Current Blockchain Themes: Thoughts on Data Availability

Joseph Poon <joseph@lightning.network>
Stanford CBR 2018-07-06

General Thoughts, the Timeliness of CBR

There are not enough people thinking in this space. It's still small, but incredible demand exists to explore the problem space. Space is in its infancy; many problems to be solved. Caused by high learning curve?

Deeply Interdisciplinary. A high need for a venue for collaboration.

The blockchain space currently has a significant issue around collaboration. Projects don't talk to each other to the point of Galapagos Island effect (different names for the same thing)

You have to do it live. Cryptoeconomics involves human incentives. Models work up to a point, but can be like playing poker with fake money. Live experimentation may be necessary. The blockchain is a place where research is actively deployed.

Summary

Everyone has their own pet problems in the blockchain space, this talk is about my view and one of the problems I'm working on. One of those is around Information Availability. This talk is colored by my views, there may be different perspectives.

Proofs (computable assurance) is predicated upon data availability. Withholding is a principal attack vector.

This is a common theme in cryptoeconomic solutions, see as an example the incentives Proof-of-Work mining.

Lightning punts the problem to a two party channel. Plasma Cash is an iteration on that and depends upon a multiparty live on-chain game.

Blockchain, Private and Public Data

Cryptography normally deals with secrets.

Encryption. Private keys for decryption.

Digital Signatures. Only private key owner(s) can generate proofs. We of course use this for authorization for spending of coins, which thereby is ownership.

One of the principal issue around trust, proofs, and cryptosystems on the blockchain is around availability of information. The availability of the message itself being verified. Historically cryptography is primarily about ensuring secrets remain secret and usable, instead the blockchain also is dependent upon ensuring/incentivizing that ***information released in the future*** is available and public.

Example: Data Availability Game in Proof-of-Work

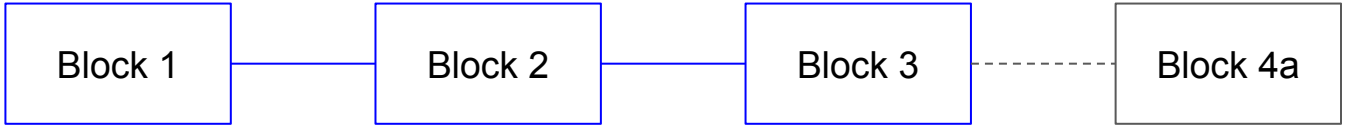
Block Publication Delays/Withholding. This is essentially delaying data availability to maximize one's return. PoW doesn't fully solve this problem, but it does mitigate a lot of the profit of doing so assuming that mining is sufficiently decentralized (the last part remains to be seen).

The Nakamoto consensus mechanism (PoW mining) is functionally a strategy for rapid propagation of information. Due to the way mining works, one is individually incentivized to propagate blocks as quickly as possible via a game whereby **one's rewards are higher by maximizing block information availability to peers**.

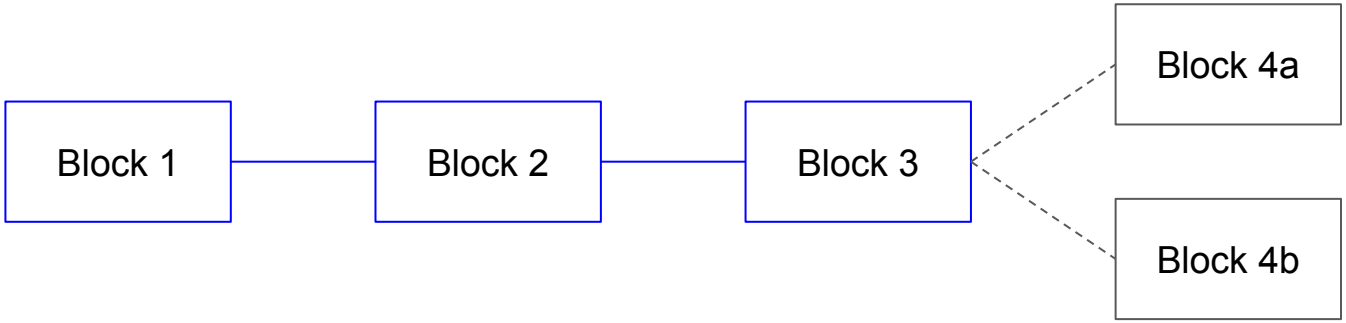
One does not know ex-ante when one has found a block whether it will be a winning block which is built upon. This is the key to the mechanism, winning blocks are probabilistic and competitive at the same time.

