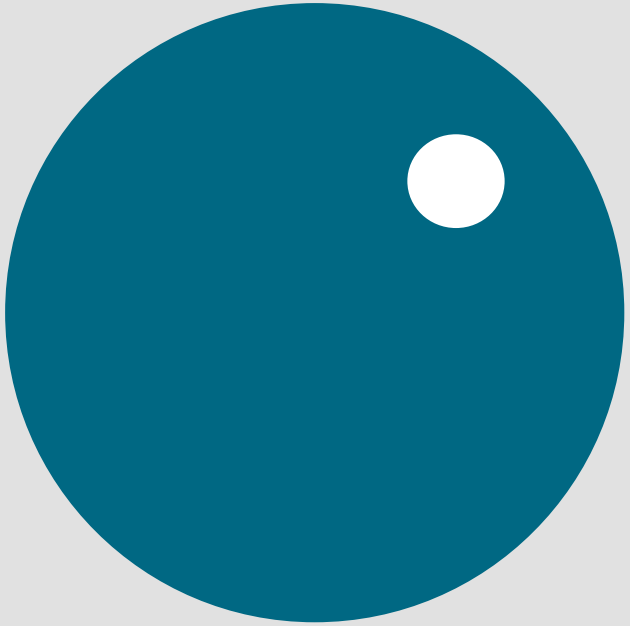
- Making MPC unrealistic in this mega setting

