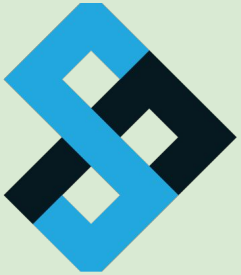


SmartPool:

practical decentralized pool mining

Loi Luu, Yaron Velner, **Jason Teutsch**, and Prateek Saxena

August 18, 2017

NUS
National University
of Singapore

TrueBit

האוניברסיטה העברית בירושלים
THE HEBREW UNIVERSITY OF JERUSALEM

Mining pools

Miners' role in cryptocurrencies

Definition: A *cryptocurrency* is a decentralized network which maintains a permanent, public ledger. The ledger is called a *blockchain*.

- Bitcoin's blockchain contains financial transactions.
- Ethereum's blockchain contains stateful programs called "smart contracts."

Anonymous *miners* maintain the integrity of the blockchain in exchange for protocol-generated *block rewards*.

The mining process

Miners race to solve a hard, computational problem.

1. The first miner who successfully solves this problem broadcasts his answer to the network.
2. The miner appends a *block* of new transactions to the blockchain and receives a block reward.
3. The race begins again on top of this new block.

Block Problem: *Let $D \geq 0$ be some fixed "difficulty." Find a* nonce *such that*

$$\mathsf{hash}(\mathrm{block}, \mathrm{nonce}, \mathrm{data}) \leq D$$

A block satisfying the above property is *valid*.

Bitcoin block rewards as of August 2017 are 12.5 BTC (∼43,000 USD).

Pooled mining

The probability of a given miner finding the next block is:

$$\frac{\text{miner's CPU power}}{\text{network's total mining power}}.$$

New blocks occur, on average, every 10 minutes on Bitcoin and 15--20 seconds on Ethereum, but:

- an ASIC hardware may take years to mine a single block, and
- most miners prefer more steady income.

Therefore miners "join hands."

***Mining pools*:**

- share CPU power and rewards among miners,
- reduce reward variance, and
- add security through increased participation.