Presume a normal PoW chain.
(This example may be obvious, but could be helpful in designing other systems)

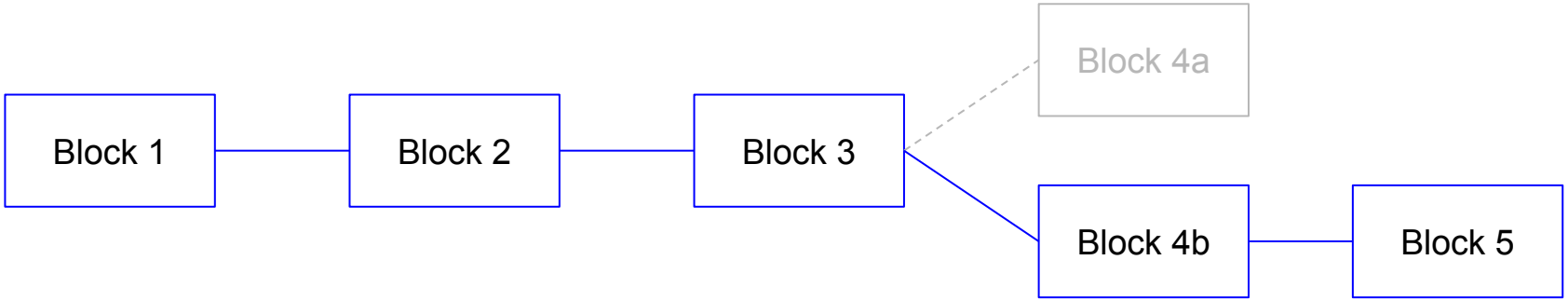
If you find block #4, you don’t yet know if your block will be built upon



If someone else finds a block at the same time, they could also be built upon



If you didn’t propagate 4a wide enough, then there is incentive to find 4b and build upon that



Example: Data Availability Game in Proof-of-Work

PoW works with data availability incentives due to uncertainty. Just because you found a block doesn't mean that you were the block producer. That relies upon everyone having the data available.

This is very different than traditional leader-based distributed systems. For example, many traditional round-robin distributed consensus mechanisms know who the next leader is. PoW is probabilistic leader election, to the point where one doesn't even know if one is the leader **after** a winning block is found.

Because one doesn't know if one is a leader (block found) after finding a winning blockhash, this creates incentive in one's action *after* block creation towards data availability of the entire system. Lack of certainty is used to inject incentives.

An open problem for novel cryptography

Can you ensure data availability across disparate participants using novel cryptography? There may be novel ways to use cryptography to have consensus mechanisms. E.g. novel use of BLS ensures lack of knowledge for block selection or leader assignment such as Dfinity's approach.

