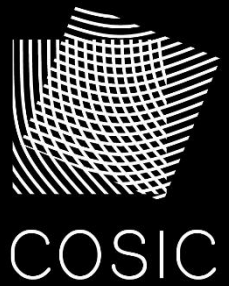
What PoS Cannot Achieve (But PoW Can)



Ren Zhang
ren@nervos.org
nirenzang

Outline

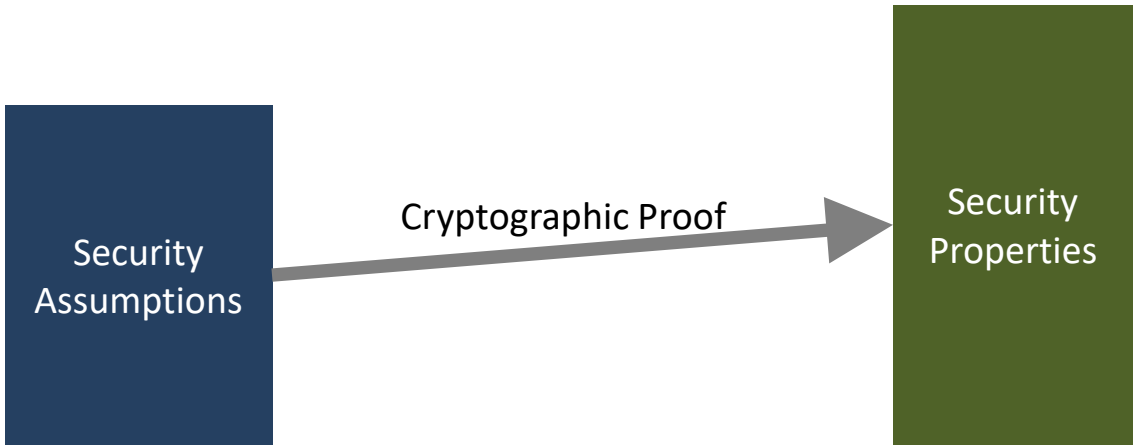
- Why Do I Love Bitcoin's Nakamoto Consensus?
- What Do We Know Before NC?
- What's New in NC?
- PoW's Boundaries?
- How Does PoS Fit into This Picture?
- Can We Do Better Than NC?

A Consensus Protocol Answers …

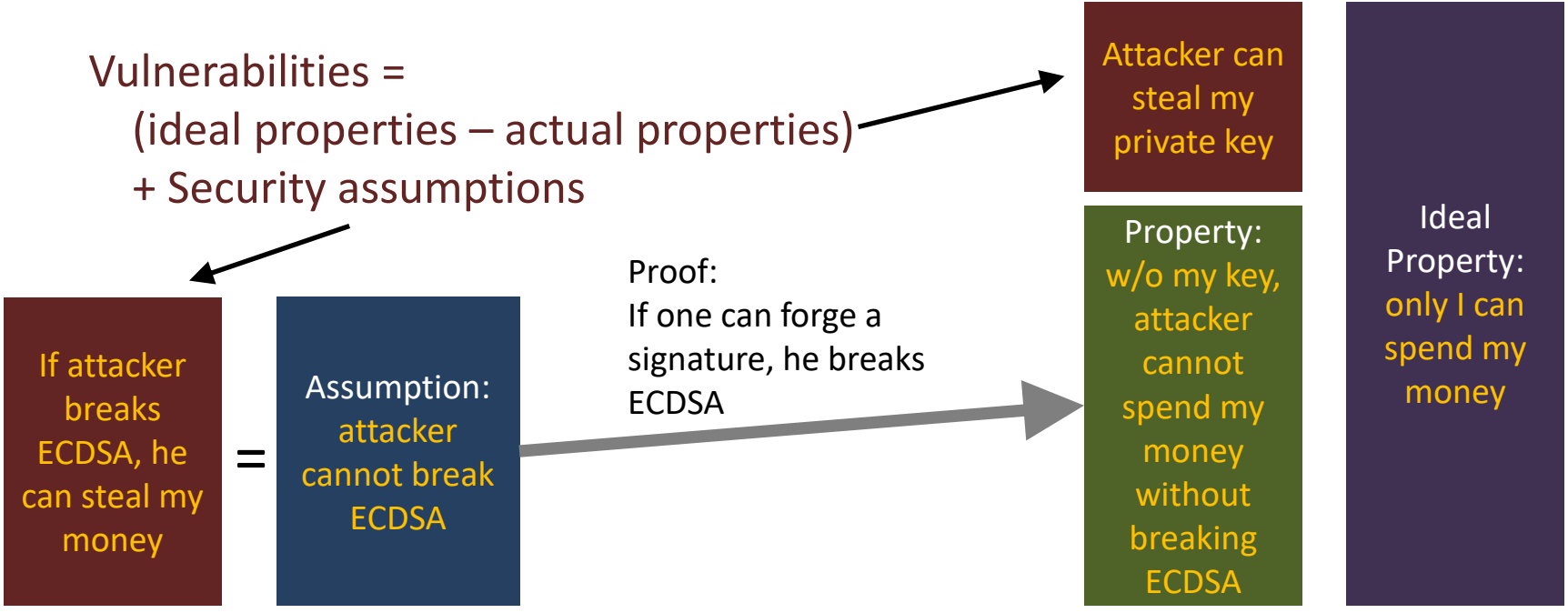
- Who can write history?

