Strong Federations - Summary and Review

Srikar Varadaraj

November 15, 2017

1 Problem Motivation

- Bitcoin's PoW was designed to process transactions every 10 minutes.
- Even after transactions are added to the blockchain, merchants typically wait until more blocks (4-5) are added because there is a constant risk of rollbacks/reorganization.
- Other payment networks leave final settlements in limbo for up to 120 days, so Bitcoin is still better in that respect.
- Regarding privacy, addresses can be traced on the public Blockchain and anonymity can be compromised.
- Overcoming slow Bitcoin transaction speed + threat of rollback (5 block confirmation time) is a central problem.

2 Previously Proposed Solutions

- Solutions to the above problem have often resorted to centralization, which creates a SPOF(Single Point Of Failure) risk.
- Recent Ripple attack demonstrates the SPOF risk. Ripple and Stellar have small blocktimes ($<$ 1 min, often on the order of seconds), but face SPOF risk.

2.1 Paper's contribution

The authors claim to provide a system that satisfies the following criteria:

- Public Verifiability
- Liquidity
- No SPOF
- Multiple Asset-Type Transfers on the same Blockchain
- Privacy

The authors also claim that their implementation "**Liquid**" satisfies the above properties.

2.2 Strong Federations and Sidechains

Q. What are Sidechains?
A. Blockchains that allow users to transfer assets to and from other blockchains.
Sidechains work by *locking the assets in a transaction on one chain, making them unusable there, and then creating a transaction on the sidechain that describes the locked asset.*
A transaction from the main chain to a side chain operates as follows:

- Send asset on main chain to special "freeze" address.
- Use "in" channel of federated peg to request a sidechain transfer.

- Equivalent assets are unlocked or created on sidechain
- Use "out" channel of federated peg to request a transfer from sidechain to main-chain.
- Once the **Strong Federation** reaches consensus on the validity of the transaction, the asset on the main chain is "unfrozen".

The above description immediately raises the following problems:

- If new "assets" are created on the sidechain, what is this process of "creation"? These must be destroyed at the end, but how does this relate to block difficulty, etc.

The authors refer to Bitcoin's PoW system as a type of DMMS(Dynamic Membership Multiparty Signature) since the membership in the verification process is dynamic, with signers choosing to host a node or not, according to their will.
A federated model has a **fixed signer set**. The DMMS is replaced by a standard **multisig**.This process increases the speed and scalability of the system. **However, we compromise decentralization and true trustlessness.**
More in-depth explanation of Strong-Federations:

- Members of Federation serve as "protocol adapters" between main chain and sidechain.
- Knowledge of private-keys enough to spend (as in all blockchains).
- Two types of **functionaries** in a federation: **Blocksigners** and Watchmen.
- Blocksigners sign blocks, ensuring consensus on new blocks, and advance the side-chain ledger.
- Watchmen are online when assets are transferred between different chains. They sign the blocks of the pegs.
- **Federated Blocksigning** is simply the method for achieving consensus in a Strong Federation. Signers signal intent to sign a block and if a threshold is met, they go ahead and sign the block in a typical multisig fashion.

The authors claim that when evidence of "tampering" is found, the system shuts down immediately. However, it is unclear at this point in the paper whether a smaller than majority of colluding functionaries can change the behavior of the system. The "incentive structure" was also not described in this structure, although it is claimed that incentives are aligned for the blockchain's security and the functionaries' benefits. However, it is true that the Strong Federation's block generation is deterministic, as opposed to PoW's probabilistic model.

- Blocksigning in a SF is robust against up to $2k - n - 1$ attackers. Note: We need k of n to sign a block.

Problems that String Federations claims to solve and their current "solution:

- Local currency trade in btc has illiquidity.
- Liquid uses btc-exchanges as functionaries.
- Smaller latency in Liquid -> increased transaction speed, reduces risk of price change during transaction (interesting point).
- Liquid supposedly has privacy guarantees (confidential transactions), but this is not as relevant to the rest of the paper.

2.3 Quick summary of the supposed benefits of Liquid

- Reorganizations cannot happen
- Faster and do not have to wait for multiple block confirmations.
- Privacy/Confidentiality. Claims that the transaction graph no longer exposes trace of money.
- Liquid creates a specific "hardened device" for key storage and signing of the functionaries. These claims do not seem very robust, somewhat silly, and mostly irrelevant.
- "Native Assets", or the ability to transact multiple assets in a single transaction. This was only described at a high level and again, does not seem to be very important.

2.4 Peg-Out Authorization

It seems that the design for moving assets from the bitcoin blockchain to the sidechain and back is the main point of the paper. However, the paper complicates it by also requiring "privacy" in these transactions. The authors refer to this mechanism as "Peg-Out Authorization". The design is as follows:

- **Setup**: Every participant has two pairs of public and private keys: (P_i, p_i) and (Q_i, q_i). The public keys P_i and Q_i are given to the watchmen.
- **Authorization**: A participant needs to authorize a key W corresponding to a Bitcoin address on the main chain. Hashes of a combination of P_j, Q_j and W are calculated. The resulting ring signature is sent to watchmen.
- **Transfer**: Then the watchmen ensure transaction outputs have an authorization proof.

This peg-out authorization mechanism is only made strangely complicated like this to ensure privacy. This isn't exactly the main problem and has nothing to do with scaling, which is the more central question.

2.5 Evaluation

In the "Evaluation" subsection of the paper, the authors claim that the participants in a Strong Federation will be naturally incentivized to take greater care of access to the federated signers under their control.
None of this is proved rigorously and is potentially not even true.

2.6 Other Proposed Solutions

- GHOST, block DAGs, Jute -> attempt to preserve decentralization, improve throughput.
- Ripple, Stellar, TenderMint -> improve throughput significantly, but more centralized (trusted signers).
- Lightning Network
- Eyal et al. came up with a Bitcoin-NG scheme

These solutions should be looked into, in particular GHOST, block DAGs, Jute, LN and the new scheme. They might be more decentralized, similar to what we are trying to achieve. The remaining part of the paper deals with hardware failure possibilities and mild suggestions for improving confidentiality, etc (nothing to do with the protocol).
The following three problems are of some relevance and are referred to at the highest level, though not explicitly dealt with theoretically:

- Rewriting History - Possible if majority of blocksigners collude. Easily detectable and bad signers can be replaced.
- Transaction Censorship - Also possible to detect since we can analyze the patterns of all blocksigners.
- Confiscation of locked Bitcoins, i.e. preventing a transaction from sidechain to main Bitcoin chain. Can be done if enough watchmen collude and refuse transfers.

None of the above were dealt with in detail and there doesn't seem to be a strong theoretical basis for many of the claims in this paper. Further analysis is required.