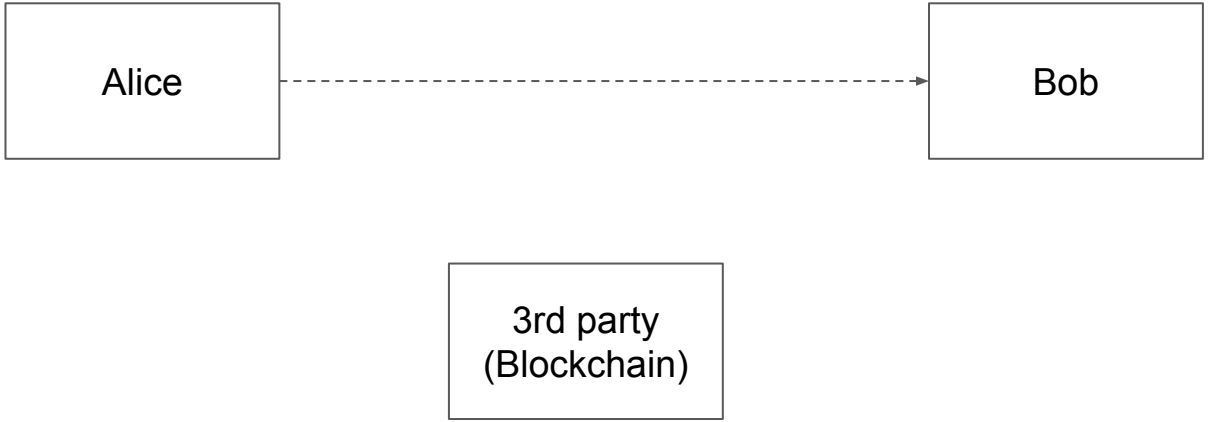
zk-SNARKS/STARKS may not be a panacea. A meme in this space, "But what if you used SNARKS?" While you may be able to have assurance around computation, you need data availability to generate and verify proofs. Reliance on proofs without data availability requirements may actually increase risks of failure.

Time and Randomness a recurring theme. Things like verifiable randomness functions or games with group signing as a way to have delayed release of information with deterministic results (but unknown ex-ante).

Blame is hard!

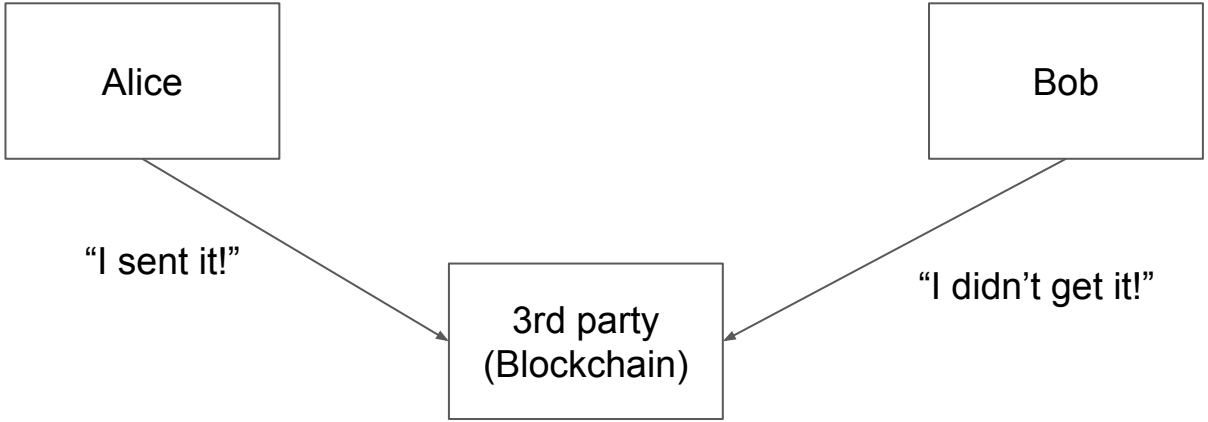
It's not possible to know without an interactive game who withheld information. The most you can do is do a challenge-response game, but that can be griefed by constantly challenging or constantly withholding (and then who bears the cost?)

Presume Alice will be required to send information to Bob *in the future*. There is no way to prove to a 3rd party that Alice has done so.



