Proof of Honesty: Coalition-Proof Blockchain Validation without Proof of Work or Stake

John P. Conley
Vanderbilt University

Sirius Group Meeting

Cheriton School of Computer Science

University of Waterloo

November 2018

What is Blockchain?

- A data system

  Data System $\supset$ State Machine Replication System $\supset$

  Distributed Ledger Technology $\supset$ Blockchain

- A consensus system
  - Proof of Work
  - Proof of Stake
  - Proof of Authority
  - Governance
  - .....

Data Systems

Data systems require:

- Hardware
- Communications infrastructure
- Network and communications protocols
- Data formats and standards
- Lots of other stuff

Consensus Mechanisms

Consensus mechanisms have the two main jobs:

- Establishing a canonical version of the current state of the data
- Making sure the canonical view is correct

In addition, it would be nice if:

- All copies of the database are identical or synchronize quickly
- All copies of the database are available for use
- Altering the data in unauthorized ways is difficult or impossible

Honesty and Consistency

The CAP Theorem tells us:

No distributed data store can simultaneously provide more than two out of the following three:

- Consistency: Every read receives the most recent write or an error
- Availability: Every request receives a (non-error) response – without the guarantee that it contains the most recent write
- Partition tolerance: The system continues to operate despite an arbitrary number of messages being dropped (or delayed) by the network between nodes.

If we could have all three, we would have a canonical view of the state of the database.

Note, however, that have the in most recent write does not imply that the write is correct or honest.

Honesty and Consistency are logically different properties

Honesty and Consistency

PoW:

- Longest chain rule to get canonicalness
- Recursive Hashing of block to make certain rewrites detectable
- Hashing/nonce search to make rewrites difficult
- Honesty? 50% BFT

PoS:

- 2/3 stake weighted voting to get canonicalness
- Recursive Hashing of block to make certain rewrites detectable
- Honesty? 33% BFT

BFT and Security

Two mining pools own more than half the hashing power of the bitcoin network.

It would cost somewhere between $1B and $3B for a bad actor starting from scratch to mount a 51% attack on Bitcoin.

Ethereum and other blockchains would cost much less to attack, and if an attacker already owns enough hardware, it is even cheaper.

PoS is even cheaper to attack

Who would do such a thing?

- USA – Stop tax evaders, money launderers, and criminals
- China or Russia – Cyber warfare.
- North Korea – Just for fun.
- Canada?

We are going to make a key assumption for this talk:

Perfect Nonmanipulable Networks (PNN): The network that nodes use to communicate is fully functional in the sense that it allows all nodes and users to send messages to one another without latency. In addition, if any node fails to send a message required by protocol or falsely claims that a message was not received, it is provable.

This is similar to Tendermint's Gossip Communication and Liveness assumptions.

PNN drives canonicalness in the PoH protocol describe below.

PNN is a completely ridiculous assumption

Achieving canonicalness without PNN is possible with certain network and protocol elements, however, this is a subject for a different day.

Algorithmic Game Theory

The perspectives that economists and computer scientists bring to the table are different, and each have their value.

Blockchain protocols are have their roots in **algorithmic game theory** which adapts traditional noncooperative game theory for use in computational environments.

Agents using protocols without a complete understanding of how they work may have difficulty determining fully optimal actions. As a consequence, agents are often modeled as following *ad hoc* behavior patterns.

For example, agents might be assumed to be either **honest** or **malicious-type** players since fully rational play may exceed their cognitive limitations.

Algorithmic approaches tend to pay less attention to certain other elements of games and mechanisms:

- Multiple equilibria.
- Refinements of Nash equilibrium.
- Effects of information and belief structures on equilibrium in sequential games.

Mechanisms vs. Protocols

The problems typically addressed by protocol builders and economic mechanism designers are also different in at least two important ways.

First:

Mechanisms: Agents have private information.

Protocols: The truthfulness of validators is externally observable and provable.

Second:

Mechanisms: The designer generally sets up a game in which he imposes both a strategy space and a payoff function. If agents participate, they have no choice but to choose one of the permitted strategies and receive payoffs as determined by the designer.

Protocols:The builder also sets up rules that are supposed to be followed and a specific set of messages and actions that are allowed by protocol. Validators, however, can send any messages they wish. Rewards and punishments exist only on/in the blockchain being validated and must be written and agreed upon by the validators themselves.