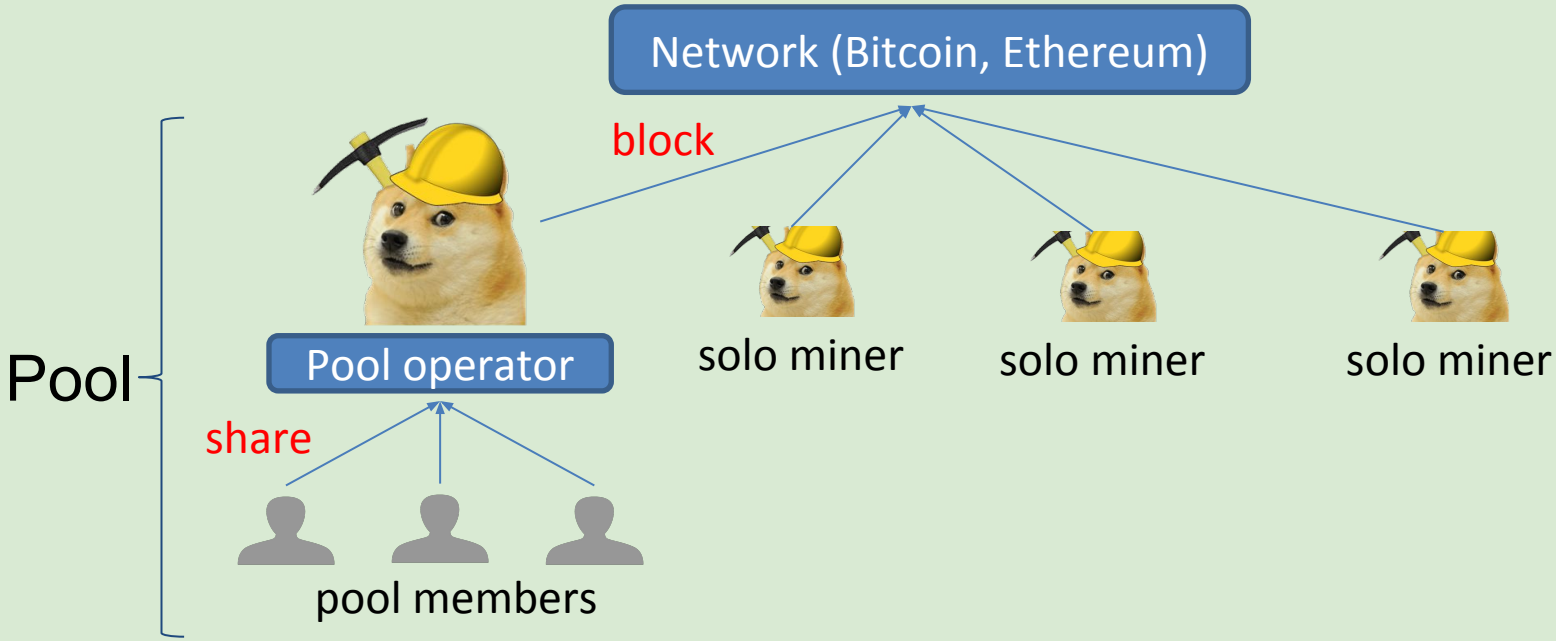
Measuring pool member contributions

A *valid share* solves a Block Problem with relaxed difficulty (*d* >> *D*).

Valid block	hash(block, nonce, data) ≤ D
Valid share	hash(block, nonce, data) ≤ d

- Every valid block is a valid share.
- *d/D* fraction of valid shares are valid blocks.
- Pool (roughly) pays *d/D* block reward per share.

Mining network overview

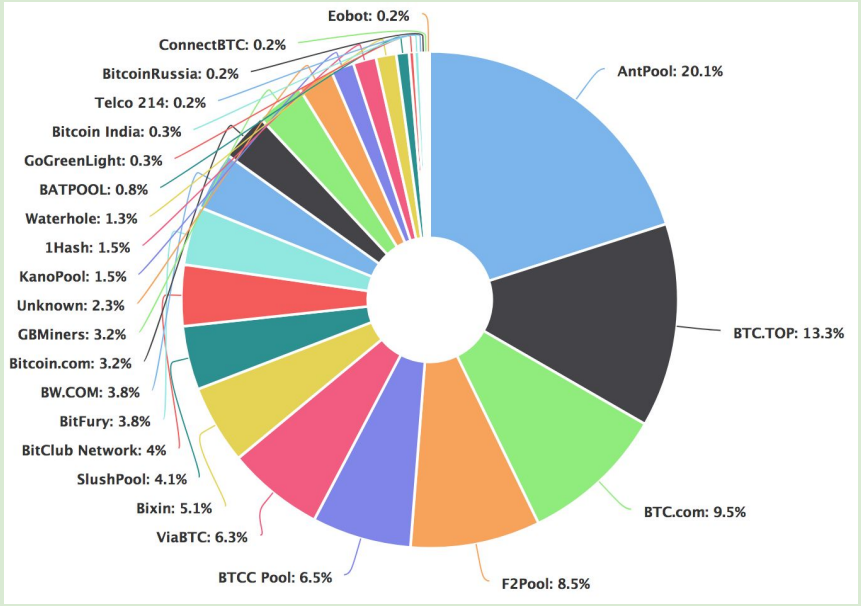


Pool members must submit shares such that block rewards are payable to pool operator.

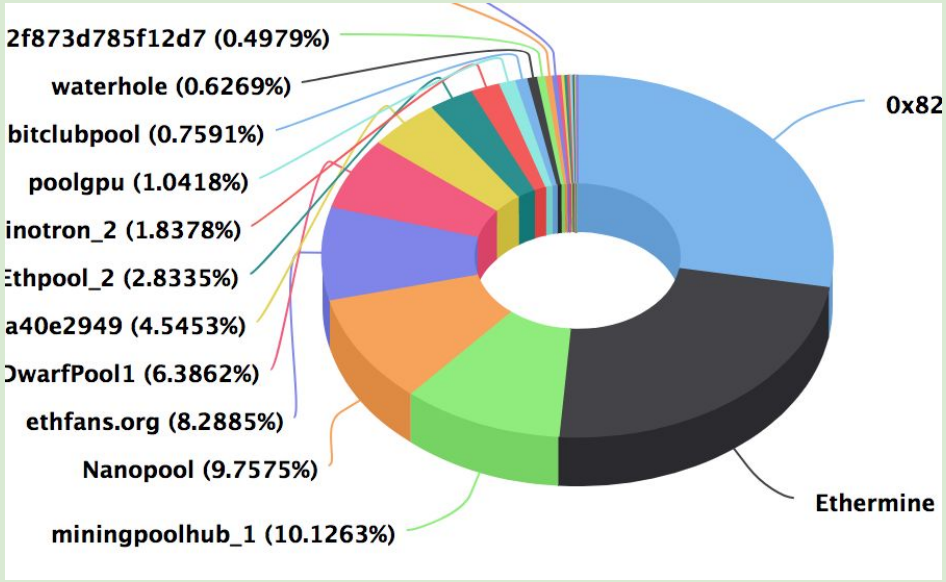
Centralization in Bitcoin and Ethereum

95% of Bitcoin's hash power lies in 10 pools.