- How do we choose among conflicting versions of history?

Here is a Protocol



Here is a Protocol

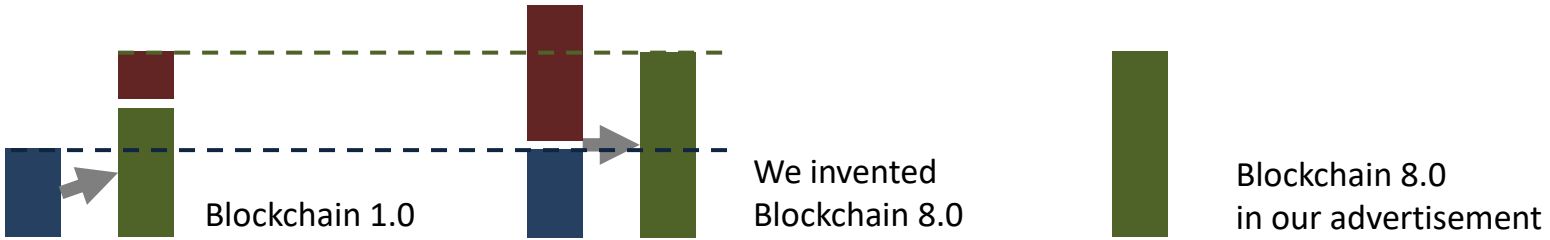


A Common Marketing Strategy

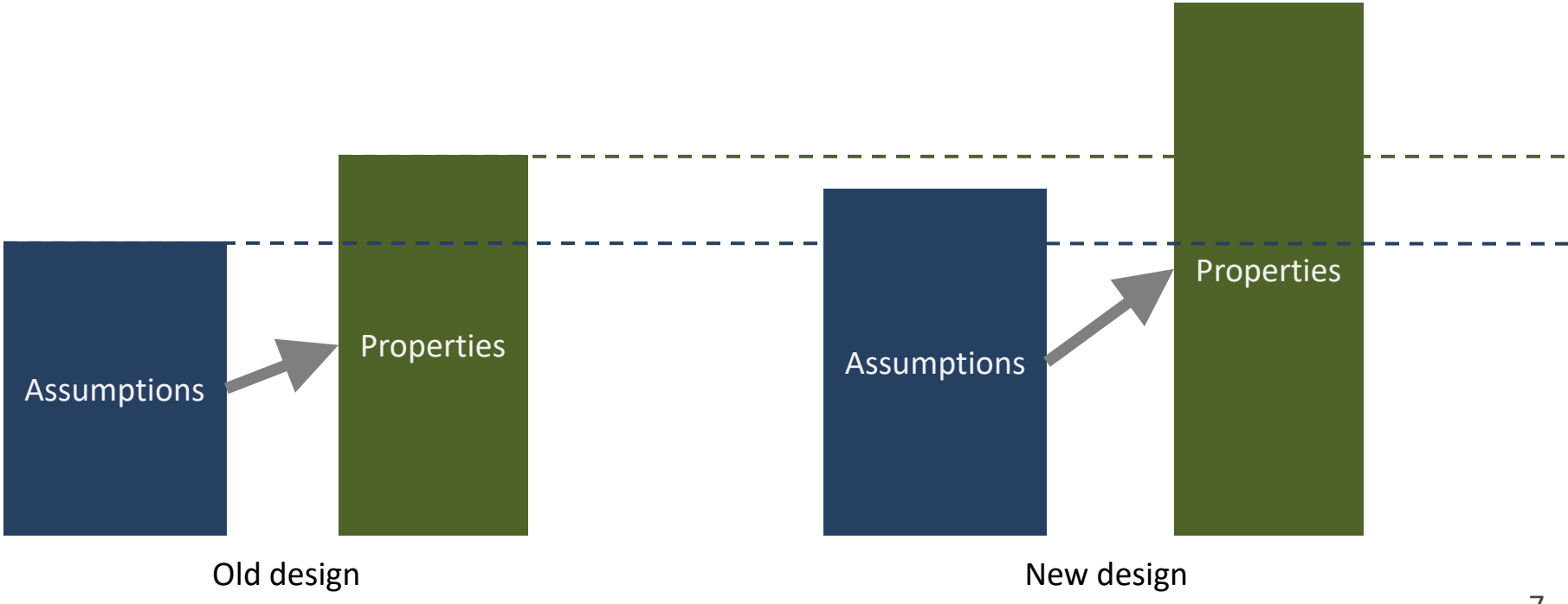
Vulnerabilities = unachieved properties + assumptions

The strategy:

