# BitcoinPoS (BTCS)

W H I T E P A P E R

# *The Conversion of Bitcoin to Proof-of-Stake*

Toshi Konami
dev@bitcoin-pos.org

October 01, 2021

**Abstract**

*Proof-of-stake implemented as a distributed consensus mechanism at the base layer of bitcoin improves the energy and decentralization vs the current proof-of-work consensus mechanism.*

***https://www.bitcoin-pos.org***

# 1 Introduction

Bitcoin was the first well known implementation of a cryptocurrency not dependable on trusted third parties nor central authority. It consists of a distributed chain of blocks, secured, verified, and maintained in a completely decentralized way by using full node operators and miners. Bitcoin's challenge was to reach a solution for the double-spend problem.

In Bitcoin, double-spending is prevented using a distributed consensus mechanism that implements proof-of-work. Distributed consensus is achieved by introducing an opportunity cost from outside (computation time, and energy) and in return providing rewards within the system. Bitcoin is a digital gold that has become under scrutiny for its massive energy consumption. It is estimated that Bitcoin consumes over 125 terawatt-hours and that exceeds the annual electricity usage of Norway. This energy consumption is not going unnoticed as new laws have been put in place in various states to ban proof-of-work outright. As a result, Texas can be seen as centralizing Bitcoin's PoW hashpower as most of the United States mining is flocking to Texas.

Mining centralization is a significant risk in modern day Bitcoin. About 4 mining pools control 75% of Bitcoin's hashpower; this is unacceptable and must be fixed. Why does Bitcoin have such a centralized mining problem? The answer is quite simple, there is a high barrier cost to setup an ASIC mining farm, not to mention noise pollution. The average user cannot cope with these factors and they choose to remotely mine on one of the four main mining pools.

Energy and decentralization issues are becoming harder to ignore. Migrating Bitcoin to Proof-of-stake (PoS) distributed consensus mechanism would solve both of these problems. Why hasn't the majority of the Bitcoin community migrated to PoS? Probably for many factors, the major reasons being billions of dollars already invested in mining equipment and 100% loyalists to Satoshi Nakamoto.

Increasing the energy consumption and driving up the hash rate increases voting power for proof-of-work. To migrate Bitcoin to PoS, BitcoinPoS simply replaces the PoW energy consumption with a one coin equals one vote system (PoS).

## 2 Fork Methodology

BitcoinPoS is the first Proof of Stake (PoS) hard fork of Bitcoin that still allows its users to import their Bitcoin private keys to obtain BTCS with a 1:1 ratio. It was created by staking real BTC; this makes BitcoinPoS a pure PoS transition from BTC PoW. One can mine BTCS using real 2014 BTC keys with a balance or from staking existing BTCS.

BitcoinPoS is a fork of Bitcoin at block 301448 (May 2014). Block 301447 is the last PoW block and all Bitcoin keys (@block 301447) transfer to the BTCS chain as it is a fork of BTC. Block 301448 is the first block that uses PoS. Block 301448 was chosen because it marks the point in which about 60% of all BTC has been mined. We understand that some in the PoW BTC community are too stubborn to make the switch and would do anything in their power to destroy BitcoinPoS. Starting at a block from 2014 gives the community opportunity to mine the remaing 40% of coins to hold off an attack from the PoW community.

At block 301448, it can be seen below that the prevous hash is a PoW hash with many leading zeros(yellow) while the next block hash (301449) no longer uses a PoW with leading zeros, but rather a PoS hash that is rather random looking.

"previousblockhash":
"0000000000000000**1aec522c9842e8d14213c323b7d21cc313a0b8bfcc0923f3**",
"nextblockhash":
"5d2a34aeac66bc6038d71278e472cf0cd1d2f0b1d98b12159e19ec935bffa4dc",

Besides PoS, everything else is identical to Bitcoin. To learn about the halvings, blocktime, software forks, and anything else, please refer to the Bitcoin documentation and the original Bitcoin whitepaper[1].

# 3 Proof-of-Stake Architecture

The BitcoinPoS codebase is nearly identical to Bitcoin. The significant difference exists in the PoS consensus algorithm and when that PoS consensus becomes active. Up until block 301447, BitcoinPoS is 100% identical to Bitcoin PoW. At block 301448 is when the codebases diverge from PoW to PoS.

## Staking Prerequisites

Staking is the process of holding coin in a wallet to support the operations of a blockchain network. Staking forces locking your coin to receive rewards. BitcoinPoS only locks coins for 2 transactions, this allows the user to have access to their coin quickly.

The following prerequisites apply to staking BTCS:

–The coins to be staked need to be matured; this means that the unspent outputs (UTXOs in short) need to have a depth in the main chain of at least 2 blocks.

–The coins to be staked need to be in compatible address/transaction types (P2PK and P2PKH are supported).

-Use a legacy address that starts with a ‘1’ when staking.