80% of Ethereum's hash power lies in 6 pools.



Bitcoin's mining power distribution (https://blockchain.info , 28 Jul 2017)



Ethereum's mining power distribution (https://etherscan.io , 28 Jul 2017)

Censorship

Centralized mining pool operators can:

- influence which transactions enter the blockchain,
- control Ethereum's gas limit,
- inflate gas prices, and
- collude with each other (e.g. selfish mining, 51% attack).

THE MERKLE

F2Pool Allegedly Prevented Users From Investing in Status ICO

Other single points of failure

- Network partition vulnerability [Apostolaki, Zohar, Vanbever IEEE S&P ‘17]
- Geopolitical consolidation
- Pool members must trust their operator to pay fairly for shares
- Protocol upgrades (e.g. failure to activate Segwit)

Decentralized pool operators

Smart contracts

Ethereum is a **decentralized platform that runs smart contracts:** applications that run exactly as programmed without any possibility of downtime, censorship, fraud or third party interference.

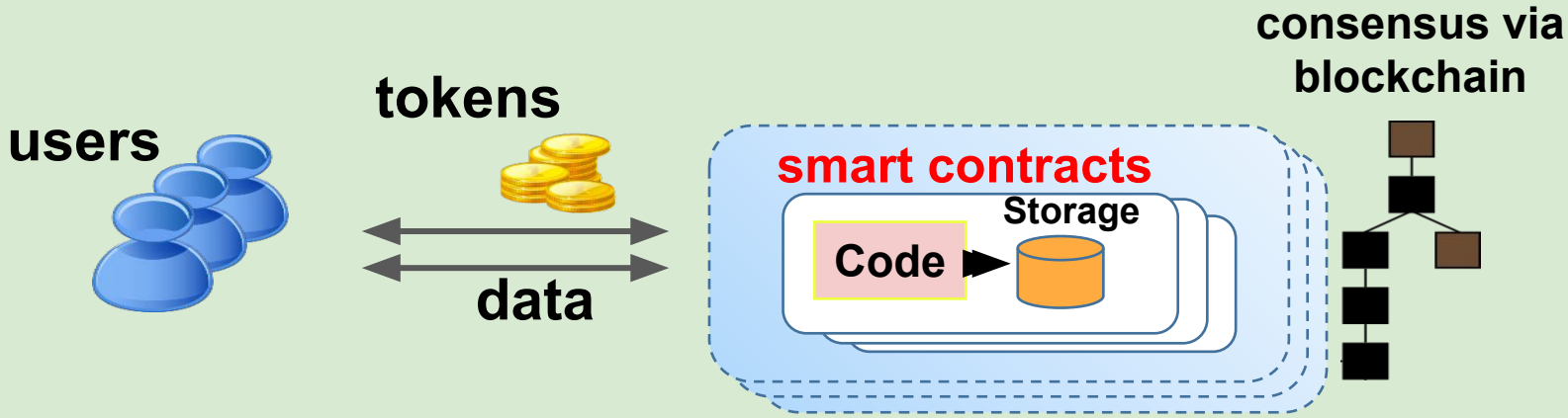
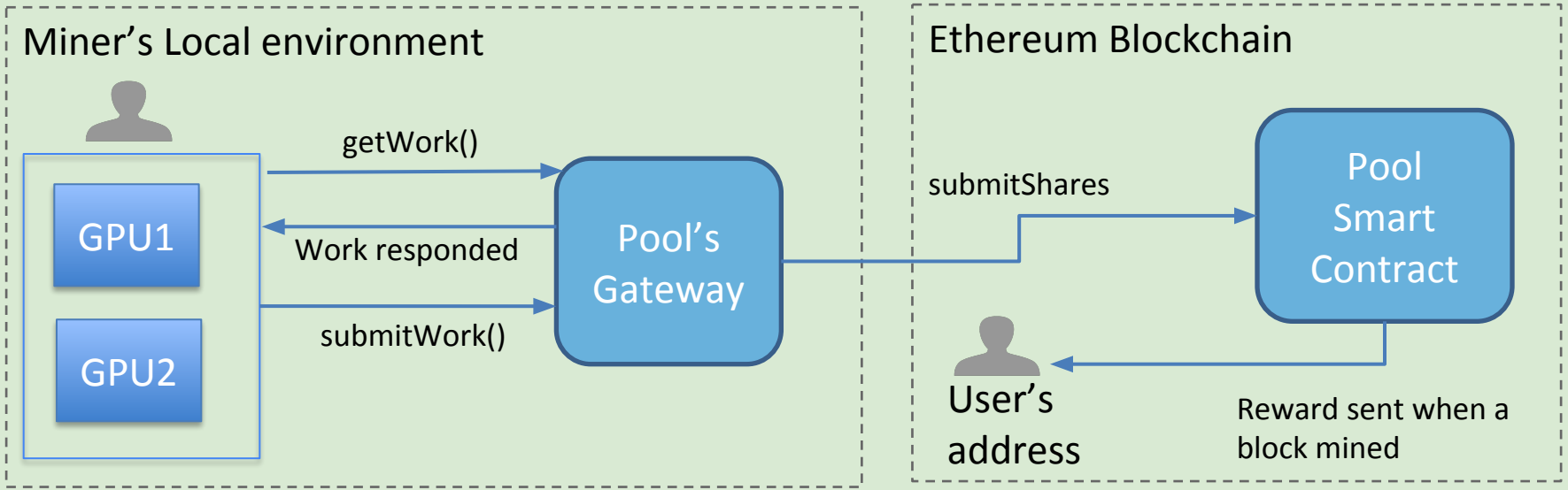