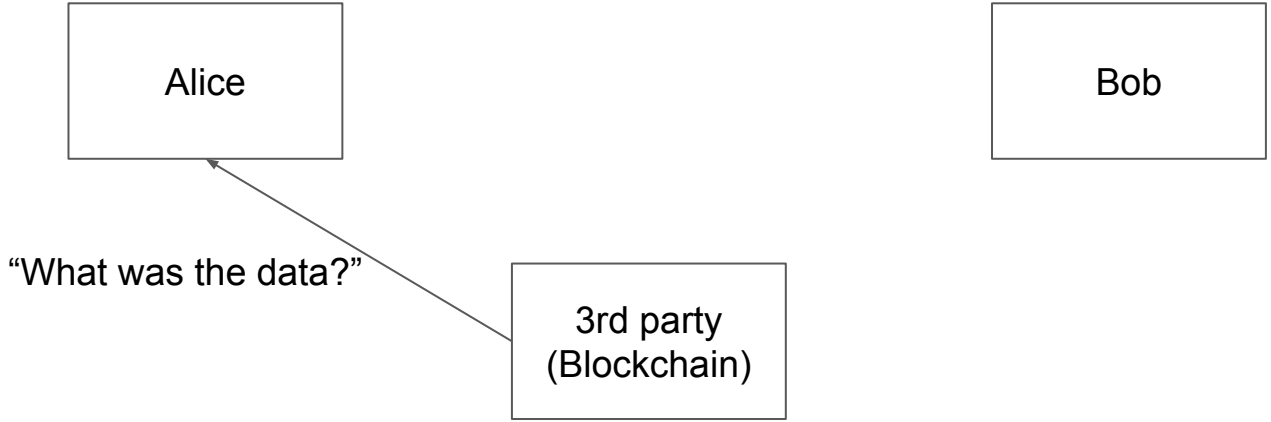
Blame is hard!

Alice tells the Blockchain she has sent information to Bob but actually has not. Bob disagrees. It is her word against his. Creating a multi-step process is possible, but committing to it ex-ante is not possible.

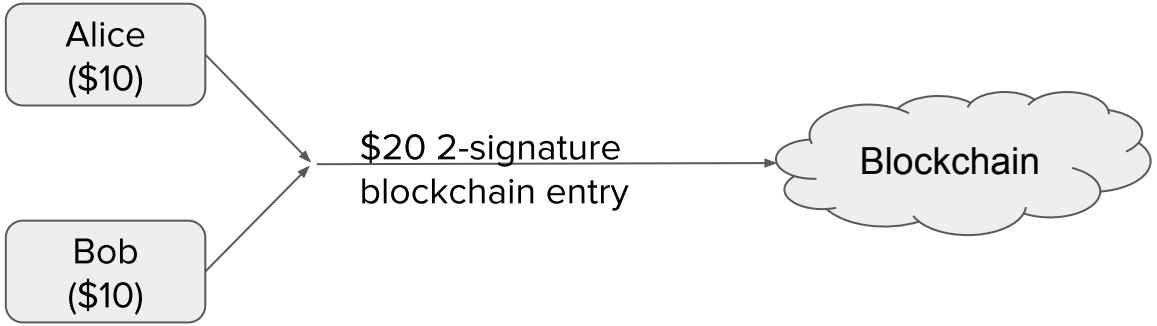


Blame is hard!

The most that the Blockchain can do is to ask Alice for the data later and if she doesn't publish it then she gets penalized. But we don't know if Bob is lying and Alice actually sent it. This creates significant overhead and is basically Alice obligating to release data to the Blockchain instead of sending data to Bob.

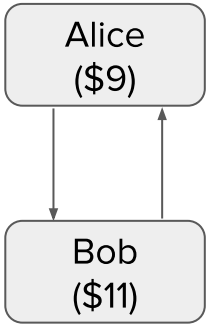


Example: Lightning Network Data Availability



The blockchain only sees a $20 2-signature ledger entry

Two participants assign funds on the blockchain into an entry which requires both parties to sign off to spend from the blockchain entry.



Alice and Bob exchange digital signatures directly with each other every time they want to update their balance in this local off-blockchain “channel.”

They can close out and settle the transaction at any time on the blockchain claiming their off-chain balance (since they’ve signed off but not broadcasted the old balances). They have also exchanged directly with each other a cryptographic bonded proof revoking the old balances, so only the newest one can be used.

Example: Lightning Network Data Availability

Lightning is a design which focuses on local data availability first and foremost. It only works because there is localized availability guarantees. Global state does not need the entire record because in the event of dispute both parties have sufficient information to attest to the current state.

