Transactions as Proof-of-Stake

by Daniel Larimer
dlarimer@invictus-innovations.com

November, 28th 2013

Abstract

The concept behind Proof-of-Stake is that a block chain should be secured by those with a financial interest in the chain. This paper will introduce a new approach to Proof-of-Stake that utilizes coin-days-destroyed by every transaction as a substitute for the vast majority of the security currently provided by Proof-of-Work. Unlike prior Proof-of-Stake systems in which only some nodes contribute to the proof-of-stake calculation, we present a new approach to Proof-of-Stake whereby all nodes generating transactions contribute to the security of the network. The result we speculate that the network immune to known attacks against Bitcoin or Peercoin.

Background

The concept of Poof-of-Stake was first introduced as a means to counter known attacks on the Bitcoin network, primarily the 51% attack.

Existing Proof-of-Stake systems such as Peercoin are based upon 'proof blocks' where the target the miner must meet is inversely related to the coin-days-destroyed. Someone who owns Peercoins must choose to become a Proof-of-Stake miner and commit some of their coins for a period of time to secure then network.

The creators of Peercoin recognized that Proof-of-Stake in this form was insufficient so they rely upon a hybrid system whereby both Proof-of-Stake and Proof-of-Work are used to secure the network. As the difficulty in the Proof-of-Work increases the block reward decreases which should work to automatically throttle the amount of Proof-of-Work mining.

Despite using Proof-of-Stake, Peercoin still relies on 'mining' with respect to Proof-of-Stake which inherently limits the number of people and the percentage of the money supply available to secure the network via Proof-of-Stake. The incentive given for Proof-of-Stake mining is a an average 1% return on your stake. This incentive is currently dwarfed by the 8% inflation paid to the Proof-of-Work miners.

Peercoin isn't the only proof-of-stake system proposed. Others involve various forms of signature blocks, lottery selection of signers proportional to transaction size, etc. None of these ideas have been as throughly adopted as Peercoin in the market.

Despite Peercoins success, its application of Proof-of-Stake does not fully solve the problem of double spending or denial of service. Ultimately the network is still secured by Proof-of-Work and it is still possible to mine secret alternative chains that could be used to perform a double spend. An actor such as a government could still acquire enough hashing power to win over 51% of the proof-of-stake blocks and all of the proof-of-work blocks. All Peercoin has achieved is to increase the cost of attacking the network without changing means by which the network can be attacked.

The Purpose of Mining

In most literature regarding crypto-currencies and mining, the focus has been primarily on coin distribution and securing the network from a monopoly of hashing power. Mining rewards are seen as necessary fees used to purchase 'security' and the general theory is that that the more mining the better. Existing Proof-of-Stake systems carry over this mentality and pay people who 'mine' on a Proof-of-Stake basis.

For the purposes of this paper, we would like to focus on the more critical role that mining plays that is entirely independent of security or currency creation. This role is to determine in a decentralized manner whom will publish the next block. A purely proof-of-stake system must still determine which of 1000's of computers will build and publish the next block. These blocks must come at regular intervals large enough that the network may reach consensus between every block.

There is no cost for producing or broadcasting a block and in a centralized system and a single computer could be assigned the task of publishing blocks every couple of minutes. In a decentralized system this task my be distributed far and wide. Because the cost of producing and broadcasting a block is essentially 0, block rewards worth as little as a couple of cents would be enough to motivate this behavior.

The challenge for a decentralized system is to find a way to limit broadcasting to at most one or two nodes every couple of minutes. The solution to this problem is a traditional proof-of-work whereby statistically nodes that perform no additional work will randomly select just a single node to broadcast the next block.

Suppose every node in the network were to always mine at a constant rate of one hash per minute. The difficulty would adjust such that even at this very low and steady hash rate, only one block would be produced per minute.

With traditional Proof-of-Work mining someone could take such a low hash rate as an opportunity to take control over every block produced; however, in a ideal Proof-of-Stake system it wouldn't matter how much hash power someone had they would be unable to use that hash power to block transactions nor effect a double spend.

We therefore assert that the most efficient Proof-of-Stake algorithm will result in no financial incentive to increase hash power used for Proof-of-Work. Using this understanding the hashing algorithm becomes irrelevant and network security is no

longer subject to centralization in the hands of miners, botnets, or any other organization with a large amount of capital.

The Reason Proof-of-Work Provides Security

If we are to remove Proof-of-Work from a block chain's security model then we must fully understand the nature of the security provided by Proof-of-Work and find a suitable replacement. On network's like Bitcoin the computational power used to find hashes is extremely high. This high level of hash power can be used as a proxy for total combined investment into a particular block chain. A rational individual can generally assume that the largest investment represents majority consensus as to the the truth.