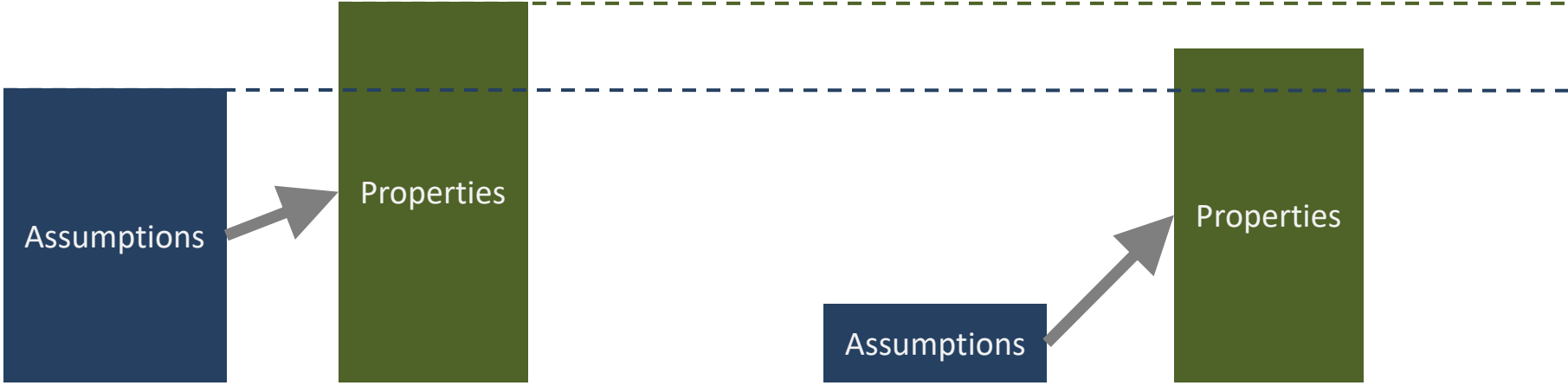
1. Achieve new properties by making stronger assumptions
2. Advertise only the security properties



