

Proof of Less Work

BY CHENG WANG

cheng@alephium.org

www.alephium.org

1 Introduction

This paper is about PoLW to reduce the energy consumption of Nakamoto PoW[1] without sacrificing security. Surprisingly, in practice, the transition from PoW to PoLW could lead to security gain. Note that all the analyses in this paper are based on Nakamoto PoW, but the conclusions should still work for other PoW algorithms with different difficulty adjustment algorithms.

There are three essential parts of PoW which make it the most widely used and tested consensus algorithm for blockchain: decentralized randomness, mining rewards, and high mining costs. Decentralized randomness is really what we want for blockchain, while mining rewards and high mining costs are to keep the network running properly. However, there is no need to consume a huge amount of energy to achieve decentralized randomness because many other coins with lower hashrate than Bitcoin still have stable block time. In this paper, we show that we could design an improved PoW algorithm that attains the three essential parts of PoW with far less energy consumption.

This paper is inspired by the recent and great paper from Itay, Alexander and Ittay [2] which describes several algorithms with reduced external costs in the physical world. It's also inspired by the white paper of Alephium project [3] which introduces dynamic mining rewards with lockup. We propose a family of algorithms to reduce the energy consumption of Nakamoto PoW. In particular, exponential PoLW could in theory reduce the energy consumption by an arbitrary factor.

The key point is as the *IncentivizedInternalExpenses* algorithm in [2] to keep high block generation costs by encouraging the miners to spend funds in a way internal to the network. In PoLW, the miners are able to give up part of the coinbase rewards so as to get a weight (> 1) for the work (block hash) they have done. In some sense, the miners do both actual mining by finding better hashes and virtual mining by burning part of the coinbase rewards. When the work weight gets higher, the actual mining done in the physical world would be less, though the cost in total would not decrease.

From Section 2 to Section 6 we discuss first about our linear PoLW. Then in Section 7 we generalize it to other PoLWs. Discussions about practical applications starts from Section 8.

2 Linear PoLW

Let W be the amount of work (hash target) needed for one block. Let the maximal block reward be 1 coin. The miner could choose to burn $1 - \alpha$ coin ($0 < \alpha \leq 1$) in order to get a weight for its mining work (i.e. block hash). In the end, the miner gets only α coin back as reward. Note that we could design in such a way that the miner gets α coin directly from the coinbase reward, but the upfront mining costs would be lower. We will get back to this later when analyzing the security of PoLW.

If the actual work of the new block is W' , then the weighted work is $\left(1 + \frac{1-\alpha}{\gamma}\right)W'$, where $\gamma \leq 1$ is a system parameter that could vary in different blocks. Each miner could choose a different α to ample it's mining work.

W	amount of work needed for a regularly mined block
W'	actual work the miner produces
$\alpha \leq 1$	actual block rewards. maximal is 1 coin
β	cost for regular block mining
$\gamma \leq 1$	system parameter for weight calculation

Table 1.

3 Mining Strategy

The question arises that which α should a miner use to maximize its return. Let's say a miner M has x coin to use for mining the block. Let β coin be the resource cost of W work for the regular Nakamoto mining. Then in the equilibrium case, the probability of M getting the block is

$$p_M = \frac{x\{1 + (1 - \alpha_M)/\gamma\}}{\beta}$$

The expected return is

$$p_M \alpha_M - x = \left(1 + \frac{1 - \alpha_M}{\gamma}\right) \alpha_M \frac{x}{\beta} - x$$

The maxmized return is

$$\max(p_M \alpha_M - x) = \max\left(\left(1 + \frac{1 - \alpha_M}{\gamma}\right) \alpha_M\right) \frac{x}{\beta} - x$$

We could easily deduce that

$$\alpha_M = \frac{1 + \gamma}{2}$$

Therefore, the long term strategy \mathcal{S} for a miner is to set α to be $\frac{1+\gamma}{2}$ to mine new blocks. However, in short term, the miner could adjust its α for better expected return. We show that strategy \mathcal{S} is an equilibrium strategy by an ideal analysis in the next section.