Local vs. Global Data. Create incentives for local actions to be honest, but all actors must have sufficient data to make proofs globally.

As a consequence, liveness is required for state updates to self-attest to data availability. Liveness with local state updates and some level of liveness for interactive proofs/disputes globally.

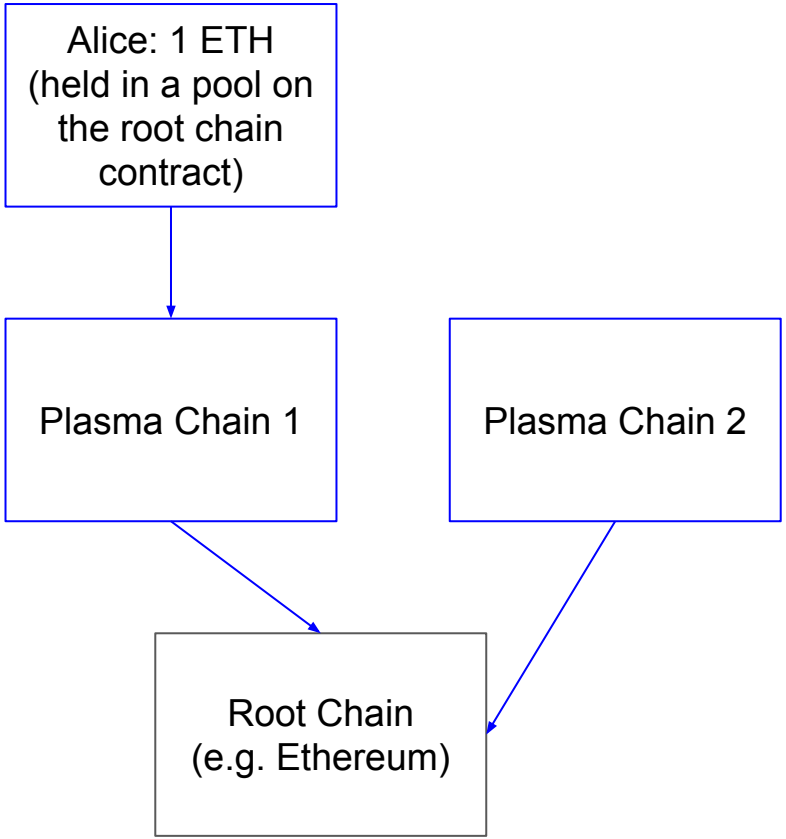
Plasma: Data Availability

Merkelize committed data to a root blockchain to have ordering with small data footprint, “blockchains on blockchains”

Allows for deposits/withdrawals from root chain of native coin/token(s).

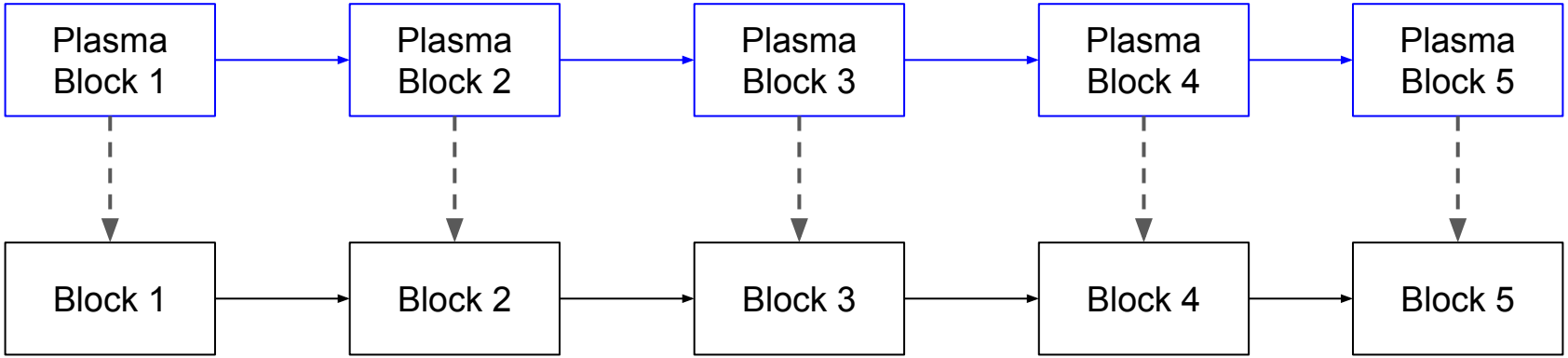
Achieved via a cryptoeconomic game during the withdrawal process

