Delayed Proof of Work (dPoW) Whitepaper

jl777 | 30th Aug, 2016 | v1.0

ABSTRACT

In this whitepaper we discuss a completely new cryptocurrency consensus mechanism, that is as secure as the proof-of-work blockchain to which it attaches itself to (In this case: Bitcoin), but does not require computing power and energy to be wasted.

This system is called Delayed Proof of Work (dPoW) and is achieved by notarizing blocks created in the initial blockchain on the Bitcoin blockchain, ensuring that once the information is engraved on the Bitcoin blockchain, it would be required both blockchains in question to be compromised.

EXECUTIVE SUMMARY

INTRODUCTION

In 2008, Satoshi Nakamoto published the white paper introducing Bitcoin, the first form of peer to peer, trustless cryptographic currency, a system that relies on Proof of Work to mint new coins and to maintain itself. In PoW, contribution is weighted by the computational power, rather than one threshold signature contribution per party, which allows anonymous membership without risk of a Sybil attack.

Bitcoin focuses on the transfer of value from point A to point B, but since the inception of Bitcoin, hundreds of new cryptocurrencies have surfaced. Some focus on improving this system with faster, cheaper or anonymous transactions, while others use the blockchain to create completely new systems or applications. These cryptocurrencies (excluding the clones of clones) often serve a specific purpose or function.

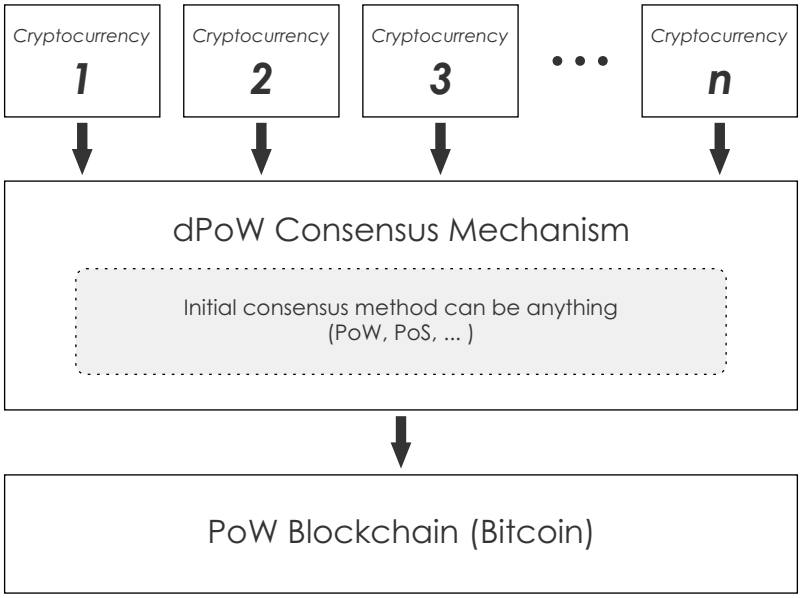
Many types of consensus systems have been created over the years, but despite being more efficient than Bitcoin, they often pale in comparison to Bitcoin's security, and thus are not suitable for use in high value transactions. Today, thousands of nodes secure the Bitcoin network and more than half a million dollars are used daily to secure one single cryptocurrency. The question then arises about how we could make the overall system more efficient?

For other cryptocurrencies to become as secure as bitcoin they would have to secure their network with as large hashing power. However this is not economically possible and would make the overall system even more energy inefficient. The solution would be to allow other other cryptocurrencies to take advantage of bitcoin’s hashing power.

Delayed Proof of Work is a solution that utilizes multiple existing methods into a single hybrid consensus system that at the same time is as energy efficient as PoS while being secured by bitcoin PoW. Thus dPoW allows even the weakest of blockchains to benefit from bitcoin's hashrate and this in turn makes Bitcoin's power usage much more eco-friendly as it is securing the entire ecosystem of dPoW in addition to itself.

THE INITIAL CONSENSUS METHOD

The initial consensus method can be anything and the delayed Proof of Work consensus can be build on top of it. Thus at its core the dPoW blockchain could be either a Proof of Stake (PoS) or Proof of Work (PoW). Furthermore, the dPoW consensus method can attach itself to any PoW blockchain, but because Bitcoin has the highest hashrate it is an obvious choice.



Picture 1: Any consensus method can be expanded into dPoW and then attached to a PoW blockchain.