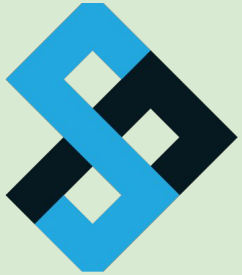
Diagram credit: Andrew Miller

Turning a smart contract into a pool operator



Miner uses the pool's address as the "coinbase" address in their blocks.

SmartPool properties



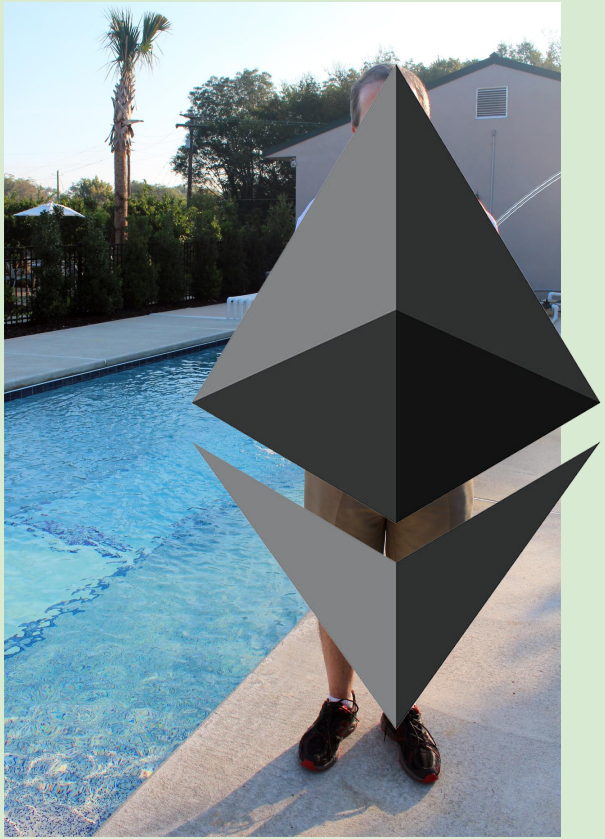
Unique to SmartPool:

- **Decentralized.** No central point of failure.
- **No censorship.** Individual miners choose shares.
- **Low cost.** Miners pay only Ethereum transaction fees.
- **Trustless, automatic payouts.** Smart contracts eliminate social contracts.

Preserved from traditional mining pools:

- **Low variance.** Miners choose share difficulty.
- **Fair.** All participants receive rewards in proportion to their contributions.
- **Open.** Anyone can join or leave at any time.
- **Anonymous.** Mining addresses are untraceable.
- **Retrofitting.** SmartPool works in Ethereum.
- **No security deposits.** Switch on a computer and start mining!

