Making Sense of Plasma

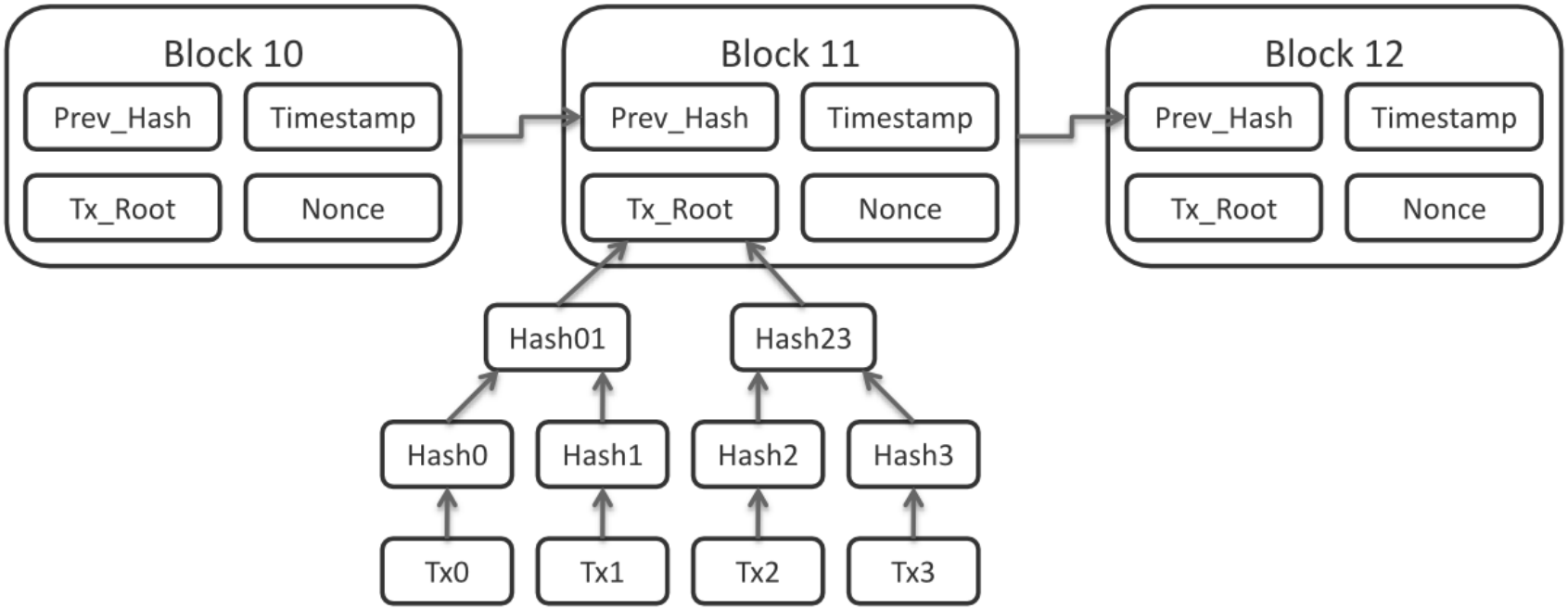
Thomas Sibut-Pinote

February 22, 2018

Bitcoin

- Decentralized currency;
- Described by its history of transactions, decomposed into blocks $\rightarrow$ **blockchain**;
- Validators: *miners* build the blockchain for a reward;
- Signatures to identify accounts;
- Cryptographic Hash Functions:
  1. Make it hard and slow to build new blocks;
  2. Compress data into Merkle Trees.