Making this assumption opens the door for attackers to abuse the trust placed in this metric and perform several known attacks including: double spend, denial of service, and selfish mining. All of these attacks operate based upon the creation of alternative chains. The security provided by the decentralized hash power is via making counterfeit chains expensive to produce. With this security model there is a direct relationship between the cost of the security and the cost of attacking the network.

The denial of service attack is perhaps the most destructive on the network, while the double spend attack is potentially profitable. Both attacks disrupt the smooth operation of the network and therefore detract from the value of the currency. A government that wished to legalize, but control a crypto-currency could mine at levels that are not profitable for any private organization and thereby achieve a monopoly that allowed them to filter transactions at will.

If we are to replace proof-of-work then the replacement must be capable of preventing counterfeit chains from being easily produced. Lets look at the key to using hash power to produce counterfeit chains. Both double spending and selfish mining depend upon the attacker maintaing a secret alternative block chain that is longer than the public chain. All that is required to stop these attacks is to make the production of secret chains impossible with less than 50% of the money supply.

The denial of service attack does not require a secret block chain. To prevent this attack the selection of the longest block chain must be based upon a metric other than Proof-of-Work.

Transactions as Proof-of-Stake

Every transaction on the network carries with it an implicit Proof-of-Stake in the network. The creator of the transaction wants the network to accept it and the receiver of the transaction is making decisions on whether or not to ship goods based upon whether or not the network has accepted the transaction. It is clear that those behind the transaction have a stake in the health of the network. After all, the network is worthless if transactions cannot be executed as expected. A well functioning network will have

thousands of transactions every single block. This represents thousands of stake holders who could be contributing to the security of the network.

A coin-day represents the number of days since a particular coin was last transacted on the network. At any given point in time there exist a limited number coin-days and they accrue in the hands of those who hold large balances for a long period of time. As a result coin-days can be seen as a proxy for stake in the network. Coin-days are destroyed every time there is an transaction involving those coins and therefore cannot be reused.

In order for a 51% attack to be successful in a Proof-of-Work system, the attacker must keep their alternative chain secret. Once they have locked in the profits from their first spend, they can broadcast the longer secret block chain which will invalidate the original transaction. Keeping solved blocks secret is also used in the selfish-mining attack which can be effective with much less than 51% of the hashing power.

In order to prevent this kind of behavior we must make it impractical for miners to maintain secret block chains. If every transaction that is broadcast contains the hash of a recent block and the block chain enforces the rule that the transaction can only be included in block chains that build off of that block then no one will be able to build secret block chains that leverage the coin-days-destroyed of transactions in the public chain.

Now that the transactions are committed to a certain public chain, the best block chain can be measured by coin-days destroyed first and Proof-of-Work second. As the coin-days destroyed by a block increases, the difficulty and block reward should approach but never reach 0. In a very active network with heavy transaction volume the amount of mining required also approaches, but never quite reaches 0.

The base difficulty for the Proof-of-Work must still adjust to maintain a regular block interval as the number of computers in the network grows, but as transaction volume increases (and thus coin-days-destroyed increases) the mining difficulty will increase such that after discounting for the average transaction volume the expected time to find the next block stays around the target interval even as the value of the mined block approaches the cost just sufficient to justify assembling and broadcasting the block.

Under this model blocks that contain below the average transaction volume will become exponentially expensive to mine. Meanwhile blocks that contain above average transaction volume will become exponentially easier to mine. The consequences are that security of the network has been entirely decoupled from the cost of securing the network. Empty blocks will be rejected in favor of blocks that destroy more coin-days regardless of the proof-of-work. This will prevent denial of service.

Secret alternative block chains would require more coin-days-destroyed than the public network. This kind of attack is not sustainable due to the time it takes to acquire coin-days.

Withholding Transaction Attack

As a consequence of giving miners that include transactions an advantage, there exists the potential that some miners may choose not to relay the transactions they receive. Fortunately, it is very easy for a node to detect peers that fail to relay transactions because every node in the network should receive an inventory broadcast from every peer they are connected to. Any peer that consistently fails to relay transactions can be identified by simple and automatic analysis of local network broadcasts. Once identified they can be disconnected and have no hope of mining anything at all. The process used to detect misbehaving nodes also optimizes network performance by minimizing redundant broadcast messages.

This particular behavior is only incentivized with large block rewards. If the network were to pay next to nothing for finding a block then there is no significant incentive to withhold transactions.