Protocol idea #1: substitution



What's wrong with this picture?

- Number of submitted shares is large.
- Ethereum network cannot handle high transaction volume.
- Cost to verify a share on-chain may exceed share reward.

Protocol idea #2: reduce the number of shares

P2Pool's "sharechain" takes the following approach:

> Adjust share difficulty so that shares occurs once every 30 seconds.

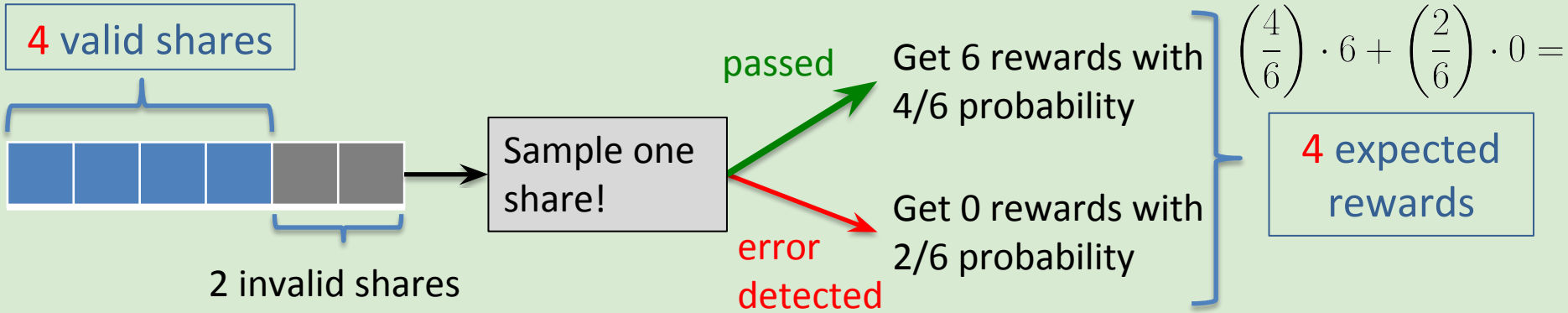
This scales poorly: more miners implies greater payout variance.

Other sharechain complications:

- High variance payouts (due to infrequent blocks)
- Many orphan shares (due to short block time)
- Poor security at small scale
- Need to check for redundant shares and incentivize block submission

Protocol idea #3: probabilistically sample shares

SmartPool batch submission:



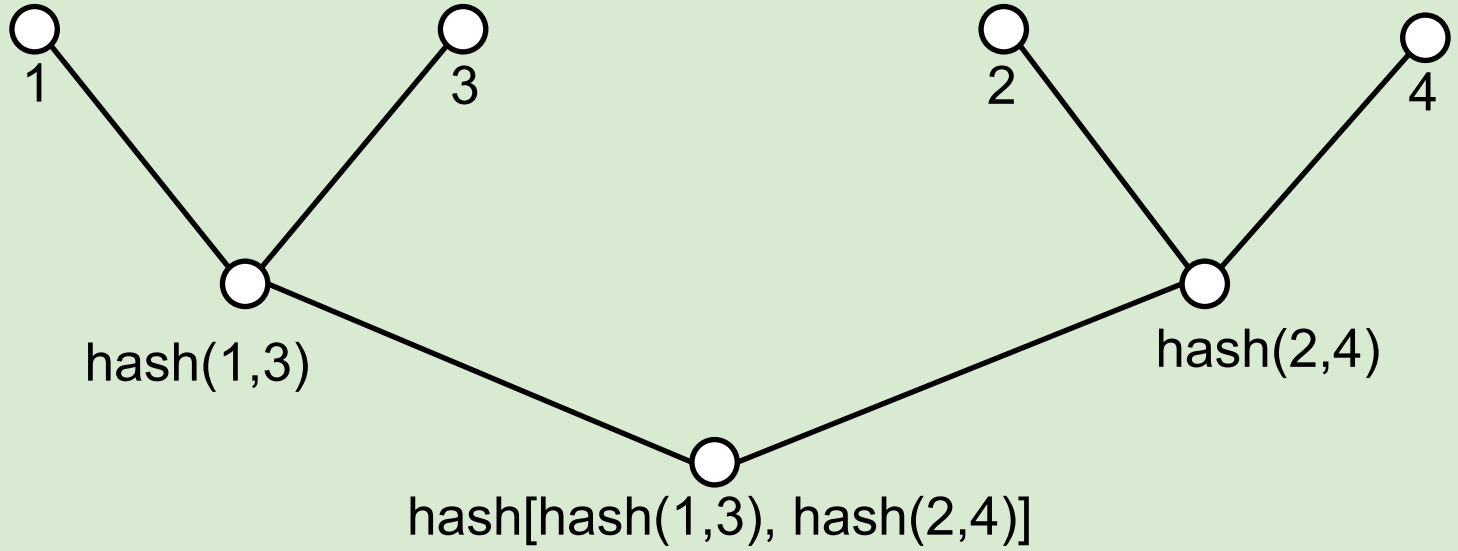
Adding invalid shares to submission does not change expected rewards.