Where this Bites

Honesty is endogenous

- Dishonest ≠ Broken

Multiple equilibrium

- Right side/left side
- All honest/all dishonest

Information and expectations are critically important

- Battle of the sexes
- ETH worth $1000 or $100
- Increasing mining rewards

Equilibrium definition

- Nash (example: prisoners' dilemma)
- Dominant Strategy
- Coalition Proof

Mechanism Design for Blockchain

We propose a mechanism design solution to blockchain validation consisting of two main elements:

Proof of Honest (PoH)

Catastrophic Dissent Mechanism (CDM)

Note: The accompanying paper also describes a hub and spoke network topology, a message space, and work-flow for validating nodes in detail. We give a brief sketch of this below.

1. Users choose a node and send it a transaction.
2. Each node accumulates transactions until the block currently under construction is complete, verified, and committed to the existing blockchain.
3. One node is chosen randomly to act as hub for the next block.
4. Nodes send transactions bundles and a hash of their Current Ledger State (CLS) to the hub.
5. The Hub collects the transactions received from the nodes and sends this back to each node as a bundle of candidate transactions.
6. All nodes (including the hub) start with the same CLS and the same bundle of candidate transactions. These are used in combination with the business logic of the chain to check the validity of each candidate transaction.
7. The set of valid transactions are put into a block which is committed by each node to its version of the chain and the CLS is updated.
8. All nodes check the hash of the CLS of other nodes. If any CLS hash is different then the node initiates an audit using the CDM.
9. The process begins again with each node returning to step 3.

PoH Works as Follows

- Chain Discovery: Users discover a given blockchain as well as any forks that might exist. In practice, users might be directed to a node that validates an application that a user wishes to use or finds a node though a web search or consulting a forum.
- Honesty Checking: Users inspect the chain and its forks, if any, to any degree that they wish to determine the honesty of the nodes and the validity of the chain or forks. This is done automatically via user client software.
- Transaction Creation: Users choose a node and send it a transaction.
- Block Writing and Commitment: Nodes either follow or don't follow workflow and protocol rules. Eventually, nodes create and commit a block and update the leader as they see fit, honestly or dishonestly.

Note that PoH is **User Centric**: Users determine block and ledger validity

PoW, PoS, and other protocols are **Node Centric**: If a consensus of nodes agrees, then a block and ledger are valid

99% BFT

The key feature of PoH is that if a single honest node exists, it constructs an honest chain. Users can then discover this honest node and chose to send transactions only to this node.

This simple idea produces blockchain with a BFT of 99%. It does not matter how many dishonest nodes exist. If there is at least one honest node, no tokens can ever be stolen from rational, honest users.

Dishonest forks written by the dishonest nodes ends up being a **fictional ledger** in which dishonest nodes and users steal tokens from one another. No honest user has any incentive to transact on this ledger and so it ends up being **orphaned**.

A Unanimity Game

Of course, 99% BFT is fairly large improvement over existing consensus protocols that offer 50% BFT at best.

What if all validating nodes are dishonest? To bring achieve 100% BFT, we must add another element to the mechanism based in a kind of unanimity game.

A Unanimity Game

- Agents are offered a chance to play a game in exchange for a one dollar admission fee.
- Each player who pays the fee is sent to a room where a name is written on the wall. Players are asked to write this name on a piece of paper.
- The papers are then gathered and compared. If they all have the same name, then each player is paid two dollars.
- If there is any disagreement about the name, all players get zero (which gives each a net payoff of negative one dollar).

Note that there are many Nash equilibrium including truth-telling is a Nash equilibrium.

This is a feature of most consensus protocols as well.

A Unanimity Game with Auditing

Add the following:

- If all agents write the same name, the named individual gets $1000 (like a transaction on a blockchain ledger).
- All agents sign their papers.
- If there is disagreement about the name, then the door to the room is opened, and the name on the wall is read.
- Any player who wrote down the correct name gets $2 of plus an equal share of a $1000 bonus.
- Players who wrote down an incorrect name receive nothing.

Equilibrium of a Unanimity Game with Auditing

Truth-telling is the unique coalition-proof equilibrium. (implementation)

Suppose all agents tried to collude and write down one of their own names and then share the $1000 received.

