# 6824Coin
# A Novel Decentralized Proof-of-Stake Cryptocurrency

6.824 Final Project Writeup

James Lovejoy, Henry Aspegren
{*jlovejoy*, *henrya*} @ mit.edu

May 4, 2018

## 1 Introduction

Currently most cryptocurrencies use Proof-of-Work. However this consensus mechanism consumes large amounts of energy: Bitcoin alone uses more energy than the entire country of Cuba [1]. Proof-of-Stake has been proposed as an energy-efficient alternative. In Proof-of-Stake, a single participant - the leader - is selected to propose the next block. Leaders are selected probabilistically in proportion to how much currency the leader has "staked". Ideally this "stake" constitutes a bond that the leader will lose if she "misbehaves." However realizing Proof-of-Stake in practice has been challenging. Indeed at the time of this writing, no Proof-of-Stake currency has ever achieved widespread adoption[1]. One of the main reasons for this is the *Nothing-at-Stake* problem. In Proof-of-Work, a miner should only mine on one chain - the one that the network will ultimately accept - because the miner must spend a scarce resource - energy - to do so. However in existing Proof-of-Stake designs, the cost of producing a block is negligible. This means that there is nothing to prevent a leader from equivocating and endorsing multiple competing chains. In fact this is may be the rational choice. The *Nothing-at-Stake* problem presents a serious security issue because the network may lose consensus on the 'correct' chain. In this work we propose a novel Proof-of-Stake consensus protocol that solves this problem by cryptographicallly preventing equivocating. In our protocol, the leader can only extend the chain once and exepnds a resource (although not a computation one) to do so.

[1] As defined by being among the top 5 largest currencies by market cap

# 2 Related Work

## 2.1 PeerCoin

PeerCoin [2] was the first attempt at a Proof-of-Stake cryptocurrency. 6824Coin uses a consensus mechanism inspired by PeerCoin's original. However because of the *Nothing-at-Stake* problem PeerCoin uses a centralized "checkpointing" scheme. This is a major security concern since a corrupted checkpointing system is a single point of failure for the entire system.

## 2.2 Algorand

Algorand [3] is another alternative to Proof-of-Work. However Algorand, as described by its authors, is not Proof-of-Stake and has a different security argument entirely. The security of Algorand depends on a Byzantine Agreement protocol that is run among a randomly selected set of participants. These participants are selected according to their coins, but there is no concept of a 'stake'. 6824Coin is different from Algorand since it uses Proof-of-Stake and does not required an agreement protocol to achieve consensus.

# 3 6824Coin Consensus

In 6824Coin , the consensus algorithm must determine the following:

1. Who can propose the next block?
2. Which chain do I follow?

Since the network is decentralized - each participant determines this individually. However the entire network will enforce these rules only if a majority of the participants do so. A sketch of the security argument will be presented in 5.1.

## 3.1 Joining the Stake Pool

To participate in the consensus algorithm, the participant must create a special transaction that *stakes* coins. This *stake* constitutes a bond to the network that will be lost if the staker equivocates and produces a block for more than one chain. The staking transaction consists of following:

$$stake = \{value, R\_point\}$$

The *value* is the total number of coins staked and the $R\_Point$ is a cryptorahpric commitment (@JAMES HERE). This cryptographic commitment has the useful property that if the staker proposes a block for more than one chain he will lose the value of the coins entirely.

### 3.2 Selecting a Block Proposer

In Proof-of-Work, a miner (block proposer) can be anyone who can find a nonce such that

$$SHA\_256(block\_id||nonce) < target\_difficulty$$

$target\_difficulty$ is a adjusted by the network as miners come and go to keep the rate of block generation relatively stable. In 6824Coin , a block proposer is selected according to their *stake* rather than the ability to find a nonce. Instead a block proposer must meet the following criteria:

$$SHA\_256(block\_id||stake||current\_time) < target\_difficulty * (value \times age)$$

where *value* is the coin value of the *stake* transaction and *age* is the number of blocks that have passed since the *stake* transaction. Note that there is no nonce involved and that the search space is finite. It takes exactly one SHA256 hash per output per second to participate in the consensus protocol. By comparison, commercial Bitcoin miners easily use over 4 Billion SHA256 hashes per second, consuming orders of magnitude more energy. Additionally note that the older and more valuable the *stake* transaction, the easier it is to propose a block. The calculation of $target\_difficulty$ is identical.