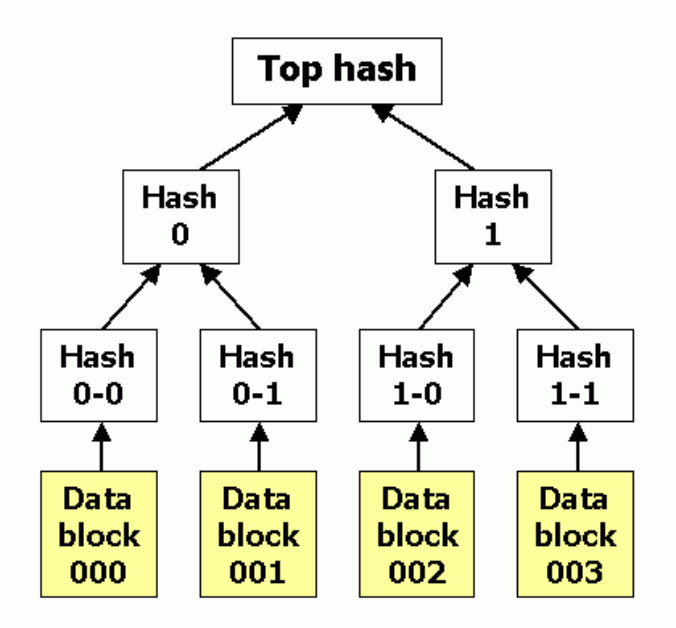
Bitcoin Block Structure



Why a chain:

- Orders blocks in time;
- Makes it harder to fork a new chain with a common history.

Merkle Trees



Key properties:

- A single hash commits to a whole set of data;
- Logarithmic size proof of inclusion of data.

Ethereum

- Just like Bitcoin, a decentralized cryptocurrency with miners;
- Accounts can be programs: *smart contracts*;
- Transactions can be calls to smart contracts;
- Every single (full) node checks every single transaction.

Problem

Approximate capacity in transactions per second:

- Bitcoin: 7 tx/s;
- Ethereum: 20 tx/s;
- Visa: $56,000$ tx/s;
- AWS: A lot.

Bitcoin blocks are full and transaction fees are off the charts.

How can they be scaled up?

Using Bigger Blocks

Build $k\times$ bigger blocks, get $k\times$ more transactions.

Advantages:

- Straightforward to implement.

Inconvenients:

- Validators need more processing power $\Rightarrow$ less decentralization;
- Limited parallelization (at least in current model).

Sharding

Cut transactions in small sets, split validators between sets.

Advantages:

- Doesn't break decentralization: each validator does small amount of computation.

Inconvenients:

- Data availability may be lowered;
- Less validators per transaction $\Rightarrow$ less security.

Using Several Independent Blockchains

When a blockchain is full, use another.

Advantages:

- There are already 1000's of blockchains.

Inconvenients:

- Bad security (especially PoW);
- Difficulty to interact between blockchains (atomic swaps).

Payment Channels

ON WHITEBOARD

Lightning Network

Idea: if there is a global graph where edges are payment channels, most transactions can happen off-chain.

Advantages:

- Payment channels are very fast;
- No trust is lost compared to on-chain transactions;
- Significantly smaller fees.

Inconvenients:

- Have to stay online to monitor bad behaviour;
- All parties involved have to sign new states: doesn't scale to multi-party transactions.

Specification

At this stage the Plasma authors want a way for several parties to conduct computations off-chain (for scalability and decentralization) in a way that can be disputed on-chain.

Specification

At this stage the Plasma authors want a way for several parties to conduct computations off-chain (for scalability and decentralization) in a way that can be disputed on-chain.

Can't store the full state of computations $\Rightarrow$ Merkle tree

Specification

At this stage the Plasma authors want a way for several parties to conduct computations off-chain (for scalability and decentralization) in a way that can be disputed on-chain.

Can't store the full state of computations $\Rightarrow$ Merkle tree

Can't store every single step: regular checkpoints.

Specification

At this stage the Plasma authors want a way for several parties to conduct computations off-chain (for scalability and decentralization) in a way that can be disputed on-chain.

Can't store the full state of computations $\Rightarrow$ Merkle tree

Can't store every single step: regular checkpoints.

So, regular checkpoints containing a Merkle root with transactions and the current state of computations.. We're talking about something very similar to the headers of a blockchain.

Simplest Version of the Idea

When no one is cheating:

- A dictator builds blocks, publishes them to users and sends headers to the parent chain;
- A smart contract on the parent chain receives block headers;
- Blocks are finalized after a challenge period;
- All child transactions are just money transfers;
- Users can deposit funds in the smart contract and retrieve them in the child chain;
- Users can withdraw funds from the child chain through the smart contract.

Some questions

- Does the child chain really need to be a blockchain?
- How are users protected from invalid transactions/invalid blocks/double spends?
- How can funds be securely deposited/withdrawn to/from the child chain?
- How does a transaction take place on the child chain?