Note that even if the Miner M 's mining algorithm or machine is more efficient than the others, the optimal value of α_M is still $\frac{1+\gamma}{2}$.

4 Equilibrium Strategy

Let's assume that all the miners work together and use the same α to try to maximize the return of mining. In order to make the weighted work reach the target work W , the miners need to cost $\frac{\beta}{1 + (1 - \alpha)/\gamma}$ coin in the physical world. The actual return of the miners is

$$R(\alpha) = \alpha - \frac{\beta}{1 + (1 - \alpha)/\gamma}$$

With some calculation, we get the following result

$$\max R(\alpha) = \begin{cases} 1 + \gamma - 2\sqrt{\beta\gamma} & \text{when } \gamma \leq \beta \leq \frac{(1+\gamma)^2}{4\gamma}, \text{ with } \alpha = 1 + \gamma - \sqrt{\beta\gamma} \\ 1 - \beta & \text{when } \beta < \gamma, \text{ with } \alpha = 1 \\ 0 & \text{when } \beta > \frac{(1+\gamma)^2}{4\gamma}, \text{ donot mine} \end{cases}$$

We see here that when the mining cost is very low, the miners will set $\alpha = 1$ and it degenerates to the classic Nakamoto mining. However, as the mining cost goes up, the miners will have to set $\alpha = 1 + \gamma - \sqrt{\beta\gamma}$ to maximize its return. Therefore, we see that in the non-equilibrium case, the miners could set α to be different from $\frac{1+\gamma}{2}$ for better mining return.

Equilibrium Case. In equilibrium, $\max R(\alpha)$ should be equal to 0. In such case, we have $\beta > \gamma$ and $1 + \gamma - 2\sqrt{\beta\gamma} = 0$. Therefore

$$\begin{aligned} \beta &= \frac{(1+\gamma)^2}{4\gamma} \\ \alpha &= 1 + \gamma - \sqrt{\beta\gamma} = \frac{1+\gamma}{2} \end{aligned}$$

We have shown that α will be $\frac{1+\gamma}{2}$ for all the miners in the equilibrium case. Therefore, \mathcal{S} is the equilibrium strategy in the case where mining profit is negligible to be 0.

5 Security Analysis

We only compare our algorithm to the classic PoW algorithm to see the security differences in a theoretical setup without taking care of external facts from the physical world. There will be discussion about practical implications of PoLW in later sections.

We just need to consider two cases here: a) successful attacks, b) unsuccessful attacks.

Successful attacks. When an attacker successfully attack the network, it gets back both the mining revenue and external revenue like double spending. In both PoW and PoLW, the attacker needs to first invest 1 coin to mine a new block and then get the reward back in the equilibrium case. The mining revenue is close to 0 when the mining market gets mature for both PoW and PoLW. Therefore, the return of the attacker would be the same for both algorithms, so as the security.

Unsuccessful attacks. When an attacker failed in attacking the network, the total costs of it would be the external mining cost and optionally burnt coins. If we slash the burnt coins in the side forks using merkle proofs for example, then the costs of a failed attack would be 1 coin for each block for both PoW and PoLW. In this case, the security is the same for both algorithms. If we choose not to slash the burnt coins, then the security of PoLW would be lower, but the gain is that when a block from an honest miner got orphaned, the lost of the honest miner is also lowered.

We ignore the other metrics analysis similar to that of paper [2], which should be intuitive.

6 Energy Consumption

The actual work done by the miner is $\frac{\beta}{1+(1-\alpha)/\gamma}$ coin. In equilibrium where $\beta = \frac{(1+\gamma)^2}{4\gamma}$, $\alpha = \frac{1+\gamma}{2}$, the work amount is equal to the mining reward $\frac{1+\gamma}{2}$, less than 1 coin. Therefore, it costs less energy compared to Nakamoto PoW.