No matter what the initial consensus method is the other cryptocurrencies can use the dPoW blockchain to secure their network. Because the dPoW blockchain is secured by the Bitcoin hashrate also the other cryptocurrencies attached to dPoW blockchain are secured by Bitocin's hashrate. In other words, coin X will send dPoW transactions to dPoW blockchain and the dPoW blockchain will send transactions to the Bitcoin blockchain.

Every cryptocurrency would be able to attach itself directly to the Bitcoin blockchain but because of the transaction fee difference it is expected that other blockchains would prefer to utilize an already working dPoW system.

NOTARY NODES

Notary nodes are needed to record data to the Bitcoin blockchain. Thus the dPoW consensus method will end up with two different nodes: notary nodes and the other 'normal nodes'. Notary nodes have to be elected where as normal nodes can be run by anyone.

In order to create any data that the entire blockchain will depend on, Sybil attacks must be prevented. Additionally, higher reliability and consistency can be achieved with a smaller set of high performance nodes than a larger but random collection of peers. From Bitshares/Steemit we have learned about delegated PoS system where the PoS stake is used to vote for witness nodes, which are then chartered with the power to create blocks. Delayed Proof-of-Work's need for a set of nodes to rely on is well served by this type of model and they are called notary nodes.

Notary nodes are elected by stakeholders and since those notary nodes will earn block rewards it is expected that the financial interests of the stakeholders is to be voting for notary nodes that they are in control of or at least comfortable with. This system isn't fully decentralized from a purists standpoint, but with 64 notary nodes up for election by stake combined with the large scale distribution it is expected to have a very good representation that will make any type of 51% attack highly improbable.

We end up with a group of 64 notary nodes that are responsible for notarizing the blocks from the dPoW blockchain on the Bitcoin blockchain. Both the notary nodes and the stakeholders that vote for the notary nodes have a strong economic incentive to behave honestly and to vote truly.

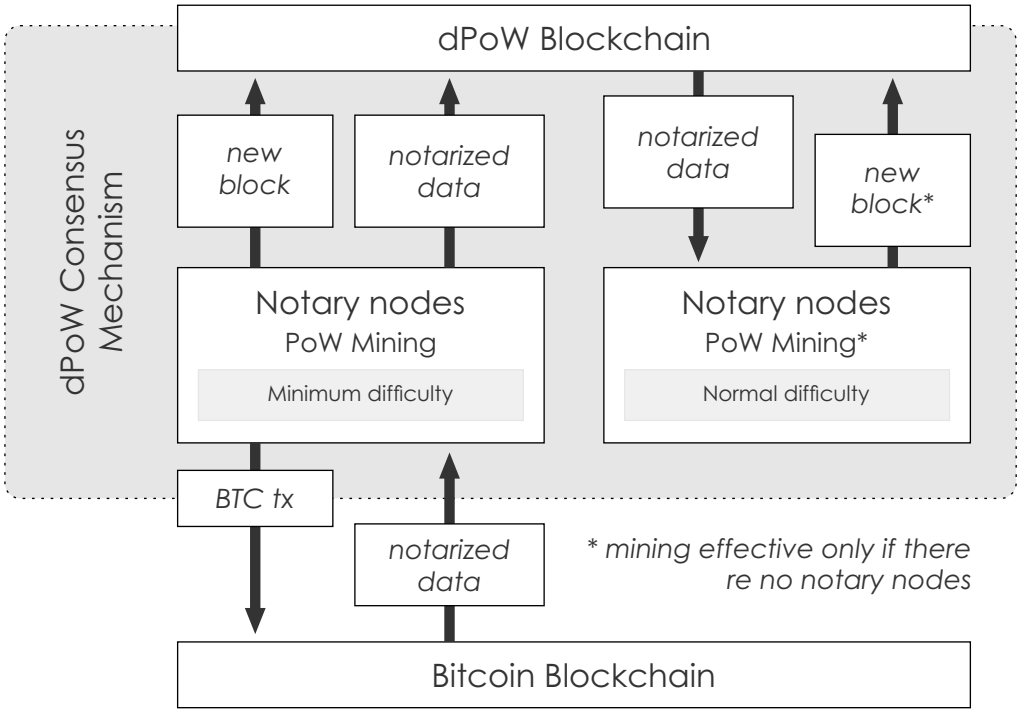
GRACEFUL DEGRADATION

Without the notary nodes there cannot be the Bitcoin provided security. However, with proper design it is possible to allow the dPoW network to continue seamlessly using only its initial consensus security.