Any single agent who defected and called for an audit would get the $1000 bonus which is more than an equal share of the $1000 that the coalition tries to steal.

Knowing that at least on agent will certainly defect, the other agents will abandon the attempt to collude, and so truth-telling is the only equilibrium that remains.

The Catastrophic Dissent Mechanism

We assume that:

- All nodes are dishonest (at least self-interested)
- All nodes are able to communicate and coordinate, and trust one another not to deviate from agreements to be dishonest and thus share the resulting profits.
- Many of the nodes are Sybils run by the same agent.

We need some notation to formally define the game:

V – the total value that dishonest nodes think they can move off-chain.

N – number of validating nodes.

A – number of independent agents running validating nodes.

H – number of node who decide to honestly report the bad behavior.

D – number of nodes deciding to follow the conspiracy and behave dishonestly.

G – amount held in the GBB of each node.

T – Transactions processing payment to nodes who behave honestly.

The Game

Agents:

$n \in \{1 \ldots N\} \equiv \mathcal{N}$ – validating nodes

Strategies:

$v_n \in \mathbb{R}_+^1 \; \forall n \in \{1 \ldots N\}$

Payoff Function:

$F_n : \mathbb{R}_+^N \rightarrow \mathbb{R}^1 \; \forall n \in \{1 \ldots N\}$

We interpret v_n as the amount that agent n proposes to steal and move off-chain.

If $v_n = 0$, the node is behaving honestly and also reports any thefts by other nodes.

If $\sum_n v_n > V$, then we interpret this as coordination failure which results in all dishonest nodes being unsuccessful in their attempts to steal tokens.

The Game

Thus:

$F_n(v_{1,}\ldots,v_n,\ldots v_N)=$

T	if $v_n=0\,\forall\, all\, n\in\{1\ldots N\}$	(all nodes honest)
$T+\frac{GD}{H}$	if $v_n=0$ and $\exists m\in\mathcal{N}$ such that $v_m>0$	(honest with dishonest nodes)
$-G$	if $v_n>0$ and $\exists m\in\mathcal{N}$ such that $v_m=0$	(dishonest with honest nodes)
$-G$	if $v_n>0\,\forall n\in\{1\ldots N\}$ and $\sum_n v_n>V$	(dishonest coordination failure)
v_n	if $v_n>0\,\forall n\in\{1\ldots N\}$ and $\sum_n v_n\geq V$	(dishonest with coordination)

The result

Note that 100% BFT means that even if 100% of the node are dishonest then blockchain is correctly validated.

This statement is a bit incomplete since it treats dishonesty as if it were an innate characteristic.

It does not describe what motivates agents to be dishonest or what incentive, information, or belief structures might motivate them to follow protocol despite their “dishonesty”.

We therefore propose a notion of blockchain security based in game theory.

Strategically Provable Security (SPS): A Blockchain has Strategically Provable Security if truth-telling and faithful execution of the protocol by all the validating nodes is the only coalition-proof equilibrium.

This leads to the major result of the paper:

Claim: *If* $G > V/N(A-1)$ *then the blockchain satisfies SPS*

Catastrophic Recovery Protocol

Note that the size of the GBB (G) needed depends much a unanimous conspiracy steal and distribute to its members (V)

The smaller the residual value, the less benefit there is from forming such a conspiracy.

To reduce this value, we add another element to the CDM called the Catastrophic Recovery Procedure (CRP).

The CRP allows users to pick up the validation of the blockchain from the last honest block if there are not honest nodes and so no honest chain.

If other users agree with their assessment, a new Active Node List is formed and the chain continues honestly.

The CRP is costly, confusing, and extremely undesirable. However, it prevents a coalition of all nodes from being able to force users to choose between accepting their dishonesty or walking away from their accounts.

There are a number of ways that this might be implemented, so we will not go into details here.

Conclusion

Existing blockchain consensus protocols

- Have multiple equilibria
- Offer 50% BFT at best

This paper proposed a new mechanism in which

- Validating nodes are anonymous.
- Any agent can join the validation network
- There are no central points of failure
- PoH consensus gives 99% BFT
- CDM provides SPS, that is, makes truth-telling the unique coalition-proof equilibrium

Thanks very much!

More details can be found on my webpage:

http://www.jpconley.com

or at the Geeq Project web page:

https://geeq.io/