Problem: What about duplicate shares?

Solution: Force miners to "sort" their shares using "augmented Merkle trees!"

Augmented Merkle trees

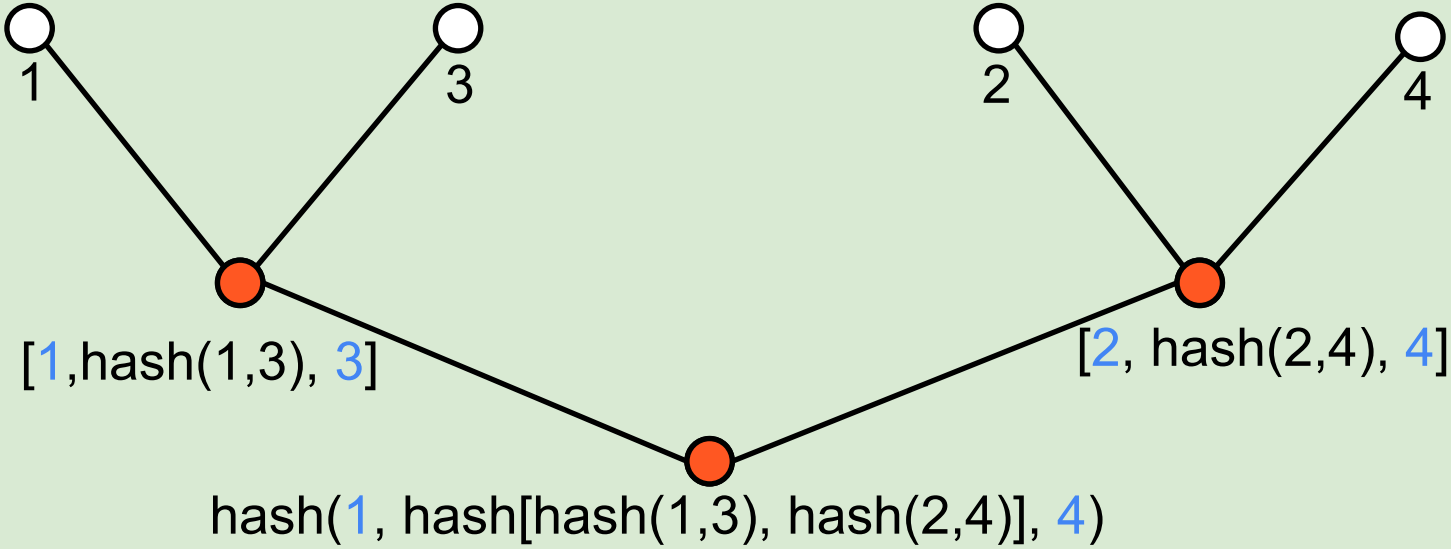
Merkle trees



Definition: In a *Merkle tree,* every child is the hash of its parents.

Data at the root forces commitment at the leaves.

