Over the past few years there has been considerable research into “proof of stake” based blockchain consensus algorithms. This is a vastly more efficient alternative to proof of work “mining” and enables blockchains to operate without mining’s high hardware and electricity costs. The first, chain-based proof of stake , mimics proof of work mechanics and features a chain of blocks and simulates mining by pseudorandomly assigning the right to create new blocks to stakeholders. The other school, Byzantine fault tolerant based proof of stake, is based on a thirty-year-old body of research into BFT consensus algorithms such as PBFT .

Casper follows this BFT tradition, though with some modifications. Accountability allows us to penalize malfeasant validators, solving the “nothing at stake” problem that plagues chain-based PoS. The penalty for violating a rule is a validator’s entire deposit. Because proof of stake security is based on the size of the penalty, which can be set to greatly exceed the gains from the mining reward, proof of stake provides strictly stronger security incentives than proof of work.

Dynamic validators. We introduce a safe way for the validator set to change over time . We introduce defenses against long range revision attacks as well as attacks where more than 1/3 of validators drop offline, at the cost of a very weak tradeoff synchronicity assumption . Casper’s design as an overlay makes it easier to implement as an upgrade to an existing proof of work chain.

Within Ethereum, the proposal mechanism will initially be the existing proof of work chain, making the first version of Casper a hybrid PoW/PoS system. In future versions the PoW proposal mechanism will be replaced with something more efficient. For example, we can imagine converting the block proposal into a some kind of PoS round-robin block signing scheme. In this simple version of Casper, we assume there is a fixed set of validators and a proposal mechanism which produces child blocks of existing blocks, forming an ever-growing block tree.

Casper’s job is to choose a single child from each parent, thus choosing one canonical chain from the block tree. If the public key of the validator is not in the validator set, the vote is considered invalid. We presented Casper, a novel proof of stake system derived from the Byzantine fault tolerance literature. Finally we introduced extensions to Casper to defend against two common attacks. 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. The current Casper system builds upon a proof of work block proposal mechanism.

Future Work :

We wish to convert the block proposal mechanism to proof of stake. We wish to prove accountable safety and plausible liveness even when the weights of the validator set change with rewards and penalties. Another problem for future work is a formal specification of a fork-choice rule taking into account the common attacks on proof of stake. Future workpapers will explain and analyze the financial incentives within Casper and their consequences.