Allows for membership set addition without significant on-chain data.



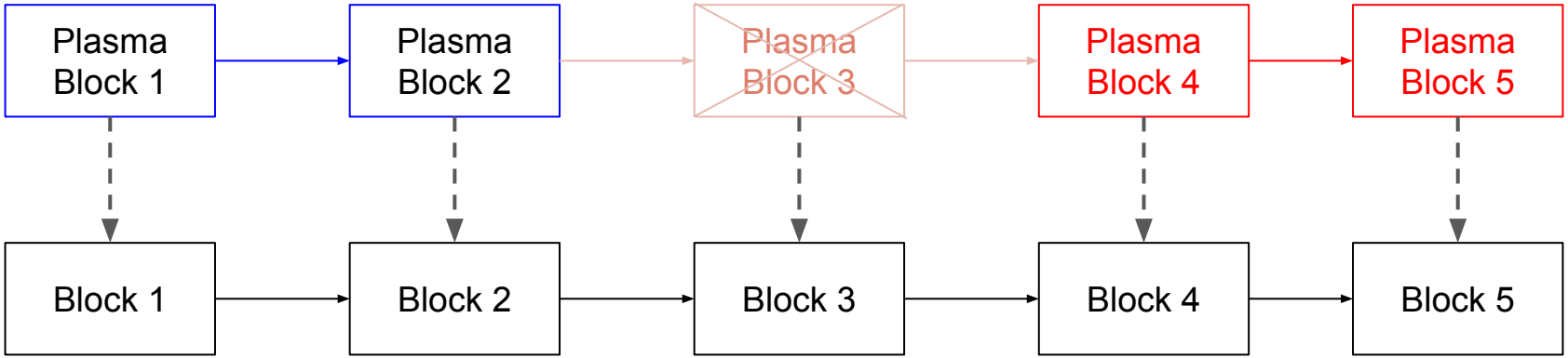
Plasma Data Availability Problem Example

Only blockhash/header committed to the root chain (dashed lines). This means that a minimum amount of data is submitted to the chain (blockhash, sig(s)). This submission is a commitment to blockstate as well as creates ordering