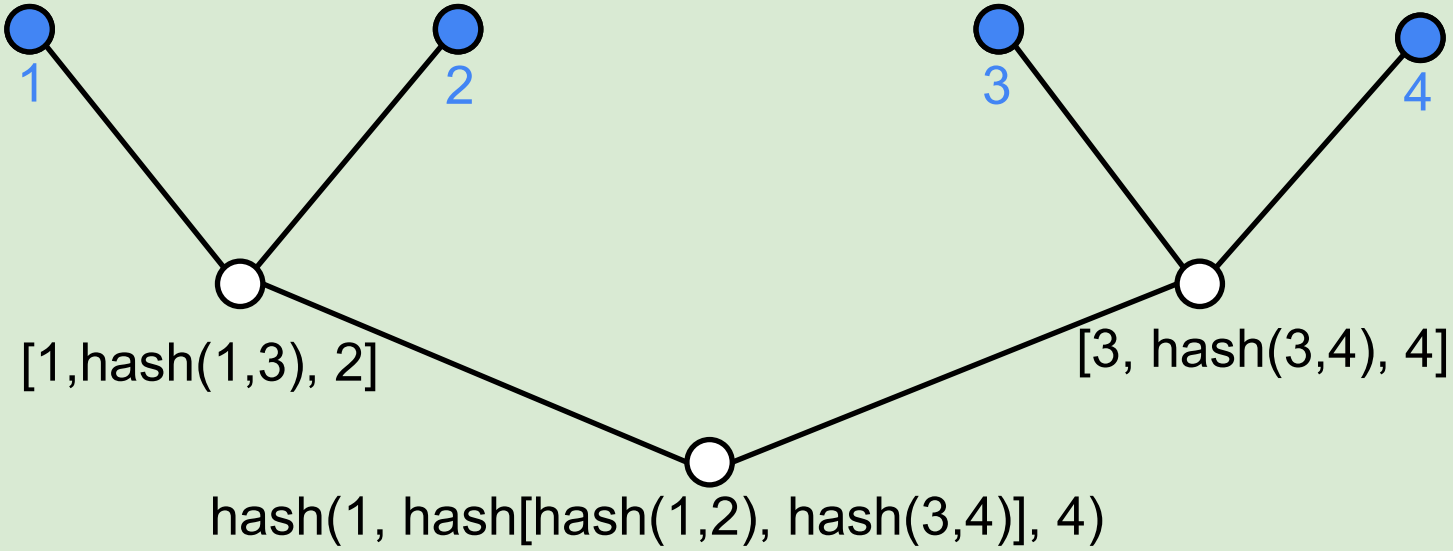
Augmented Merkle trees



Definition: An *augmented Merkle tree* contains additional **min** and **max** values at each non-leaf node where **min** is the minimum of its parent's **min** (and similarly for **max**).

The root witnesses a "sorting error."

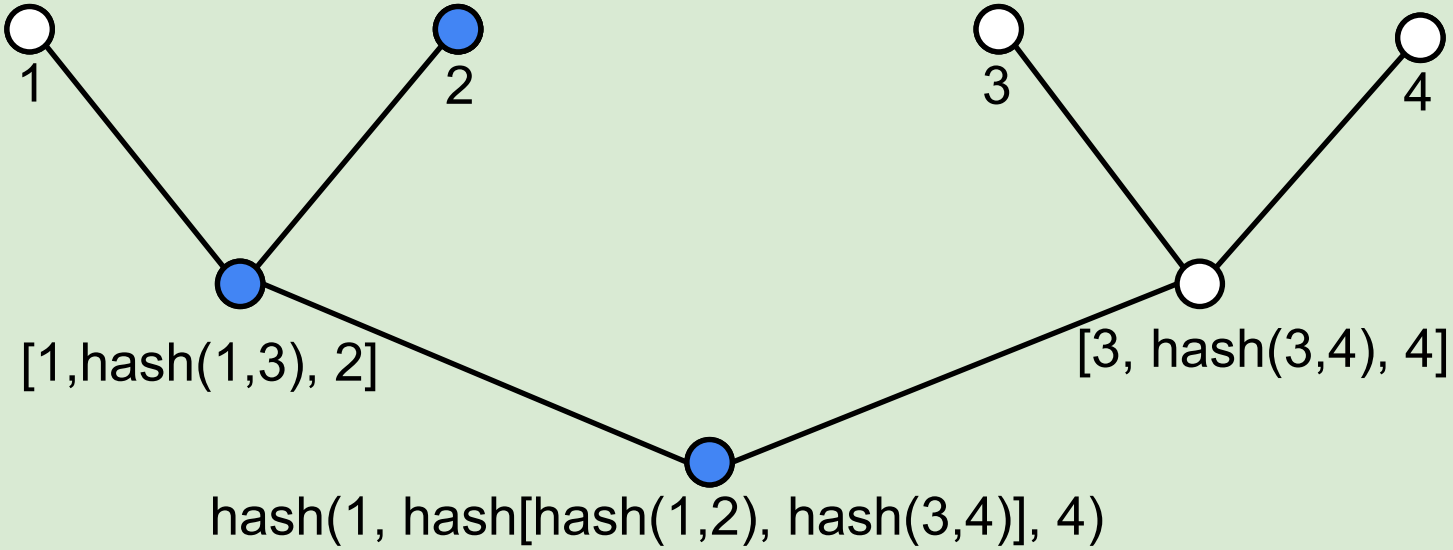
Sorted augmented Merkle trees



Definition: An augmented Merkle tree is *sorted* if its leaves occur in strictly ascending order from left to right.