One method to achieve graceful degradation would be to use the Peercoin style PoS that is based on utxo age and other parameters so that each node is staking normally and the notary nodes provide an overlay of the notarized hashes using the same method for third party chains. This way, the dPoW network can proceed as normal even if all the notary nodes have disappeared.

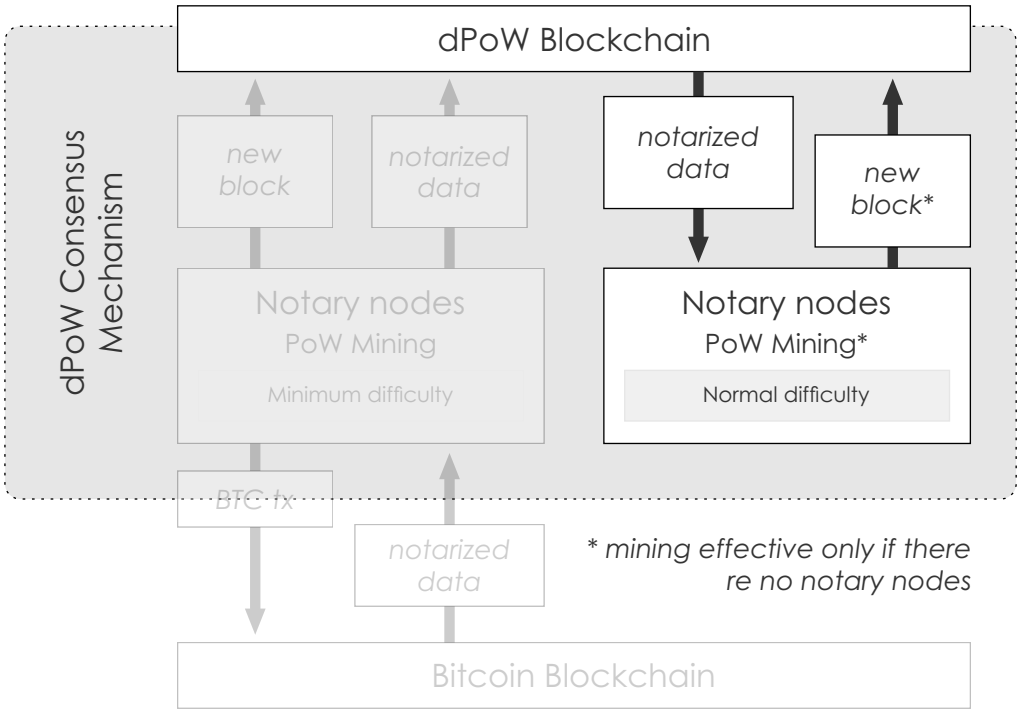
Peercoin allows both PoW and PoS to coexist. The PoS calculations are given a large preference over the PoW by shifting the calculated hash value. By using a similar method, two totally different methods of winning a block can coexist and this allows the notary staking to also be an optional aspect of the system. Of course, in the absence of the notaries, some end user nodes will have to be staking.

Another option is to use PoW as the initial consensus method. In this case both notary nodes and the other normal nodes can mine. However notary nodes will find most of the blocks as they can mine with minimum difficulty where as other nodes have to mine with normal difficulty.



Picture 2: Delayed Proof of Work consensus mechanism for an initial PoW consensus method.

If for some reason all the notary nodes would disappear, then the other nodes with normal difficulty would be able to mine and find blocks. The nodes would get the already notarized data from the dPoW blockchain and thus the already notarized records would still be as secure as Bitcoin.



Picture 3: A scenario where all notary nodes would have gone offline.

The dPoW consensus mechanism isn't depended on the notary nodes, as without them the blockchain could continue its operations. If all the notary nodes would go offline no new data could be notarized to the bitcoin blockchain. The other normal nodes cannot send the group signed bitcoin transactions, but they could read and validate the already notarized history from the dPoW blockchain.

The key is to understand that dPoW is a system of different components that are designed to work together, but are able to perform the critical function of block generation in isolation. The end result is a dPoW that is robust and decentralized for creation of blocks, with the notarized data being generated by the elected notaries. It is thus not possible to categorize dPoW as any single type of consensus.