Attack on the Network by Large Shareholders

Given a money supply M there exists 365*M coin days per year that can be consumed validating a block. Someone with 50% of the money supply would be able to consume 0.3% of the available coin-days per block on average. Someone with 0.004% of the money supply who transacts just once per year would also consume 0.3% of the coin-days when they make their transaction. Overall however, someone attempting to attack the network by owning a large number of coins that they transact every block would have disproportional influence in the short term.

To counteract this kind of attack, the first 24 hours after a transaction is spent should not count toward accumulation of coin-days. For example: given 144 blocks per day, someone who keeps their coins for 288 blocks has 144x as many coin-days as someone who keeps their coins for just 145 blocks. Someone with a large number of coins would only be able to exercise their influence once every 145 blocks and their influence would be less than 1% of someone who holds the same number coins for 48 hours.

In practice, no one would be able to accumulate enough of the money supply to earn enough coin-days that they could have a oversized influence over more than a couple of blocks per day on a sustained basis. Even with a lot of coin-days they still have to do some mining to win the block. A double spending attack requires an individual to maintain a significant advantage for many consecutive blocks. This will not be achievable if the proper waiting period is selected before coin-days may start accumulating.

The security of the network is based upon the fact that no one can generate coin days fast enough to exercise them every block in a way that would allow them to generate alternative chains that destroy more coin days than the public chain. As a result it

becomes economically impossible to perform a double spend attack even with 50% of the hash power and 50% of the money supply.

Occasional Double Spend Attack

There exists the potential for someone to save a large number of coin-days and then use these coin-days to execute a single double spend attack unless certain protective actions are taken. The cost of performing this double-spend attack would be proportional to the average coin-days-destroyed per block. While an attacker cannot maintain such an attack, given a small percentage of the money supply (0.006%) an attacker could maintain above average coin-days-destroyed for 6 blocks, long enough for a double-spend attack using Bitcoin's security metric.

Bitcoin's security metric of 6 confirmations indicating near certainty that the transaction is final only applies to proof-of-work based security. Absent proof-of-work the number of blocks that have confirmed your transaction is meaningless. Instead, the metric users should care about is coin-days-destroyed since your transaction. The higher this number, the more expensive a 'random' double spend would become. In theory it is possible to have security against a double spend attack equivalent to Bitcoin after a single confirmation if that block destroyed enough coin-days.

So, how many coin-days must one wait to be destroyed to achieve a degree of confidence on par with Bitcoin?

Market forces push the cost of mining toward the value of the block reward. Someone wishing to carryout a double-spend would have to mine seven secret blocks while the public network was mining 6 blocks. With Bitcoin as of December, 2013, this would cost about $200,000 in electric costs plus the cost of the ASIC capital capable of consuming $200,000 per hour in electricity. Needless to say no one would be able to carry out a double spend attack in secret because that kind of power draw and infrastructure would make the culprit easy to identify.

The security provided by Bitcoin is not based upon the 'cost of the attack', but instead on the impossibility of executing the attack anonymously. In effect, Bitcoin gets its security from the centralization required to accumulate enough hashing power to execute a double spend attack. The reality is that the cost-of-capital and electricity is entirely covered by the value of the coins produced. Therefore the 'cost to the attacker' is essentially nothing except the risk of a lawsuit.

With proof-of-stake the cost of capital is holding funds and not moving them long enough to build up enough coin-days. Unlike ASICs, the value of the capital remains liquid and could even appreciate over time. The 'cost-of-electricity' can be thought of as coin-days-destroyed. Both the capital and coin-days are possible to own anonymously and therefore the primary means by which Bitcoin prevents double spend attacks cannot be employed directly.

If we assume a network with the same value as Bitcoin and the requirement to move your money at least once per year, then the cost of a double spend attack executed once per year would be owning 0.01% of the money supply which would equal $1.6 million dollars held for 12 months without moving.

Anyone with that kind of money would not bother attempting a double-spend over anything trivial. Therefore, I submit that for most ordinary transactions a double spend is very unlikely and the losses from such a double spend attempt would be minimal. Furthermore, the attacker could only perform it once per year. Any transactions large enough to justify an attempted double-spend would not be done anonymously.

Unlike Bitcoin where your confirmation time is entirely dependent upon miners finding blocks, someone wishing to accelerate the confirmation time of one transaction can do so by confirming it with some of their own coin-days. Large transactions can also be broken up into multiple parts where the later parts confirm the earlier parts.

Overall the combination of knowing your customers and waiting longer for larger amounts can easily make the potential for a double spend attack essentially 0. Despite this minimal risk, there is still one more level of protection that can be added.