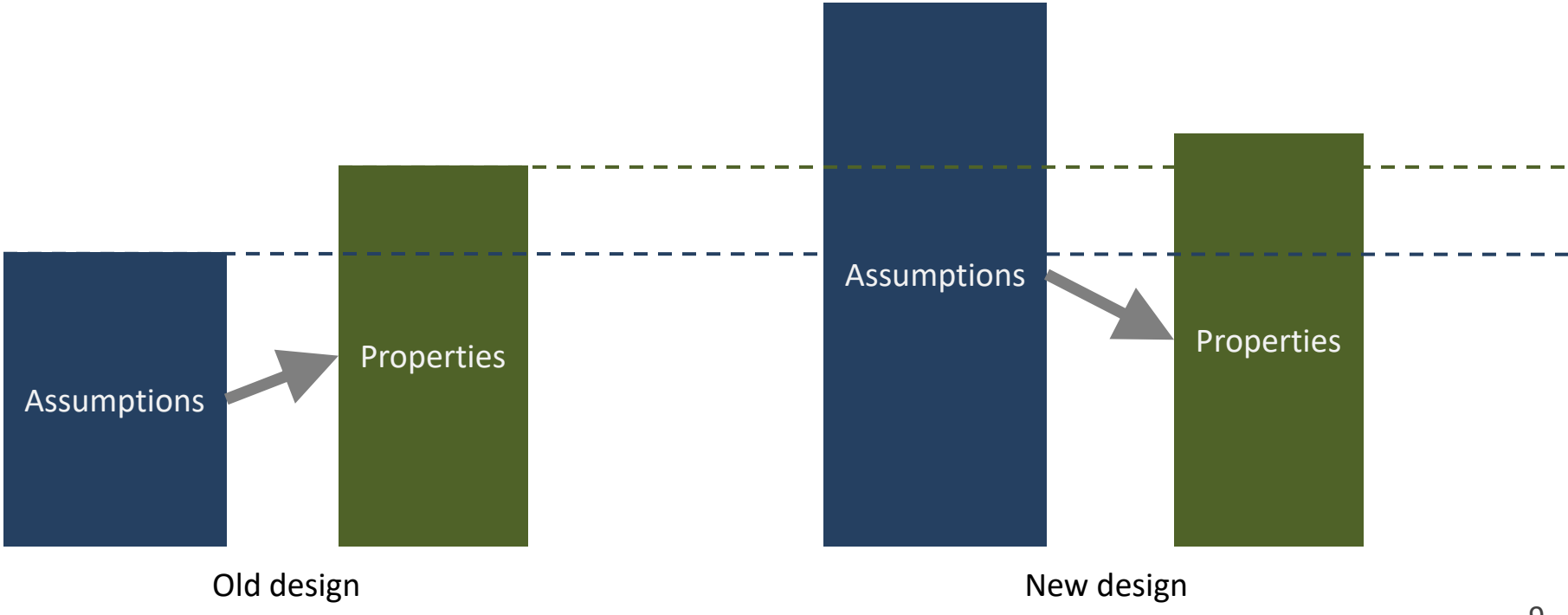
A Good Design



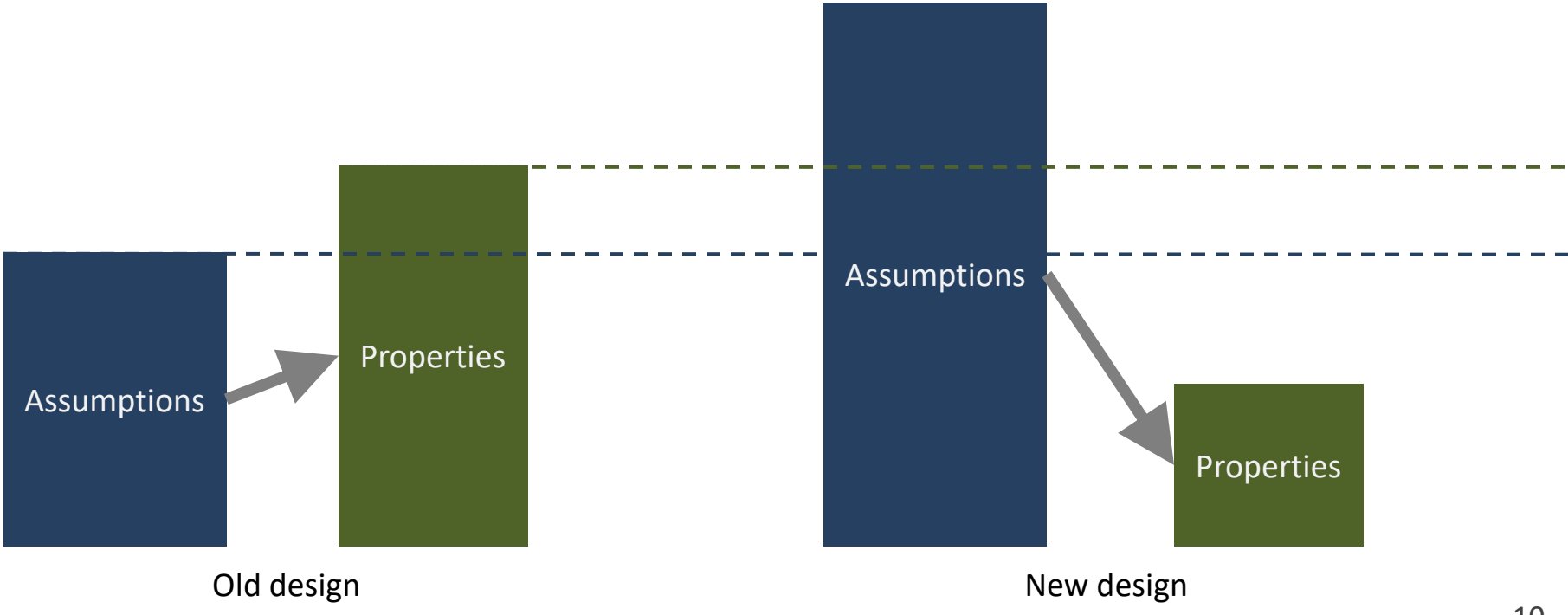
Another Good Design



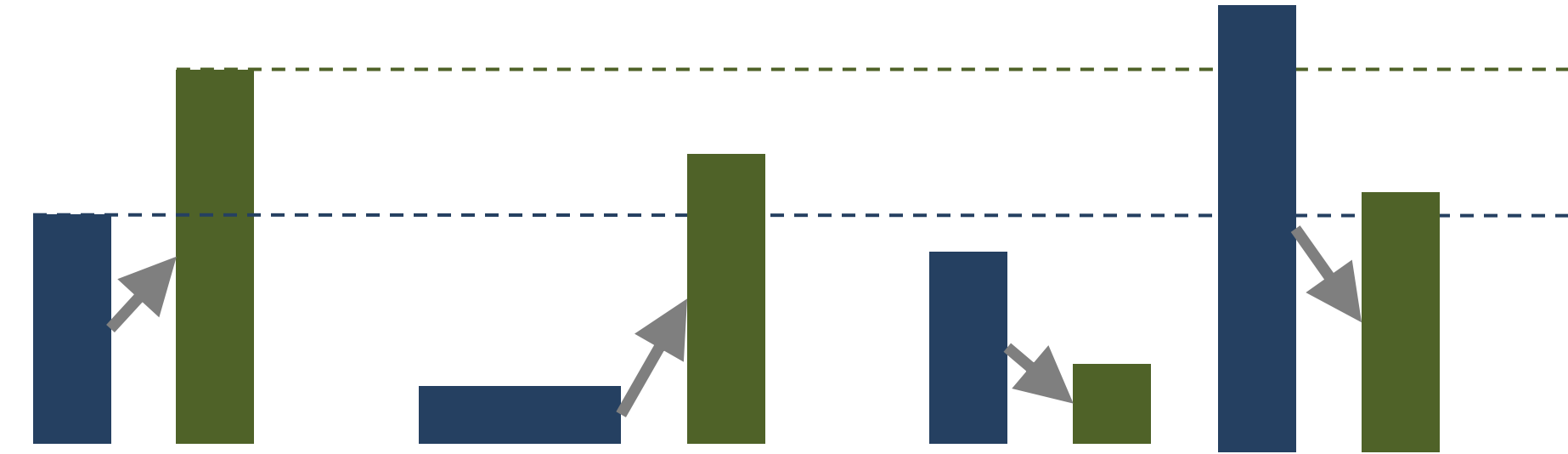
A Bad Design


Assumptions
Properties
Assumptions
Properties
Old design
New design

A Terribly Bad Design

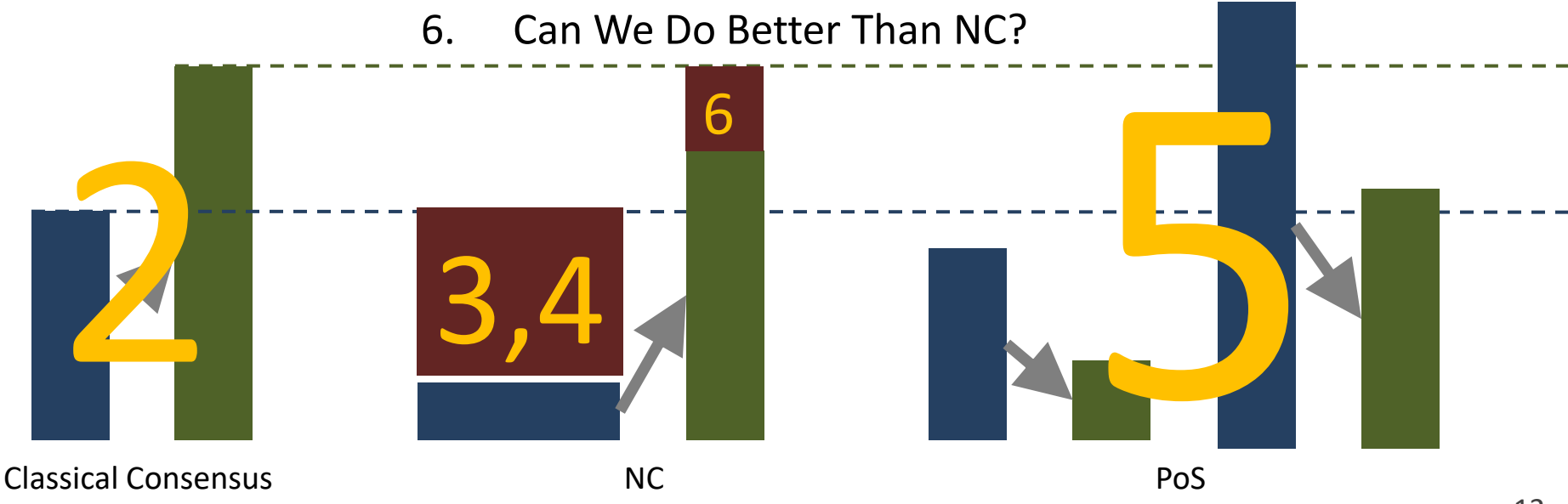


Classical Consensus, NC and PoS



Classical Consensus, NC and PoS

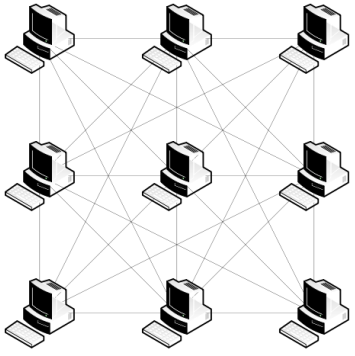
2. What Do We Know Before NC?
3. What's New in NC?
4. PoW's Boundaries?
5. How Does PoS Fit into This Picture?
6. Can We Do Better Than NC?



Outline

- Why Do I Love Bitcoin's Nakamoto Consensus?
- What Do We Know Before NC?
- What's New in NC?
- PoW's Boundaries?
- How Does PoS Fit into This Picture?
- Can We Do Better Than NC?