## Block Structure

BitcoinPoS blocks must abide by the following rules:

• Must have exactly 1 staking transaction

• The staking transaction must be the second transaction in the block

• The coinbase transaction must have 0 output value and a single empty vout

• The block timestamp must have it's bottom 4 bits set to 0 (referred to as a "mask" in the source code). This effectively means the blocktime can only be represented in 16 second intervals, decreasing it's granularity

• The block's kernel hash must meet the weighted difficulty for PoS

• The block hash must be signed by the public key in the staking transaction's second vout. The signature data is placed in the block (but is not included in the formal block hash)

• The signature stored in the block must be "LowS", which means consisting only of a single piece of data and must be as compressed as possible (no extra leading 0s in the data, or other opcodes)

• Most other rules for standard PoW blocks apply (valid merkle hash, valid transactions, timestamp is within time drift allowance, etc)

# 4 Proof-of-Stake Mining

Long ago, Bitcoin had a thread dedicated to mining blocks using PoW. Since a CPU has almost zero chance of finding a PoW solution, the developers have removed the ability to mine Bitcoin within the wallet. Instead the wallet had a block template RPC call that allows external stratum server to talk with and delegate the hard repetitious work to that of an ASIC.

Since PoS relies on the CPU wallet for mining, BitcoinPoS must bring back the ability to mine within the wallet application. The following is pseudo code on how the PoS stake mining process works.

-Create a mining thread.

-Ensure blockchain is synched and connected to at least four peers.

-Ensure we are now PoS (past block 301447)

-Grab a list of coins transactions that are eligible for staking. (have at least 2 confirmations)

-Create an empty block

-Enforce timestamps to be on 16 second boundaries. (This reduces the ability to generate more potential valid blocks)

-Verify we are still at the latest tip of the blockchain.

-Grab transactions from the mempool and add to the block.

-CreateCoinStake(look at all txs and try find a solution to the equation shown on next page)

-Sign Block

-Verify hash target and signature of coinstake tx.

-Process new block and send header/block to connected peers.

**NOTE: By default, the wallet automatically starts mining when started.**

# 5 Proof-of-Stake Computation

Like PoW, PoS also has a computation to calculate in order to submit a block. In Bitcoins's PoW, several variables can be modified at high speed to try and find a solution. In BitcoinPoS's PoS, variables do not exist that can modified at a high rate of speed; this collapses the solution into large dependence on the amount of stake held.

## BitcoinPoS kernel protocol

*coinstake must meet hash target according to the protocol:*
*kernel (input 0) must meet the formula*
*hash(nStakeModifier + blockFrom.nTime + txPrev.vout.hash + txPrev.vout.n + nTime) < bnTarget * nWeight*
*this ensures that the chance of getting a coinstake is proportional to the*
*amount of coins one owns.*
*The reason this hash is chosen is the following:*
*nStakeModifier: scrambles computation to make it very difficult to precompute future proof-of-stake*
*blockFrom.nTime: slightly scrambles computation*
*txPrev.vout.hash: hash of txPrev, to reduce the chance of nodes generating coinstake at the same time*
*txPrev.vout.n: output number of txPrev, to reduce the chance of nodes generating coinstake at the same time*
*nTime: current timestamp*
***block/tx hash should not be used here as they can be generated in vast quantities so as to generate blocks faster, degrading the system back into a proof-of-work situation.***

# 6 Variations in Staking

*hash(nStakeModifier + blockFrom.nTime + txPrev.vout.hash + txPrev.vout.n + nTime) < bnTarget * nWeight*

From the above equation, there are 6 variables that can be modified(*bnTarget* moves slowly over time*)*, however none of them at the current time step have the ability to be updated rapidly except for *nWeight* , txPrev.vout.hash , txPrev.vout.n. This leaves an attacker thinking of how to manipulate the above equation using *nWeight and* txPrev.vout.hash and txPrev.vout.n.

**1) Create thousands of tiny transactions:**

Creating many small transactions gives the benefit of many more equations in a short amount of time with the hope that one of the solutions is found. This doesn't provide any benefit because when there are N transactions, the *nWeight* for each attempt at the solution is reduced by a factor of N making it more difficult to satisfy the threshold.

**2) Create one large transaction:**

Creating one large transaction gives the benefit of the largest possible *nWeight*, however this will only provide one attempt every 16 seconds.

The above attack method outcomes are based on the assumption that the hash function creates a well formed uniform random variable. Therefore the user does not generally need to worry about how many transactions are in a wallet when considering maximizing mining rewards. There is an extreme case though. Consider a case when millions of stakeable transactions exist. The CPU could be put under too much load that it cannot finish iterating through the set of coins within the 16 seconds time interval, in this situation, it would be better to to send all coins to a new address to create a lower amount of transactions to iterate through.

## References

*[1] https://bitcoin.org/bitcoin.pdf*