Name : Agung Panjimasjaya

Casper the Friendly Finality Gadget

Summary:

Casper provides safety, but liveness depends on the chosen proposal mechanism. That is, if attackers wholly control the proposal mechanism, Casper protects against finalizing two conflicting checkpoints, but the attackers could prevent Casper from finalizing any future checkpoints.

Casper introduces several new features that BFT algorithms do not necessarily support:

- **Accountability**. If a validator violates a rule, we can detect the violation and know which validator violated the rule. Accountability allows us to penalize malfeasant validators, solving the “nothing at stake” problem that plagues chain-based PoS.
- **Dynamic validators**. We introduce a safe way for the validator set to change over time
- **Defenses**. We introduce defenses against long range revision attacks as well as attacks where more than $\frac{1}{3}$ of validators drop offline, at the cost of a very weak tradeoff synchronicity assumption
- **Modular overlay**. Casper’s design as an overlay makes it easier to implement as an upgrade to an existing proof of work chain.

There are two well-known attacks against proof-of-stake systems: long range revisions and catastrophic crashes.

Casper remains imperfect. For example, a wholly compromised block proposal mechanism will prevent Casper from finalizing new blocks. Casper is a PoS-based strict security improvement to almost any PoW chain. The problems that Casper does not wholly solve, particularly related to 51% attacks, can still be corrected using user-activated soft forks. Future developments will undoubtedly improve Casper’s security and reduce the need for user-activated soft forks.