DELEGATED STAKING

Since dPoW utilizes notary nodes for its security, there is little reason to require all the end user nodes to do any PoS staking. By using a ledger snapshot that is taken every 1000 blocks, the balance of all addresses can be known. By sorting this ledger based on the pubkey, each address will obtain the same index on all nodes. Each block, each notary calculates the best PoS hit value for 1/N'th of the addresses and this is shared with all the other notary nodes, which allows an efficient search of the winning address. The notary responsible for that address creates the block and signs it, awarding the 5% APR staking reward to the winning address and a block reward to itself. This process encourages consolidation of accounts to maximize the staking rewards and this in turn is expected to prevent a massive increase in addresses so that the computation required per block stays manageable.

A majority of notary nodes needs to approve the winning block. As a brute force reorganization protection, each notary could brute force search all possible addresses to be assured that the submitted winning address is unbeatable. With a ledger snapshot every 1000 blocks, the address balances used for the PoS calculation will be 1000 to 2000 blocks in the past. This avoids the gaming of the stakes by moving the funds around to a winning address as there is no way to know 1000 blocks ahead of time which address will win.

If the balances are 100% unchanging, it would be possible to calculate all 1000 future blocks and which address will win and to calculate variations of this to guarantee a winning block at the next snapshot. Maybe even to pre-calculate a winning chain of addresses and move funds into them in precisely the right amounts. The problem with this is that the gains are minute and also nothing prevents a second party from doing the same thing. The important thing to note is that the potential gain is from increased block rewards and not from double spending.

Another possibility is to bypass the staking process entirely and award accrued interest when a utxo is spent. Despite the method used it is independent of the dPoW consensus itself.

DELAYED PROOF OF WORK DETAILS

Let's assume the existence of a properly elected set of notary nodes with an honest majority. The network submits new transactions to *sqrt(N)+1* randomly selected notary nodes, each in turn forwards it to *sqrt(N)+1* other notary nodes. Using this process, most submitted transactions will be known to almost all the notaries within one hop. A reconciliation process can be used to ensure that any notary node can obtain the missing ones, but this is not necessary as during the signing stage all pending transactions will be used to construct the block. With the notary node being awarded the transaction fees each is motivated to include as many paying transactions as possible, though nothing prevents a node from creating an empty block. It is presumed that such a misbehaving notary will soon be replaced via voting and the damage done in the meantime is a bit of a delay.

Notice that the speed of block generation can be quite fast, especially with the requirement of high performance servers to be a notary node, so if anything, effort will be needed to slow down the blocks to keep pace with the desired blocktime. Conveniently, there is more work for the notary nodes to do. As soon as the block is finalized, its blockhash is known and this blockhash is put into a group signed bitcoin transaction. To create a group signed transaction among an unknown combination of 33 of 64 nodes is a bit of a puzzle, but if this is a stumbling block, the answer is given in the appendix 1 below.

A timing question is when (or even if) the group signed transaction should be submitted to the Bitcoin network. For now, let us spend Bitcoin txfees freely and just immediately broadcast these group approvals as soon as we get them. The second appendix will explore ways to optimize this, but it is just a cost optimization and not integral to the dPoW protection.

We find ourselves now with a stream of blockhashes appearing in the Bitcoin blockchain since both of these are already approved by the majority of notaries, which in turn are assumed to reflect the interests of the stakeholders. There is one final thing that is needed is to add a new consensus rule into the block reorganize loop. It is a simple rule and it is to refuse to reorganize a block that has been notarized by the Bitcoin blockchain. With this it becomes impossible to reorganize a dPoW chain without also reorganizing the Bitcoin blockchain itself.

A more in depth consideration of the "refuse to reorganize" rule leads us to discover some timing based edge cases, i.e. when is a specific hash properly notarized? With "now" being a relative concept, we need to rely on the block timestamps to determine which was first by pretending that the timestamps on the two chains are from the same clock. Granted this is not really the case, but it serves to deterministically decide whether to reorganize past a block or not.

ATTACKS

Let's discuss possible attack situations when we have notary nodes online, and lastly a situation when all notary nodes would be temporarily shut down.

- **Double spending attacks.** In a double spending attack, the adversary wishes to revert a transaction that is confirmed by the network. The objective of the attack is to issue a transaction, e.g., a payment from an adversarial account holder to a victim recipient, have the transaction confirmed and then revert the transaction by, e.g., including in the ledger a second conflicting transaction. In order for a double spending attack to work for a bitcoin notarized transaction, the bitcoin blockchain will also have to be rewritten. So this attack is deemed to be impractical.

- **Transaction denial attacks.** In a transaction denial attack, the adversary wishes to prevent a certain transaction from becoming confirmed. For instance, the adversary may want to target a specific account and prevent the account holder from issuing an outgoing transaction. As long as a node can connect to any of the notary nodes directly or indirectly, the valid transactions will be confirmed.