Approach: Small committee that computes

$n << N$

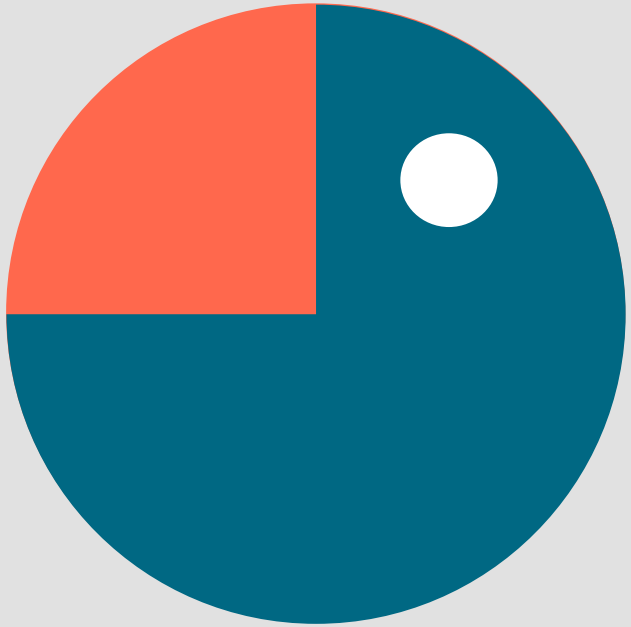


- We assume an adversary that can corrupt a fraction of the parties, e.g. N/4

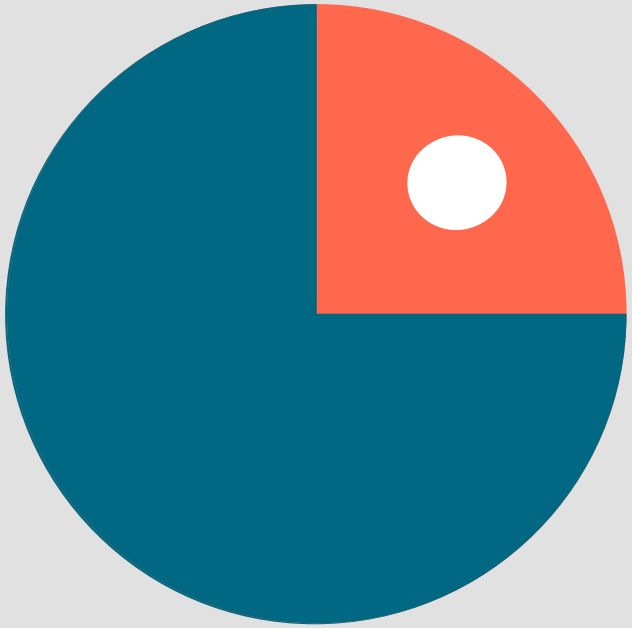


- We assume an adversary that can corrupt a fraction of the parties, e.g. N/4

- Creates another problem

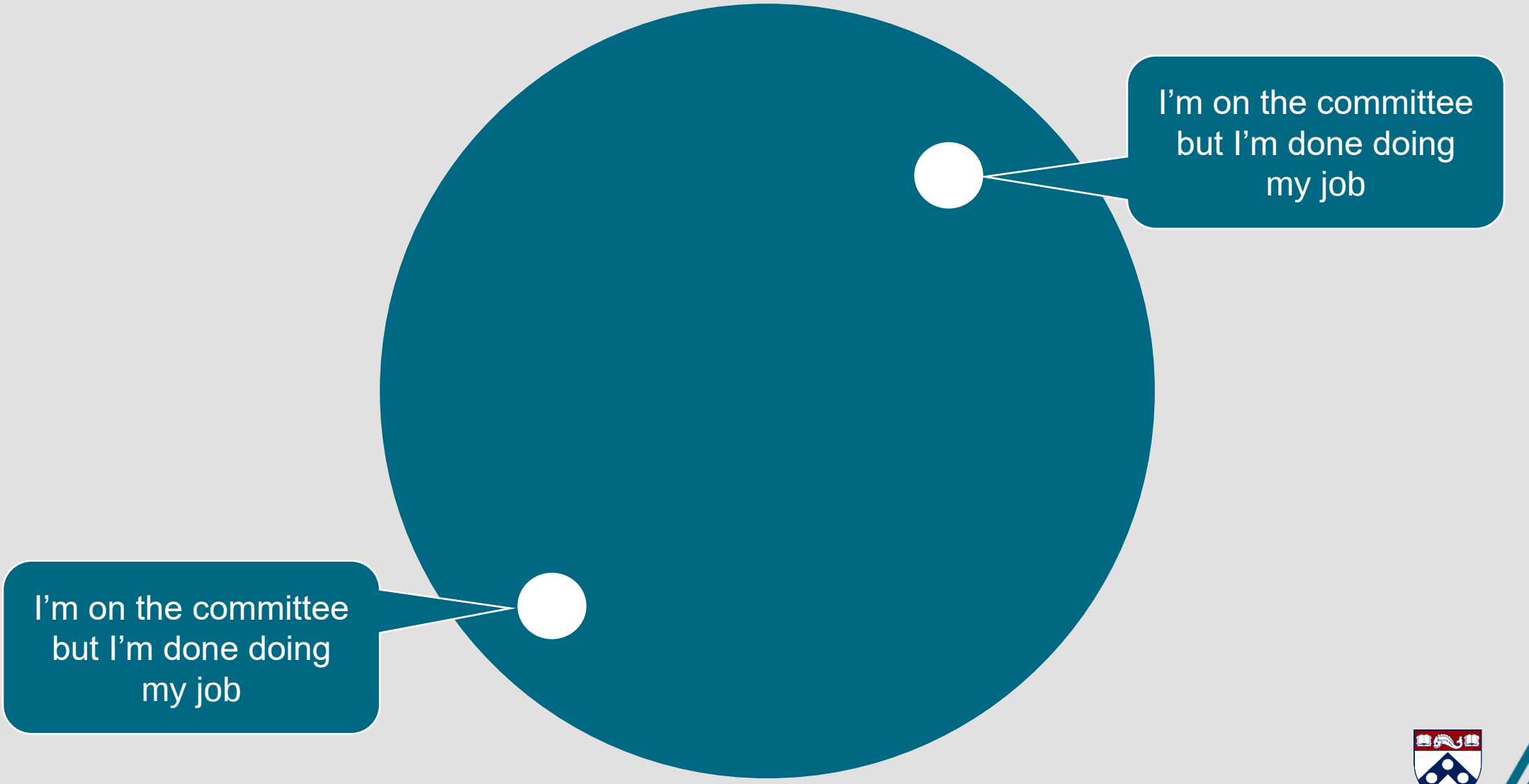


- We assume an adversary that can corrupt a fraction of the parties, e.g. N/4
- Creates another problem
- → can corrupt the full small committee