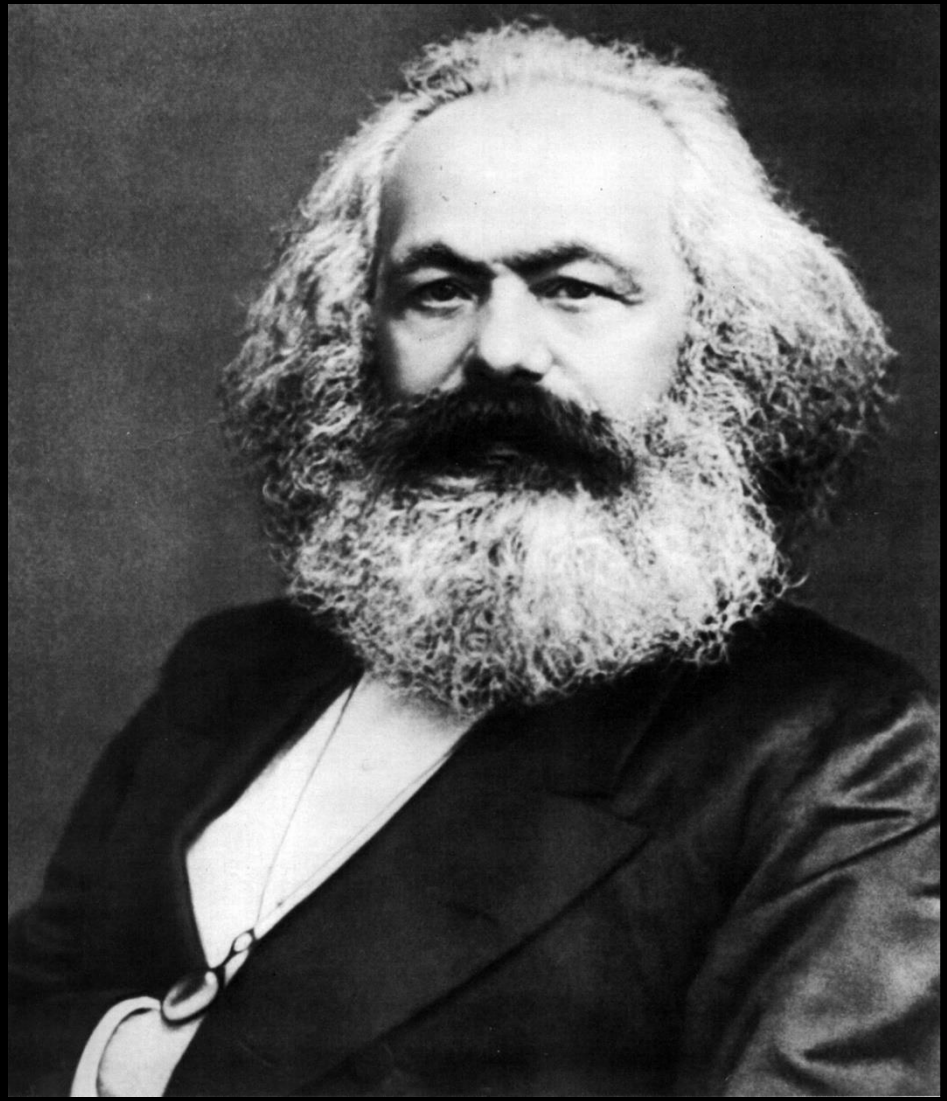
Classical/Permissioned Consensus



- # Total number of participants fixed and known by everyone
- Everyone knows everyone:
  msg sent via secure connection or with digital signature
- For late comers, the authentic history is the majority version
- Asynchronous timing assumption: network delay can be unknown
- Binary security properties: either hold or broken

Simultaneity and Mutual Exclusion

	Who produces a history step?	What happens if one produces multiple versions?
Classical	Everyone	Synchronous: get caught So >1/2 honest players is enough Asynchronous: nothing So we need >2/3 honest players



History repeats … first as tragedy, then as farce.

(on the adoption of blockchain)

Outline

- Why Do I Love Bitcoin's Nakamoto Consensus?
- What Do We Know Before NC?
- What's New in NC?
- PoW's Boundaries?
- How Does PoS Fit into This Picture?
- Can We Do Better Than NC?

Open/Permissionless Means …

Participants don't know # participants

No node ID, anonymous msg initiator

👉 No one-to-one channel, all msgs are broadcast

PoW: authentic history = most difficult to compute

?

?

Simultaneity and Mutual Exclusion

	Who produces a history step?	What happens if one produces multiple versions?
Classical	Everyone	Synchronous: get caught Asynchronous: nothing
NC (PoW)	(kind of) Everyone: more mining power, stronger chain	One cannot: each hash operation is dedicated to one history version

Outline

- Why Do I Love Bitcoin's Nakamoto Consensus?
- What Do We Know Before NC?
- What's New in NC?
- PoW's Boundaries?
- How Does PoS Fit into This Picture?
- Can We Do Better Than NC?

Limits of the Permissionless Setting

- Can we get rid of PoW?

  No. PoW is needed w/o ID (authentication)

- Can we get rid of PoW after a while?

  No. PoW must be performed indefinitely w/ 

- Is it possible to design an asynchronous protocol?

  No. The network delay must be known if # players is unknown

- Beyond “51%” assumption?

  No. Honest majority is needed w/ 

Pass, Rafael, and Elaine Shi. "Rethinking large-scale consensus."

Security Properties

Consistency

- Common prefix: two nodes *i* and *j*, two time t and t', then between $\log_{i,t}$ and $\log_{j,t'}$, one must be the other's prefix
- Self-consistency

$T_{confirm}$ Liveness

- one honest node receives *tx* at round *t*, any honest node satisfy $tx \in log$ after $t+T_{confirm}$

(All "with overwhelming probability")

PoW is necessary w/o authentication

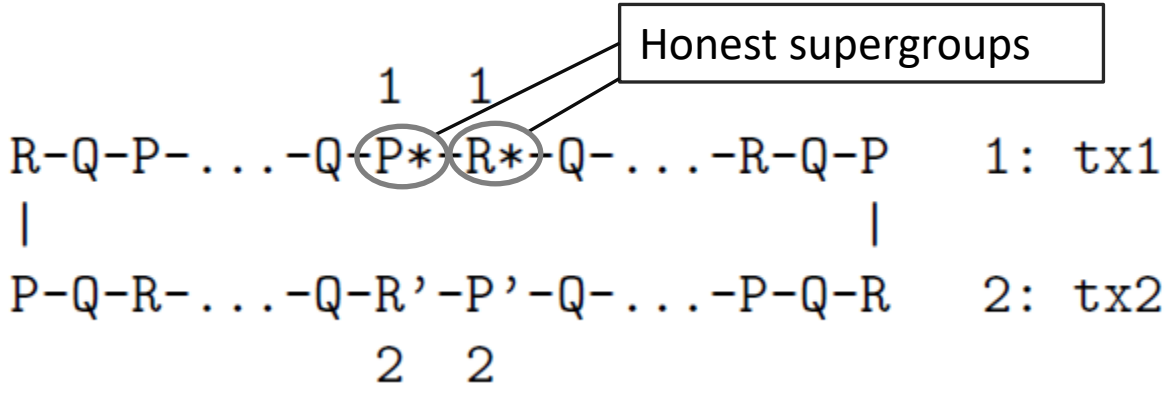
Setting

- Synchronous: $\delta=1$; N-1 honest nodes, 1 attacker
- Assuming protocol PI satisfies consistency and $T_{confirm}$ liveness