In a connected network there is never a reason for two chains to diverge by more than one or two blocks. If a new chain is discovered that is 'longer' than your known chain then chances are it is an attempted double-spend attack. Nodes that have been connected the entire time should assume they were on the public fork and that the new chain was being built in secret by a minority. Therefore, any chain reorganization beyond two blocks should require a much higher threshold than the chain you already have. This threshold could be as high as 24 hours worth of blocks which would significantly increase the cost of a double spend attack to the point of requiring 0.3% of the money supply.

Interaction with Dividend Paying DACs

Dividends are paid from the profits earned by the DAC. Dividends can only be collected by issuing a transaction that would result in the destruction of coin-days. Anyone seeking a compound return on their shares must regularly create transactions and thus contribute the coin-days destroyed to validation of the network. Because no dividends are paid on shares held for less than 24 hours someone with a large balance would lose all of their dividends if they attempted to reuse their large balance on a regular basis.

Offline Transactions

Offline Transactions would not necessarily have access to the current head of the block chain at the time they are signed. Therefore, they would be unable to verify the current head block at the time they are signed. Therefore, the only coin-days that count for the purposes of the transaction are those between the output and head block included in the transaction. For chains that pay dividends, the same rule can be applied toward

collecting dividends. The result of these policies is to strongly encourage transactions to sign a recent block rather than rely on offline transactions. This will maximize the number of coin-days available to secure the network.

Migrating Transactions from Minority Forks

With existing block chain designs, transactions are relatively independent from the blocks which contain them. In the event that there is a chain fork transactions from the minority fork can be migrated from the minority chain to the majority chain when the networks reconnect. The only transactions that are invalidated are those transactions which depend upon one of the coinbase transactions from the minority fork. It is for this reason that Bitcoin requires coinbase transactions to mature 120 blocks before they can be spent.

In practice it is rare for there to be a fork longer than a couple of blocks. For all practical purposes the 6 blocks required for a normal confirmation is enough to be secure against a chain fork. The current recommendation for anyone who notices a chain fork is to stop all transactions until minority network can rejoin the majority or people could be the victim of double spend attacks.

Under our approach, transactions that migrate from a minority fork would not contribute to the coin-days-destroyed. This will insure that chain forks do not require individuals to re-issue transactions.

Efficient Implementation

A straight forward implementation would simply add the hash of the head block to every transaction which would add 32 bytes per transaction. However, all that is required is the block number of the head transaction. The hash of that block can be factored into the digest used to sign the transaction without adding 32 bytes to every transaction in the block. When the transaction is being verified the hash can be looked up from the block number alone.

Vesting of Mining Rewards

With CPU based coins there is a heavy incentive for miners to mine when it is profitable and then dump the coins into the market suppressing the price. With the advent of ASICs Bitcoin mining has become a capital intensive operation that requires miners to heavily invest in ASICs that will take 3 to 12 months to recover their capital costs if they are lucky enough to stay ahead of the difficulty curve. This has the effect of giving miners a long-term interest in the success of the coin because otherwise they will never recover their costs and make a profit.

With CPU based coins there are almost no capital costs and machines can be rented by the hour from Amazon. This means people are mining for profit alone and have no interest in acting in ways that support the growth of the coin they are mining.

Unfortunately, ASIC based solutions also tend to centralize control. The goal is to decentralize mining and therefore rely on a mining algorithm that performs best on the CPU. Fortunately, the economic motivations created by ASICs can be simulated by requiring mined coins to vest over 3 to 6 months and charging a penalty for selling the coins early. Vesting is thus a form of Proof-of-Future-Stake which makes it significantly more costly to use the proceeds from a 51% attack to fund the attack.

Using this kind of vesting approach is only necessary during the bootstrap phase of a new coin until enough coin-days and transaction volume can be accumulated. As mining rewards approach 0 vesting becomes irrelevant.

Conclusion

In this paper we have provided a simplified Proof-of-Stake algorithm that allows all users of the network to contribute to the security of the network against attacks. It should be economically infeasible for any actor to maintain a secret block chain that contains more coin-days destroyed than the public transaction ledger. The techniques presented solve the 51% attack, the selfish-miner attack, and provide protection against double-spending all while requiring no measurable mining effort.

Because Proof-of-Stake can eliminate the need for mining rewards, they also eliminate the need for inflation while also eliminating wasteful consumption of energy for proof-of-work. In a world with an increasing number of block chains, security through proof-of-work becomes fragmented without merged-mining and merged-mining has its own overhead. With this new approach to Proof-of-Stake merged mining is no longer required and an unlimited number of block chains can be supported without compromising the potential security of any individual chain.