But what if…

The adversary does not know who is on the committee

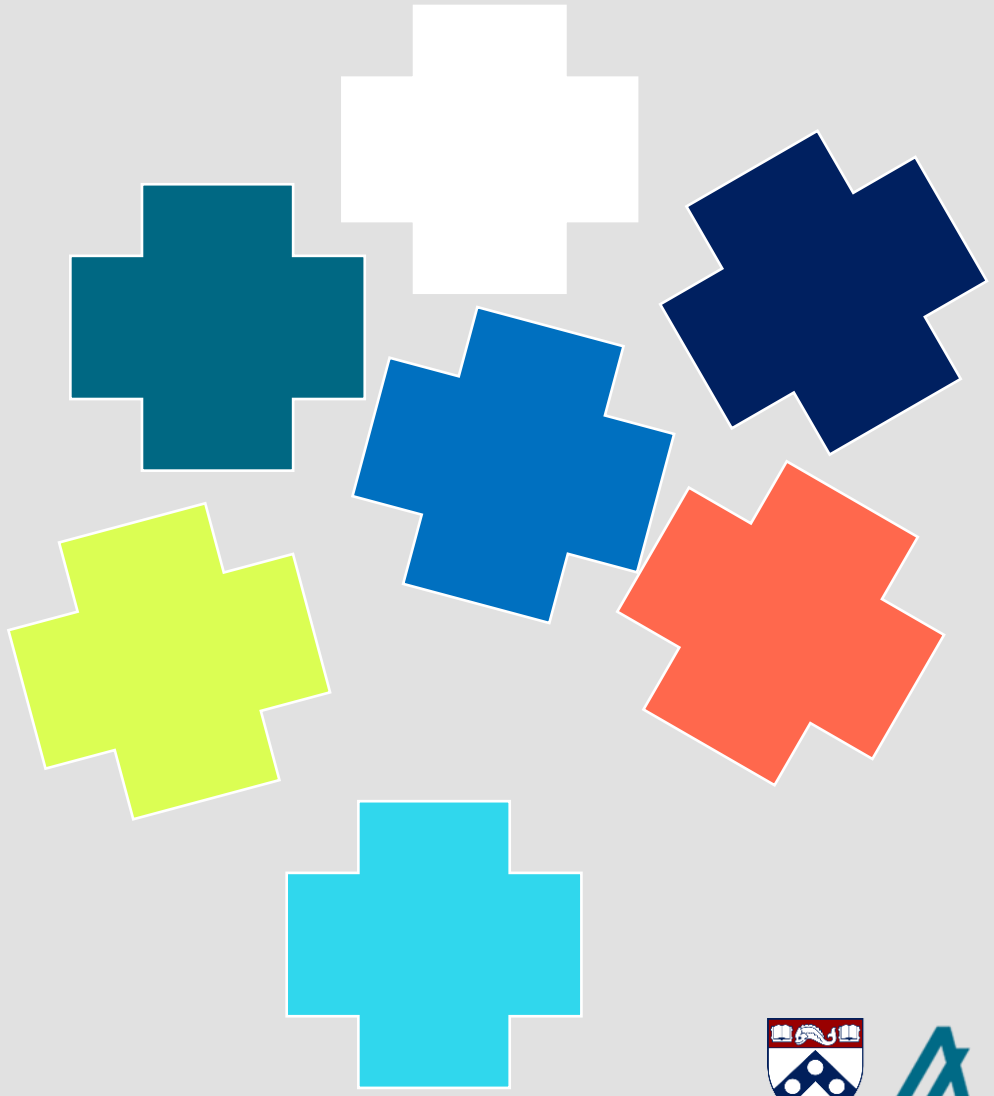

I’m on the committee but I’m done doing my job
I’m on the committee but I’m done doing my job

Self nomination

- Parties in the protocol self nominate
  - Immediately implies that the attacker does not know who is in the committee
  - (We will need to work harder later)

Hello Nakamoto (N’08)

- Self nomination: solve a puzzle



Hello Nakam (N'08)

- Self nomination: s

Solve a puzzle with proof of work

[DW'92, Back'02]

Bitcoin block suggestion

- Functionality that has: no interaction, no secret inputs
- Solve the puzzle
- Announce what the next block is

What about protocols that require communication?

- Jing-Micali ‘19 -- the Algorand protocol
  - Byzantine agreement:
    - Has interaction, multiple rounds
    - But still no secret inputs

Need better self nomination

- Self nomination has to be faster than Proof of Work
- In Proof of Stake done via Verifiable Random Function (VRF) [MRV’99]

vs

Milliseconds

In a regu[illegible]C model

Recall the problem:

If the attacker knows the committee it can corrupt all the parties

Step 1

And so on…

Player replaceability

Step 1

Step 2

Step 3

And so on…

Quite surprising, but it works

- The protocol works despite the fact that every step is executed by a different committee
- No secret information

Can we take it a step further?