7 Generalization & Exponential PoLW

Let's assume now that the weights of work is calculated by $1 + f(1 - \alpha)$. In linear PoLW, $f = \frac{1-\alpha}{\gamma}$. Similarly, let's first find out the optimal α for a Miner M in long term.

$$p_M = \frac{x\{1 + f(1 - \alpha_M)\}}{\beta}$$

The expected return is

$$p_M \alpha_M - x = \{1 + f(1 - \alpha_M)\} \alpha_M \frac{x}{\beta} - x$$

The maximized return is

$$\max(p_M \alpha_M - x) = \max(\{1 + f(1 - \alpha_M)\} \alpha_M) \frac{x}{\beta} - x$$

The optimal mining parameter would be to satisfy the following equation:

$$1 - f'(1 - \alpha_M) \alpha_M + f(1 - \alpha_M) = 0$$

We want to choose good function f such that $0 < \alpha_M < 1$. As in the equilibrium case, the energy cost would be close to α_M , we also want the optimal α_M to be small if possible. There might be many ways to choose such kind of functions. We focus on this following simple case.

Exponential PoLW. Here we take $f(1-\alpha) = e^{\gamma(1-\alpha)} - 1$ ($\gamma \geq 1$), i.e. the weight of work becomes $e^{\gamma(1-\alpha)}$. In this case, we have

$$\max(\{1 + f(1-\alpha_M)\}\alpha_M) = \max(\alpha_M e^{\gamma(1-\alpha_M)})$$

With simple calculation, we know that the optimal $\alpha_M = \frac{1}{\gamma}$.

In the following, we apply the similar analysis as previous sections, without repeating all the details.

Equilibrium. The actual return of miners is

$$R(\alpha) = \alpha - \beta e^{-\gamma(1-\alpha)}$$

$$R'(\alpha) = 1 - \beta \gamma e^{-\gamma(1-\alpha)}$$

$$\max R(\alpha) = \begin{cases} 1 - \frac{\lg(\beta\gamma)}{\gamma} - \frac{1}{\gamma} & \text{when } \beta \geq \frac{1}{\gamma}, \text{ with } \alpha = 1 - \frac{\lg(\beta\gamma)}{\gamma} \\ 1 - \beta & \text{when } \beta < \frac{1}{\gamma}, \text{ with } \alpha = 1 \end{cases}$$

In equilibrium, we have $1 - \frac{\lg(\beta\gamma)}{\gamma} - \frac{1}{\gamma} = 0$ and $\alpha = 1 - \frac{\lg(\beta\gamma)}{\gamma}$, therefore

$$\alpha = \frac{1}{\gamma}$$

Therefore, miners using $\alpha = \frac{1}{\gamma}$ is an equilibrium mining strategy.

Security. The same security analysis as in Section 5 applies here.

Energy consumption. In equilibrium, the actual mining cost (i.e. energy consumption mostly) is $\frac{1}{\gamma}$. If we choose γ to be large enough, the energy consumption could be close to 0.

8 Parameter Selection

In linear PoLW, we could choose γ close to zero so that the energy cost could be reduced to close to $\frac{1}{2}$. However, when γ is small, the weight $1 + \frac{(1-\alpha)}{\gamma}$ to the actual mining work would be huge, this will make double spending more feasible with less actual mining work.

Same in exponential PoLW, we could choose γ large so that the energy cost could be reduced to close to 0. However, when γ is big, the weight $e^{\gamma(1-\alpha)}$ to the actual mining work would be huge, this will make double spending more feasible as well.

In practice, we should set the lower bound of α to be the optimal α based on γ . When mining is in equilibrium, the attackers could not get higher weights than the other miners. We should also adjust γ when the blockchain evolves, so that the weights are lower from the beginning, but getting higher eventually. For example, we could adjust γ based on the currently actual hashrate without weights.

Now let's analyze concrete cases. Let $\pi(\leq 1)$ coin be the actual mining cost of the equilibrium mining state of each cases in the following discussion. We will discuss both linear PoLW and exponential PoLW.