In SmartPool, leaves are shares. **Share batches are ordered by timestamp.**

Submitting share batches in SmartPool



Protocol steps: The Miner and Smart contract interact as follows:

M → S: Root of sorted, augmented Merkle tree (batch submission).

S → M: Request for a random leaf (containing a share).

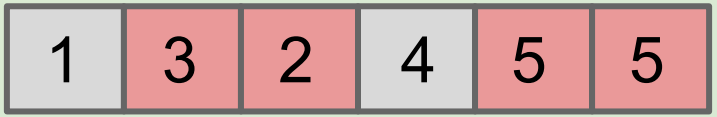
M → S: Path to leaf.

Security analysis

Sorting error example

Definition: An element *x* in an array is *out of order* if

- there exists a witness to the left which is greater than or equal to *x*, or
- there exists a witness to the right which is less than or equal to *x*.



Example: An array with 4 elements out of order.

Impact of sorting on deeper tree levels

Proposition: For any augmented Merkle tree A, the following are equivalent:

(i) A is sorted.

(ii) For every node $x \in A$, the **max** of x's left parent is less than the **min** of x's right parent.

Proof:

(i) ⇒ (ii): Induction on tree depth.

(ii) ⇒ (i): **min**(x) takes the minimum among all nodes above x.

Definition: A node which satisfies (ii) is called *valid*. A path from a root to a leaf is *valid* if all its constituents are valid. A path which is not *valid* is *invalid*.

Counting sorting errors

Theorem: Let A be an augmented Merkle tree.

(i) If A is sorted, then all paths in A are valid (see previous slide).

(ii) If A is not sorted, then every leaf which is out of order lies on an invalid path.

Corollary: Every augmented Merkle tree has at least as many invalid paths as leaves out of order. **In particular, there are at least as many invalid paths as there are duplicate values among the leaves.**

Sampling works :)

Attack strategies

An adversary who deviates from intended claim submission behavior does not obtain greater rewards.

- **Block withholding.** Sampling two shares per submission suffices to deter rational miners (with < 50% power) from dropping blocks.
- **Rearrangements.** Permuting leaves does not increase expected profits.
- **Bogus nodes.** Falsifying nodes in the augmented Merkle tree submission does not provide an advantage.
- **Repeating shares across submissions.** Later submissions must have later timestamps.

Implementation

Verifying the unverifiable: Ethash

Problem: Ethereum has a memory-hard proof-of-work puzzle.

- Each Ethash instance queries 64 random elements from a 1GB dataset.
- Smart contract storage costs ≃ 2,500,000 USD per GB (July 2017).
- 1 GB dataset changes every week.
- Infeasible to store dataset on-chain!

Solution: Store the Merkle root of each (precomputable) 1 GB dataset in SmartPool's smart contract!

- Miners can verify the roots before joining the pool.
- Miner include a "Merkle proof" of the 64 element set in submissions.

Operating in Bitcoin

Ethereum-based SmartPool protocol supports Bitcoin mining.

- Bitcoin “coinbase” transactions specify whom to pay block reward.
- SmartPool shares indicate coinbase payees.
- SmartPool maintains an ever-growing list of *recent pairs*:

  (accepted batch, payee Bitcoin address)
- Accepted SmartPool share must reference payees from recent pairs in its coinbase field.
- Recent pairs list updates periodically.
- SmartPool miners submit blocks directly to Bitcoin, and payees from recent pairs receive reward in Bitcoin.

SmartPool is LIVE!

- **ETH mined:** 86 blocks ≃ 2200 USD/day (June - July 2017)
- **SmartPool operating cost:** 0.6% fee (compared to 3% for F2Pool)
- **Number of miners:** 5 (not yet open to public)
- **Hashrate:** 30 GHs (200 mining rigs, 6 GPU each)
- **Ethereum gas cost per submission:** 2.6 million gas
  (gasLimit ≃ 6.7 million, Aug. 2017)

Project website:
http://smartpool.io

Interested in building Web 3.0?
TrueBit is hiring! jt@truebit.io