Plasma

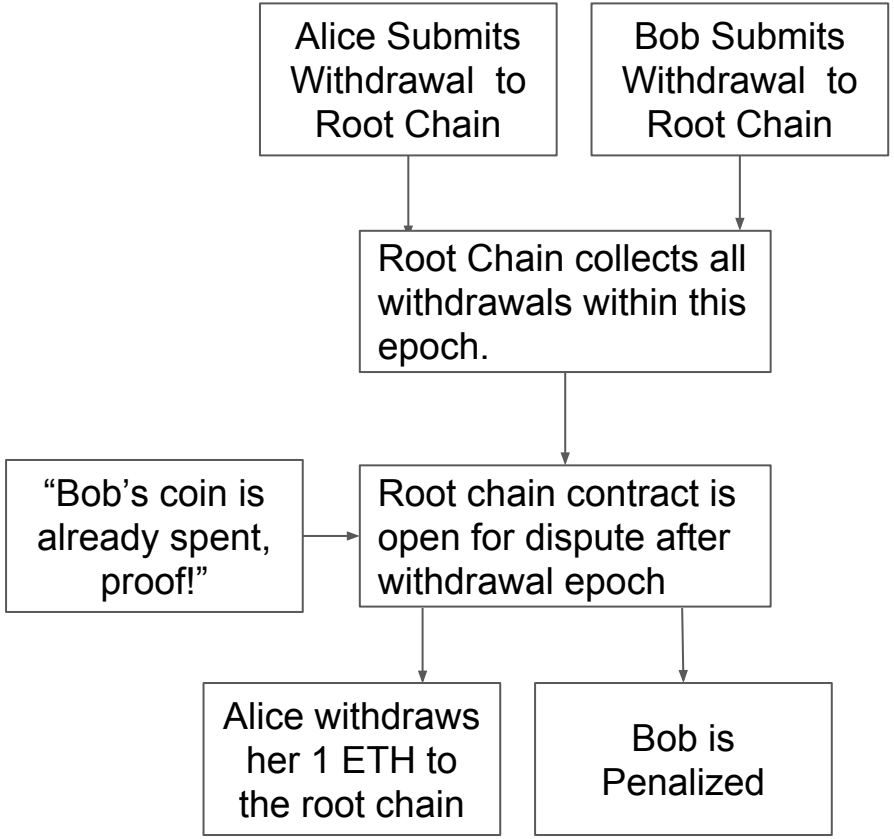
However, what if Plasma Block 3 is unavailable? A correct merkelized commitment provides the correct data to the root blockchain ("Block 3"), but the transactions therein may violate consensus (e.g. a double spend). The contract with the pool of funds would be bankrupt if a withdrawal submits a proof.



Plasma MVP Withdrawal Mechanism

1. A bonded withdrawal request is submitted to the root chain contract, a merkelized proof of coins is provided from a recent Plasma Block.
2. Root chain contract has an open specified time window to submit other withdrawals for older coins.
3. A second time window exists to publish disputes which anyone can submit, spentness proofs.
4. Withdrawals are ordered by coin age.

If a block is withheld, everyone begins to withdraw coins. The ordering by coin age lets coins spent before the withheld period to be prioritized (one should not make spends if blocks are withheld).



“Plasma Cash is Plasma”

Problem with Plasma MVP. Strong assumptions on liveness and blockspace availability in the root chain. Resolved with the “Plasma Cash” construction.

Update to Plasma. Many contributors, Dan Robinson, Vitalik Buterin, David Knott, Kelvin Fichter, Karl Floresch and many others!

Localized data availability to a TXO deposit entry. A UTXO represents a specific balance and can be spent without publishing entire states on the root chain.

Coins are non-fungible. They can be split/merged within a deposit, but deposits cannot be intermixed without cooperation of all parties.

TXO History Management

Merkelized commitment of transactions with compact inclusion/exclusion proofs. Patricia Trie or Sparse Merkle Tree provides compact proof of whether a block contains a spend of the deposit.

Client Side Validation! The client must maintain a copy of all prior blocks' merkelized commitment proof of inclusion/non-inclusion of the current TXO deposit.

In the event of block withholding, one should not make further transactions. This means that one will always have proof of the history of the coin.