Does the child chain really need to be a blockchain?

- Ordering of blocks can be provided by smart contract;
- Dictator: no need to make forks hard **by design**, not an open network
- However, what if there is a fork of the root chain?

Protection Against Invalidity In Bitcoin/Ethereum

- In Bitcoin/Ethereum, anyone can check transactions and blocks for invalidity or double spends because they are **public**;
- They **have to be public** for miners to be able to claim their rewards;
- If a transaction is invalid in a block, the block will simply be rejected by any node.

Protection Against Invalidity in Plasma

Dictator will not reject invalid blocks (he created them).

Solution: Smart contract has a method for denouncing an invalid block through a **fraud proof**: e.g. proof of inclusion of an invalid transaction.

Dictator has to put a bond which will be forfeited if someone publishes a fraud proof.

All this supposes that blocks are **public**.

How can Plasma ensure that child blocks are public?

It can't. You can't prove that you were **not** shown a block.

How can Plasma ensure that child blocks are public?

It can't. You can't prove that you were **not** shown a block.

What can the Dictator/Validator do by withholding blocks? For example:

- Make it uncertain whether a transaction was included in it;
- Add fraudulent transactions that steal money;
- Add fraudulent transactions that mint money.

How can Plasma ensure that child blocks are public?

It can't. You can't prove that you were **not** shown a block.

What can the Dictator/Validator do by withholding blocks? For example:

- Make it uncertain whether a transaction was included in it;
- Add fraudulent transactions that steal money;
- Add fraudulent transactions that mint money.

Proposed solution in Plasma:

- As long as blocks are published, monitor them for irregularities, submit fraud proofs;
- As soon as a block is withheld, exit from the child chain!

How can funds be securely deposited/withdrawn to/from the child chain?

Deposit:

1. Depositor sends x coins to the contract, with bond;
2. The Validator includes a *prepare(x coins, Depositor)* in the next block;
3. Depositor checks the block and signs a *Commit* to the Validator.

How can funds be securely deposited/withdrawn to/from the child chain?

Deposit:

1. Depositor sends x coins to the contract, with bond;
2. The Validator includes a *prepare(x coins, Depositor)* in the next block;
3. Depositor checks the block and signs a *Commit* to the Validator.

Remark: **Validity assumption**: On the root chain, no more goes out than comes into the smart contract.

Withdrawal

- Unconfirmed withdrawal: only if confirmation (step 3) of deposit hasn't happened (subject to fraud proof)
- Simple withdrawal: only allowed after confirmation (step 3) of deposit
  1. Signed withdrawal transaction submitted on root chain, with bond;
  2. Challenge delay: anyone can submit a fraud proof that the transaction was spent in child chain;
  3. Second delay: allow for earlier withdrawals to go through;
  4. If no fraud proof submitted, withdrawal can be done.

Other withdrawal types

- Fast withdrawal by atomic swap using HTLC; needs a party willing to obtain coins on child chain (Whiteboard if necessary)
- Mass withdrawal: many users withdraw their coins at the same time in a single root transaction. Complicated construct, requires coordination between users. **Necessary in case of block withholding.**

How does a transaction take place on the child chain?

Alice wants to send a coin to Bob on the Plasma chain.

1. Alice signs and broadcasts her transaction spending a previous output;
2. Validator includes transaction in block of child chain, sends header to root chain;
3. Bob signs an acknowledgement that he has seen Alice's transaction included in a valid block. This acknowledgement may be included in a later block after being signed by Alice.

The only reason for step 3 is to allow for Bob to withdraw his money faster, otherwise he has to wait until the block is finalized. Before then Alice can still withdraw.

Chain trees

Once the parent-child link has been established, rinse and repeat $\Rightarrow$ chain trees.

Questions:

- When does one accept a transaction as finalized if it happened in a child chain? A grand-child chain? An n-child chain?

Other ideas

- Instead of a Dictator Validator, a set of validators and a consensus algorithm. They suggest Proof-of-stake + tokenization as an incentive to publish blocks but not very convincing. (p.35)
- MapReduce with fraud proofs (p.39);
- Inclusion of SNARKS/STARKS instead of just block headers.