- **Eclipse attacks.** In an eclipse attack, message delivery to a node is violated due to a subversion in the peer-to-peer message delivery mechanism. In event of an eclipse attack where the attacked node is only connected to attacker's nodes, the best defense that can be achieved is to have an externally verified chaintip. However, if connection to just a single honest node can be achieved, the true mainchain can be discovered by the attacked node. In most cases the notarized data present in the dPoW chain will allow such a bootstrapping but even in the case where the attacker has created an entirely new chain from genesis, the BTC chain can be queried to find the true mainchain.

- **Nothing at stake and past majority attacks.** As with the eclipse attacks, just a single honest node with the true notarized mainchain is enough for new nodes to follow it. In the event the notarized data in the dPoW chain is intact, that is all that is needed to find the true chain. Even in the event the attacker has constructed a fake chain, then the BTC chain can be queried for the notarized data and with just a single honest node, new nodes can reconstruct the valid notarized mainchain.

- **51% attacks.** A 51% attack occurs whenever the adversary controls more than 51% of notary nodes. In this case, the attacker will be able to prevent specific transactions from going into the blockchain, but once a block is notarized onto the BTC chain, even if 51% notaries are controlled by the attacker, the notarized data can't be undone. In the even the attacker prevents new blocks from being created by notaries, the fallback consensus method will start generating new blocks.

- **Notary node attack.** If all the notary nodes would be brought offline simultaneously the dPoW network would effectively become a normal blockchain with its initial consensus method (PoW or PoS). The historical notarized data would remain intact in the dPoW blockchain but a successful attack against the normal blockchain could rewrite that history. However, in this situation the BTC blockchain can be queried for the notary data.

The dPoW consensus mechanism can be further enhanced by allowing normal nodes to check the notarized data from the BTC blockchain directly. In this case even a normal node would be able to find the correctly notarized mainchain and reject any incoming block that would undo a notarized block.

CONCLUSION

For the initial delayed Proof of Work blockchain to have Bitcoin's security it is required to pay the Bitcoin transaction fees. The transaction fees can get expensive especially as the group signed transactions are on the large side. Rest assured that as long as Bitcoin accepts payments, dPoW blockchain will be secured by Bitcoin.

The amount of effort required to achieve the first dPoW does make one want to leverage it to allow other blockchains to use dPoW indirectly. The last appendix explores ways of simplifying the integration of dPoW into a third party chain. Delayed PoW is able to secure any type of consensus. These third party chains won't have to pay the Bitcoin transaction fees, but only the fees to the initial dPoW chain.

With this system we ensure that the wasted energy is also being used to secure the dPoW blockchain and all the third party chains that choose to employ this consensus mechanism, via transactions to the dPoW blockchain. By attaching these blockchains to Bitcoin, we create an ecosystem where Bitcoin is the center of all currencies that use dPoW, meaning that there is a direct incentive for these cryptocurrencies to actively contribute to the development of the Bitcoin blockchain.

With the new dPoW consensus mechanism everybody wins. Even the most weakest blockchain can get the best security while Bitcoin is rewarded with an even more important role in the overall cryptocurrency ecosystem.

APPENDIX I

BITCOIN GROUP SIGNING THAT IS BEYOND MULTISIGNATURE LIMITS

This is a very Bitcoin specific problem, but since we are using Bitcoin it is best to solve it. We have 64 notary nodes and we want to do a MofN multisig where M is 33. The problem is that Bitcoin doesn't directly support such a large M or N. It is not practical to have the combinatorial set of all possible multisignature with smaller MofN as that will require that they all have funds to spend.

The solution to making a group signature is much simpler than solving some combinatorial multisig issue. Each notary needs to have sufficient utxo funds, preferably in the exact required denomination. Then all notary nodes broadcast to each other their signature for a tx in notary id order for a fully populated 64 input transaction. Since we don't need more than 33 signatures, the top 33 notaries that responded and as ranked by the PoS scores from their slot, create a 33 input transaction for signature. In the event there is dropout of a signer from the first stage to the second, additional rounds are done, including replacements from responders in the first round.

In the worst case scenario, a specific notary hash is unable to be computed, but this has the effect of delaying the dPoW protection as the normal blocks will continue on with the normal PoS protection. One point to note is that it is assumed that a dPoW node (notary and normal) is able to monitor the Bitcoin blockchain and Iguanacore is used to achieve this ability.