### 3.3 Reaching Consensus

In Proof-of-Work, the network achieves consensus on which chain by follwoing the chain with the most 'total work'. 'Total work' in this context is the total amount of computational resources devoted to producing the chain. This can be calculated by adding together the $target\_difficulty$ for each block in the chain. In 6824Coin , the network also follows with the chain with the most 'total worl', which is caluclated in the same way as in Proof-of-Work. However in 6824Coin this reflects the *stake*s consumed to produce that chain. In other words the algorithm follows the chain with the oldest and most valuable staked outputs.

### 3.4 Preventing Equivocation

@JAMES HERE

## 4 Implementation

To implement 6824Coin we used CryptoKernel [4], a software package being developed by the Digital Currency Initiative for prototyping cryptocurrencies [2]. CryptoKernel provides out-of-the-box networking and storage code while allowing developers to implement their own consensus modules. We implemented 6824Coin by creating a consensus module in CryptoKernel.

[2] James Lovejoy is the lead developer of CryptoKernel

We have created a genesis block for our coin and are currently running several nodes that are all participating in the consensus process. Anyone can join the network and start using the protocol, using our reference implementation on Github [5]. To join the network simply download the code in the repository and follow the simple instructions.

Because 6824Coin selects block proposers based on the amount of stake a user has, we have had to think carefully about how distribute the initial coins. In our prototype there are no block rewards and all of the coins were initially given to the Authors. However we have implemented a faucet where you can post your public key and we will send you coins. Our goal is to distribute the vast majority of the coins to anyone who would like them. The faucet can be found at TBD.

One concern is that since initially we will have the vast majority of all coins, we will almost always get to propose the next blocks. In the short term we will avoid this situation by simply not staking some our our coins to deliberately reduce our chances of producing the next block. As others join the network using the faucet this will gradually become a non-issue.

# 5 Analysis

Analyzing the security of cryptocurrencies is very difficult. A full analysis considers both the security of the underlying protocol as well as the game-theoretic implications of rational participants. A rigorous analysis is out of the scope of this paper. However we will present a sketch of why we think 6824Coin may be secure.

## 5.1 Security Argument (Sketch)

**WARNING the authors accept no responsability for the security of 6824Coin , and it may be insecure and you may lose all of your money**

The security of 6824Coin rests on the assumption that participants will choose to extend the chain with the most total work. Because of our commitment scheme a participant can only safely chose to use a given staked output once. If participants equivocate and try to re-use the staked output by extending multiple chains, then they will lose their stake entirely. If participants chose to extend some other chain instead of extending the longest chain the implication is more subtle. The participant will not lose his stake, but rather will lose the ability to propose a block using that stake. Thus participants are incentivized to stake on the chain that the network will accept, since otherwise their opportunity to stake will

be wasted. A full analysis of the security of the protocol is out of the scope of this paper and is left to future work.

## 6 Conclusion

In the cryptocurrency community, Proof-of-Stake a hotly debated alternative to energy-intensive Proof-of-Work. However, to the best of our knowledge no one has been able to realize Proof-of-Stake in practice and at scale. Furthermore, despite the hype there are not many documented protocols and, to the best of our knowledge, no open source implementations of Proof-of-Stake. This work presents a clear description of a Proof-of-Stake algorithm that solves the *Nothing-at-Stake* problem and provides an open source implementation that anyone can use. Our hope is that this work will help the community continue to debate and experiment with Proof-of-Work alternatives.

## References


[1] http://bigthink.com/strange-maps/bitcoin-consumes-more-energy-than-159-individual-countries

[2] https://peercoin.net/

[3] Gilad, Yossi, et al. "Algorand: Scaling byzantine agreements for cryptocurrencies." Proceedings of the 26th Symposium on Operating Systems Principles. ACM, 2017.

[4] https://dci.mit.edu/cryptokernel/

[5] https://github.com/henryaspegren/6.824-Final-Project