Linear PoLW. Based on our previous analysis, we have the following equations in equilibrium:

$$\begin{aligned} \beta(\geq \gamma) &= \pi \left(1 + \frac{1-\alpha}{\gamma} \right) \\ \alpha &= 1 + \gamma - \sqrt{\beta\gamma} \end{aligned}$$

With simple calculation, we got

$$\gamma + 1 - \alpha = \pi$$

The return of miners is $R = \alpha - \pi$. Let assume that $R = p\pi$, i.e. miners will make p profits in equilibrium (e.g. p could be 0 or 10%). Taking this and the formulas for α and β , we could get

$$\begin{aligned}\alpha &= (1+p)\pi \\ \beta &= \pi^2/\gamma \\ \gamma &= (2+p)\pi - 1 \\ \pi &\leq \frac{1}{1+p}\end{aligned}$$

We could use these to calculate γ . For example, in the ideal case that miners make 0 profits with $p=0$, if we want the external costs in equilibrium to be 0.6 coin, then we would like γ to be 0.2. In the case that miners make 10% profits with $p=0.1$, if we want the external costs in equilibrium to be 0.6, then we would like γ to be 0.26.

Exponential PoLW. Based on our previous analysis, we have the following in equilibrium state:

$$\begin{aligned}\beta\left(\geq \frac{1}{\gamma}\right) &= \pi e^{\gamma(1-\alpha)} \\ \alpha &= 1 - \frac{\lg(\beta\gamma)}{\gamma}\end{aligned}$$

Same as before, let the return of miners is $R = \alpha - \pi = p\pi$. We could get

$$\begin{aligned}\alpha &= (1+p)\pi \\ \beta &= \pi e^{\gamma(1-\alpha)} \\ \gamma &= \frac{1}{\pi} \\ \pi &\leq \frac{1}{1+p}\end{aligned}$$

These are the formulas to compute γ . For example, if $p=0$ and we want the external cost in equilibrium to be 0.25, then we would like γ to be 4. In the case that miners make 10% profits with $p=0.1$, if we want the external costs in equilibrium to be 0.25, then we would like γ to be 4.

Bitcoin/Ethereum/New PoW Systems. Once the project figures out which π is proper for its network in long term, we could use formulas above to compute the system parameter γ . However, the transition from PoW to PoLW must be smooth in order to avoid too much dynamics for miners. We could set an evolving lower bound for α . This lower bound starts from 1, and then gradually decreases to $(1+p)\pi$. The network could update the lower bound based on the actual mining difficulty instead of the weighted mining difficulty. Note that one option could be that the target π is very close to 0 using exponential PoLW, but the lower bound of α converges slowly to 0. In this case, we expect the coin value to raise high in long term, so that the actual value of external cost still increases, though the percentage of external cost in the total cost will get low.

Security Gains. In equilibrium, PoLW and PoW have the same security property in terms of cost of block generation. In practice, since mining is still very profitable, we would expect costs of block generation to be higher once the transition from PoW to PoLW happens. The reason is that PoLW could make the mining game converge to the final equilibrium state quicker than PoW due to the mix of external costs and internal costs.

9 Implementation considerations.

The only thing to change in the algorithm part is to use weighted work target instead of classic work target. Therefore, there is a negligible implementation overhead. However, in order to validate hash difficulty using solely block header, the value of declared coinbase reward would be included in the header. It could be just 1-2 bytes if we discretize γ properly.

It's also possible to transition from PoW to PoLW using only soft-forking by introducing burning transactions in addition to coinbase transactions. We will write down the details later.

10 Discussion.

It would be good to analysis PoLW in more complicated models. The author thanks all the feedback received from friends and anonymous people.

Bibliography

- [1] Satoshi Nakamoto et al. Bitcoin: a peer-to-peer electronic cash system. 2008.
- [2] Itay Tsabary, Alexander Spiegelman, and Ittay Eyal. Just enough security: reducing proof-of-work ecological footprint. *ArXiv preprint arXiv:1911.04124*, 2019.
- [3] Cheng Wang. Alephium: a scalable cryptocurrency system based on blockflow. <https://github.com/alephium/research/raw/master/alephium.pdf>, 2018.