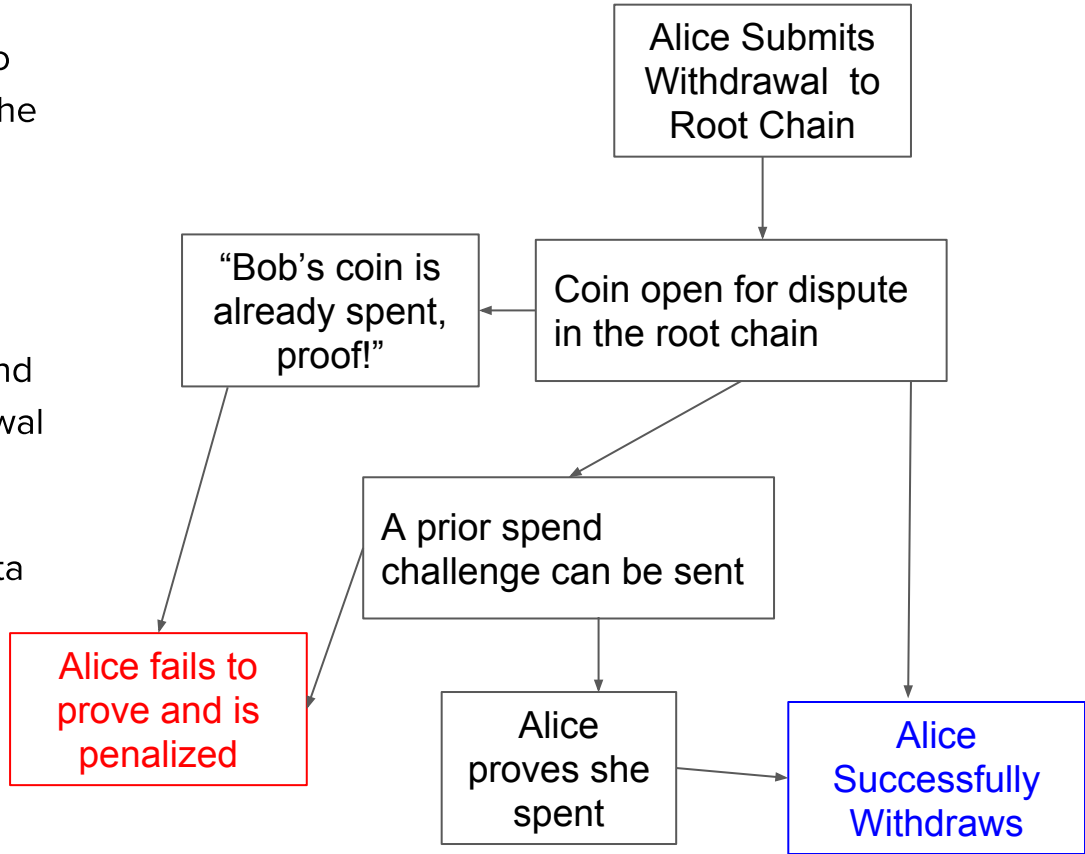
This construction binds data availability to a specific set of coins. No intermixing of coin states means all parties affected have data availability, and attestation of disputed data can be made via an interactive game.

“Plasma Cash” withdrawal mechanism

1. Bonded withdrawal of coins submitted to the root chain with merkelized proof of the most recent spend
2. Dispute period which lets you submit a proof of a spend of that coin
3. If this is from a withheld block, a dispute can be sent which requires proof of spend
4. After dispute period completes, withdrawal possible.

This takes into account future possibility of data unavailability presuming one is periodically watching the root chain, and the root chain is available and non-censoring.

Data dependencies are localized.



Summary

Various strategies on data availability. Games with uncertainty whereby propagation increases profitability, localized vs. global data availability, and pushing responsibility to the edges.

Potential for cryptography to increase incentives. E.g. validator commitments to error-correction codes, VDFs, BLS and other aggregate signatures, etc. **Compact externally usable proofs is very helpful and will become more important.**

Interested in helping CBR's research. What problems are you thinking about?

Further Thoughts. The Blue Eyes Problem is a good primer for understanding why the blockchain functions, possibly the best way to develop an intuitive feel for the mechanisms. Something something generals' problem.

The Blue Eyes Problem (xkcd)

A group of people with assorted eye colors live on an island. They are all perfect logicians -- if a conclusion can be logically deduced, they will do it instantly [and they also know everyone on the island are perfect logicians]. No one knows the color of their eyes. Every night at midnight, a ferry stops at the island. Any islanders who have figured out the color of their own eyes then leave the island, and the rest stay. Everyone can see everyone else at all times and keeps a count of the number of people they see with each eye color (excluding themselves), but they cannot otherwise communicate. Everyone on the island knows all the rules in this paragraph.

The Blue Eyes Problem (xkcd)

On this island there are 100 blue-eyed people, 100 brown-eyed people, and the Guru (she happens to have green eyes).

The Guru is allowed to speak once (let's say at noon), on one day in all their endless years on the island. Standing before the islanders, she says the following: "I can see someone who has blue eyes."

Who leaves the island, and on what night?