APPENDIX II

SUBMISSION OPTIMIZATION OF BITCOIN GROUP TRANSACTIONS

The nature of a blockchain is that given a blockhash B, it is referring to the blockhash for B-1 and assuming that blockhash B is a valid hash, then it means that all prior blockhashes are also valid. This is because each blockhash uses the actual value of the previous blockhash to calculate itself. What this means is that we only need to write the most recent notarized blockhash. By doing this, it is the same as having written all the blockhashes previous to it, just a lot less expensive.

This leads us to envision simply updating to the most recent group signed transaction. The question reduces to when should it be broadcast. Ideally, it is broadcast 30 seconds before the next Bitcoin block, but the early notification for the next block always seems to be buggy, so a different method needs to be devised.

If we send a transaction out and the Bitcoin block is delayed and we get another transaction, that will be inefficient. The way the time to the next block probabilities work, there is really no way to know, so if we pretend there is a Bitcoin block every 10 minutes, then we can offset it by 5 minutes and broadcast every 10 minutes.

Of course, the Bitcoin blocks are variable, so for lack of any apparent way to minimize the time for a notarized hash to appear on the Bitcoin blockchain, we can just say that we broadcast the first new group signed transaction that comes in after more than 5 minutes has passed since the last bitcoin block. Improvements will be devised, but for now this seems adequate to require no more than *3600 * 24 / 600 = 144* transactions per day. If the 5 minutes is changed to say 9 minutes, that will end up reducing the total costs from all the faster than 9 minute blocks.

APPENDIX III

HOW CAN A THIRD PARTY CHAIN UTILIZE DPOW WITHOUT PAYING BITCOIN FEES?

The goal is to make the fewest number of changes to a third party chain to obtain the dPoW protection. If we limit ourselves to Bitcoin compatible third party chains, then the dPoW notaries can become a normal peer so that it obtains the blockhashes through the normal process. With a delay by a specified number of confirmations, the blockhash is written to the dPoW chain.

At this point we have the need for a special signed network message that is sent from the dPoW notary(s) to the third party chain with the detection of the blockhash on the Bitcoin chain. Using this method means there is no need to enable the third party chain from communicating to the dPoW chain or even the Bitcoin chain.

Only one new network message to receive the notary information is needed along with the change to the consensus rule to not reorganize a notarized blockhash.

REFERENCES

Nakamoto, Satoshi (2008): *Bitcoin: A peer-to-peer electronic cash system.*
http://www.bitcoin.org/bitcoin.pdf

Mtchl (2014): *The math of Nxt forging*
https://www.docdroid.net/ahms/forging0-4-1.pdf.html

King Sunny, Nadal Scott (2012): *PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake*
https://peercoin.net/assets/paper/peercoin-paper.pdf

Delegated Proof-of-Stake Consensus
https://bitshares.org/technology/delegated-proof-of-stake-consensus

Miers Ian, Garman Christina, Green Matthew, Rubin Aviel: *Zerocoin: Anonymous Distributed E-Cash from Bitcoin*
https://isi.jhu.edu/~mgreen/ZerocoinOakland.pdf

Ben-Sasson Eli, Chiesa Alessandro, Garman Christina, Green Matthew, Miers Ian, Troer Eran, Virza Madars (2014): *Zerocash: Decentralized Anonymous Payments from Bitcoin*.
http://zerocash-project.org/media/pdf/zerocash-extended-20140518.pdf

Ben-Sasson Eli, Chiesa Alessandro, Green Matthew, Tromer Eran, Virza Madars (2015): *Secure Sampling of Public Parameters for Succinct Zero Knowledge Proofs*
www.diyhpl.us/~bryan/papers2/bitcoin/snarks/Secure%20sampling%20of%20public%20parameters%20for%20succinct%20zero%20knowledge%20proofs.pdf

NXT Community: *NXT Whitepaper*
http://wiki.nxtcrypto.org/wiki/Whitepaper:Nxt

Larimer Daniel, Scott Ned, Zavgorodnev Valentine, Johnson Benjamin, Calfee James, Vandeberg.

Michael (March 2016): *Steem, An incentivized, blockchain-based social media platform.*
https://steem.io/SteemWhitePaper.pdf

BitFury Group (Sep 13, 2015): *Proof of Stake versus Proof of Work White Paper*
http://bitfury.com/content/5-white-papers-research/pos-vs-pow-1.0.2.pdf