Attack

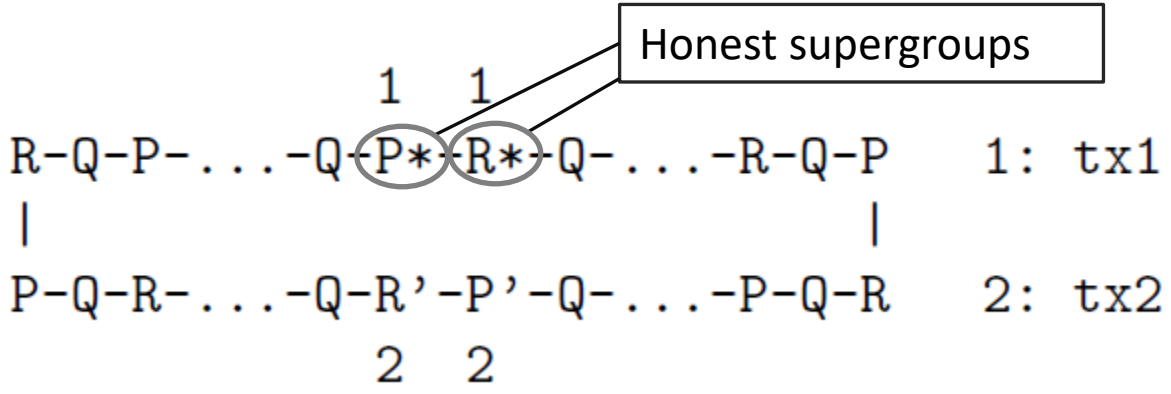
- Attacker splits honest nodes into two supergroups of (N-1)/2 each, in-group delay=0, cross-group delay=1
- Attacker creates $M > 2T_{confirm}$ supergroups of sybil nodes

PoW is necessary w/o authentication



- Each supergroup has delay 1 with two neighbors. So that the virtual distance equals the delay. Note that this doesn't violate δ=1
- P* and R* receives tx_1 at the beginning, R' and P' receives tx_2, all other supergroups receive noise
- Execute PI; note that PI doesn't know which supergroups are honest

PoW is necessary w/o authentication



- Claim 1: if any supergroup outputs tx_b before tx_{1-b}, all should do so (by consistency)
- Claim 2: P* and R* output tx_1 first, R' and P' output tx_2 first (by $T_{confirm}$ liveness)

PoW must be performed indefinitely w/ late spawning

Setting

- Attacker 1% mining power; synchronous
- PoW stops after T

Attack

- Attacker continues to construct an alternative history
- After 101T, two histories have the same amount of PoW, late comers cannot tell which one is authentic

Honest majority is needed w/ late spawning

Setting

- Attacker 50% mining power; synchronous

Attack

- Attacker construct an alternative history
- Late comers cannot tell which one is authentic

The network delay must be known if # players unknown

Setting

- Attacker 0% mining power; asynchronous; 2N honest players
- Protocol PI satisfies consistency and $T_{confirm}$ liveness

Attack

- Evenly split honest players into two groups, in-group delay is zero, cross-group $\delta > T_{confirm}$
- one group receives tx1 at the beginning, the other receives tx2; a group cannot tell between
  - the other group doesn't exist
  - the other group hasn't received our message

Open/Permissionless Means …

Participants don't know # participants

👉 Cannot be asynchronous

No node ID, all msgs are broadcast

👉 Some scarce resource is necessary, e.g. computational power

PoW: authentic history = most difficult to compute

👉 PoW never stops, <50% attacker

?

Security Properties Look Different

Permissioned Consensus

- Security properties are binary
  - N < 3f+1: broken
  - N ≥ 3f+1: security holds
- No incentive

Permissionless Consensus

- Security properties are metrics
  - Attacker > 50%: broken
  - Attacker < 50%: not completely broken
- With incentive, but easier to break than security
  - correct incentive ≠ secure
  - wrong incentive 👉 insecure

Outline

- Why Do I Love Bitcoin's Nakamoto Consensus?
- What Do We Know Before NC?
- What's New in NC?
- PoW's Boundaries?
- How Does PoS Fit into This Picture?
- Can We Do Better Than NC?

Assumptions that Raise My Alarm

- Trusted setup
- Everyone is always online
- A globally synchronous clock
- People safely destroy their used keys
- The attacker's only goal is money

Open/Permissionless Means …

Participants don't know # participants

👉 PoW: Cannot be asynchronous

👉 PoS: Must be synchronous

Algorand	*… Algorand makes a "strong synchrony" assumption that most honest users (e.g., 95%) can send messages that will be received by most other honest users (e.g., 95%) within a known time bound…*
Ouroboros	*… all players are assumed to have weakly-synchrononized clocks (all clocks are within Δ of the "real time") and all messages … are delivered within Δ time…*
Sleepy	*Players are equipped with (roughly synchronized) clocks…*

When The Clocks are Not Synchronized…

Attack ①

- Publish “on the point”
  - On time for some nodes
  - Late for the others

Does it break the consensus?