What is the next step?

Protocols that:

- Have interaction, multiple rounds
- Have secret inputs

YOSO

You **O**nly **S**peak **O**nce

- Main theorem: Can compute any function in the YOSO model.
- Provide two solutions: computational and information theoretic

Hard to design in the YOSO model

- Protocols are interactive (need to speak more than once)
- Servers hold secret information

Roles

- In MPC we have parties: $P_1,\ldots,P_n$
- In YOSO the role are going to be such things as:
  - Role: shareholder in VSS of Step 5
  - Role: Party that adds two secrets in Step 8

- The protocol design will define the roles that will execute it
- Need to be able to decompose into roles that speak only once
  - Send information to a follow-up role

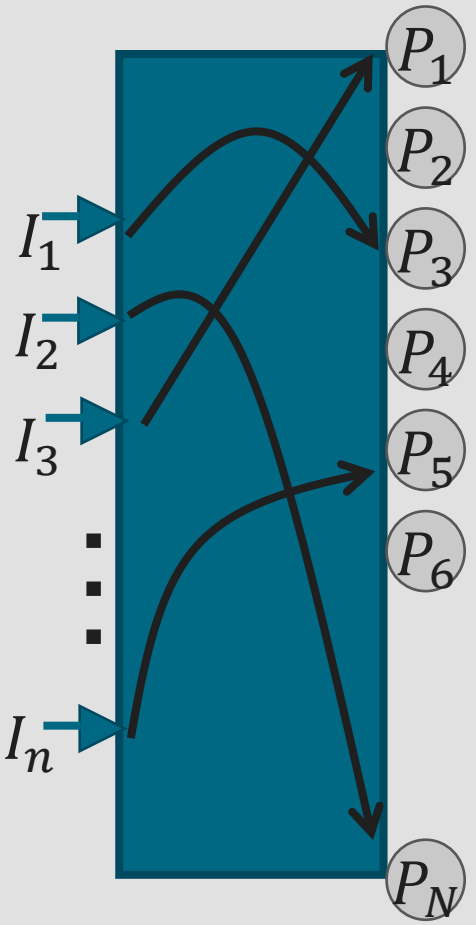
Role Assignment

- Mapping of roles to machines happens at execution time
- Mechanism for randomly and covertly assigning physical machines to roles
- Enable message delivery to future roles

MAIN TOOL

Target anonymous channels

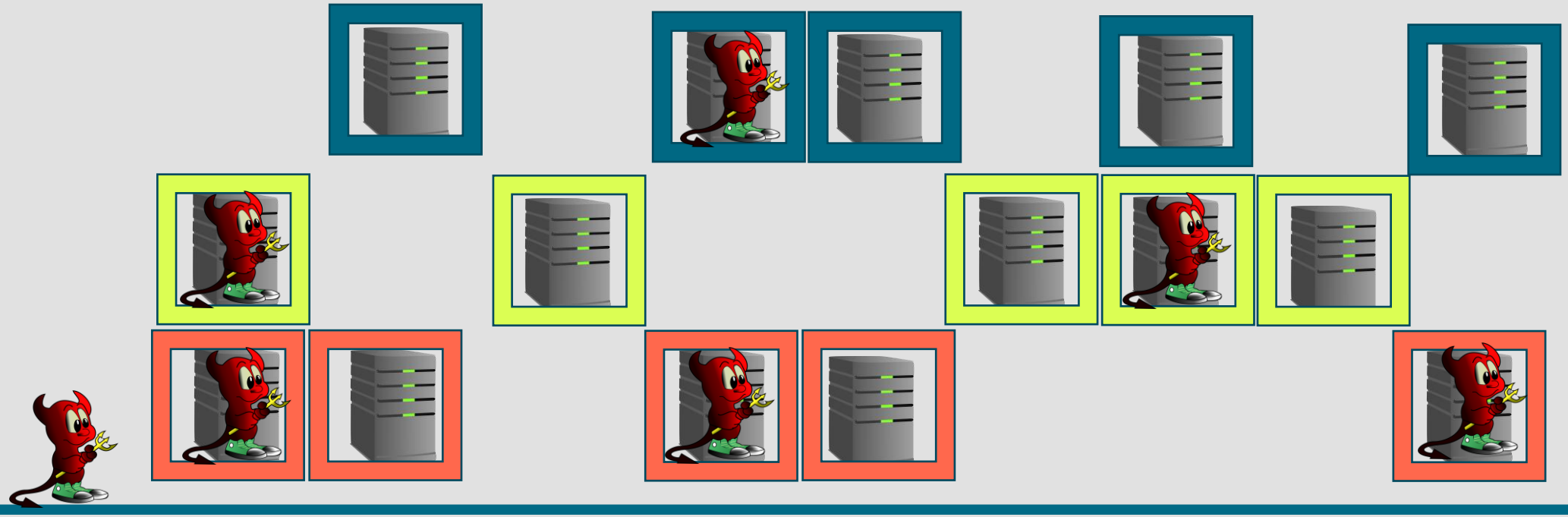
- Imagine that we had the following channels
  - n visible input ports, n hidden output ports
  - Random assignment of the output ports to an n-subset of the N nodes
- Send on the i'th input port, which represents a role, not knowing who will receive the message
- The receiver can secretly fill the role
  - It gets its secrets via encrypted messages that are sent over these channels



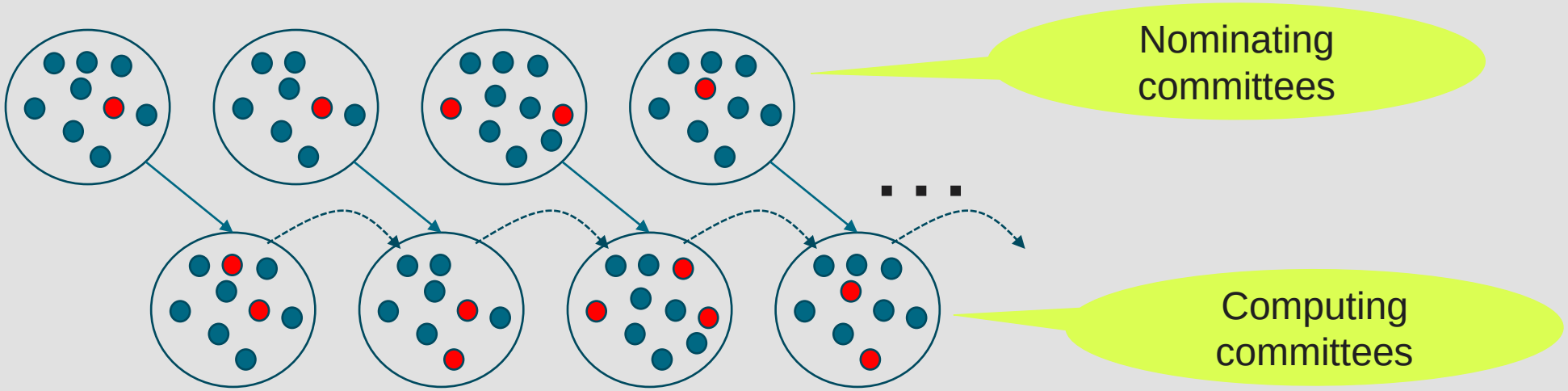
Can’t use self nomination directly

- Need to have access to the key for hidden port

Can the current committee choose the next?



Nominating committee self nominates



YOSOfying a protocol

- Can we have general techniques for converting a protocol into a protocol in the YOSO model?
  - We have some techniques
  - But some changes need to be tailored

ONE TOOL:

Speaking in the future

$s, \quad \sum s_i = s$

Time t $\quad s_1, \ldots, s_n$ (secret)

Time t+k $\quad s_1, \ldots, s_n$ (public)

- Future broadcast (for simplicity assume semi honest)
- Server, D, holds message s that needs to be broadcast later

Applications

- Threshold signatures: CA, code signing, notarization
- Key management, secure storage (incl. long-term secrets)
- (Threshold) cryptography as a service: sign, encrypt, O/PRF..
- Randomness Beacon
- Blockchain checkpoint (and cross chain)
- Blockchain as trusted party

Threshold as a service

- Key generation and refreshing in the YOSO model
- Efficient multikey/randomness generation (not in the YOSO model)

Join the YOSO model

- Improving assignment module
- Designing protocols with the YOSO model at the basis
- Specific special purpose protocols that need the YOSO model


Algorand™
FOUNDATION


Penn
University of Pennsylvania