Does it facilitate other attacks?

Open/Permissionless Means …

No node ID, all msgs are broadcast

👉 Some scarce resource is necessary for sybil resistance

PoW: computational power

PoS: coins

Simultaneity and Mutual Exclusion

	Who produces a history step?	What happens if one produces multiple versions?
Classical	Everyone	Synchronous: get caught Asynchronous: nothing
NC (PoW)	Everyone, kind of	One cannot
A typical PoS	One coin holder	Nothing, as it is undetectable

Three Implications

As the history step is produced by one coin holder, there is no one to check when it is produced

👉 Synchronous model doesn't help here

The resource is not consumed in producing history

👉 Producing conflicting histories is cost-free and undetectable

👉 Cannot distinguish between past possession and current possession

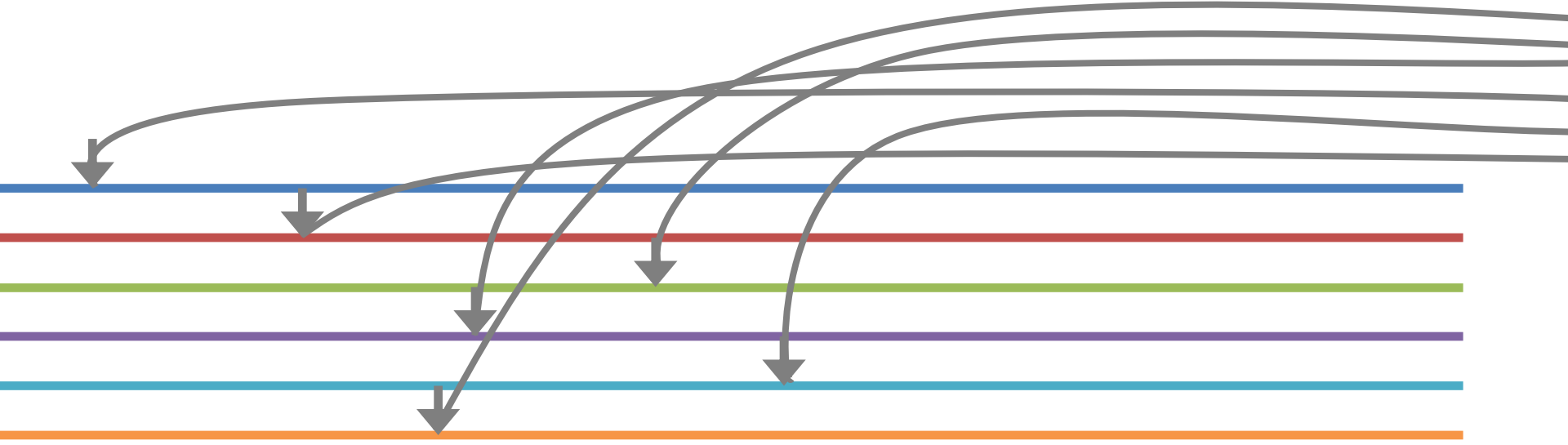
A Dilemma

- If the latest few blocks can influence the next block producer
  - ② grinding attack
  - ③ undetectable nothing-at-stake attack
- If the latest block cannot influence the next block producer
  - ④ predictable selfish mining & double-spending

Brown-Cohen et al. Formal Barriers to Longest-Chain Proof-of-Stake Protocols

⑤ Long Range Attack

- Once the attacker possesses a coin / gets a private key, he gets the right to produce alternative history with it forever



Open/Permissionless Means …

PoW: authentic history = most difficult to compute

- 👉 PoW never stops, <50% attacker
- 👉 PoS cannot tell the authentic history

Are These Attacks Practical?

In a system that:

- Must be synchronous
- No one to check when a block is produced
- Producing conflicting histories is cost-free and undetectable
- Cannot distinguish between past and current possession
- Cannot tell the authentic history

are these attacks exhaustive?

How to Address the Attacks?

Eliminate conflicting histories

- Remove attack incentive
- Make the attack impossible
- Make the attack computationally infeasible

How to Address the Attacks?

no incentive
impossible
infeasible

Attackers have different incentives;
no reward scheme discourages all of them

- Rewarding the “losers”

👉 encourages attack attempts

- Punishing forks

👉 gives the attacker a tool to punish the others

How to Address the Attacks?

no incentive

impossible

infeasible

- Consensus on checkpoints

👉 Why use PoS if you already have a consensus protocol?

- Everyone is always online

👉 What do we say about new assumptions?

- Some special participants are always online

👉 Classical consensus + trusted setup

How to Address the Attacks?

no incentive
impossible
infeasible

- Verifiable delay function
- 👉 Hello PoW

Outline

- Why Do I Love Bitcoin's Nakamoto Consensus?
- What Do We Know Before NC?
- What's New in NC?
- PoW's Boundaries?
- How Does PoS Fit into This Picture?
- Can We Do Better Than NC?

Better-Than-NC PoW?

- See my next paper
- And my next^3 paper
- And my next^5 paper
- We'll try to implement these ideas in Nervos ckb

Short Conclusion

- Tell anyone that claims to have a perfectly secure consensus protocol…


ACADEMIA IS WATCHING YOU

Thank you!

Ren Zhang
ren@nervos.org
nirenzang