2018 42nd IEEE International Conference on Computer Software & Applications

Proof of Contribution: A Modification of Proof of Work to Increase Mining Efficiency

Tengfei Xue*†, Yuyu Yuan*†, Zahir Ahmed*†, Krishna Moniz*†, Ganyuan Cao‡ and Cong Wang*†
*School of Software Engineering, Beijing University of Posts and Telecommunications
†Key Laboratory of Trustworthy Distributed Computing and Service(BUPT),
Ministry of Education Beijing, China
xtf@bupt.edu.cn, yuanyuyu@bupt.edu.cn, xahiru@gmail.com,
krishnamoniz@live.nl wangc@bupt.edu.cn
‡Arizona State University, Tempe, USA
Ganyuan.Caogb@asu.edu

***Abstract*—Proof-of-Work based blockchain wastes an enormous amount of electricity and requires expensive mining equipment. This phenomenon is getting worse and worse. We propose a new consensus protocol for cryptocurrency built on top of Bitcoin protocol, which combines Proof-of-Work component with our new algorithm Proof-of-Contribution(PoC). PoC algorithm reduces the energy consumption for Bitcoin mining by rewarding the calculation difficulty of a cryptographic puzzle. Further, when miners no longer get block rewards, they receive difficulty rewards along with the regular mining fee which motivates mining. We explored different attacking scenarios and show that the network resilient to these attacks is comparable to Poof-of-Work(PoW). Finally, we designed two experiments to simulate mining and detect the changes of difficulty to prove PoC offering a new cryptocurrency algorithm which is energy-efficient.**

***Keywords*-Proof-of-Contribution, Bitcoin, Energy-efficient, Proof-of-Work**

I. Introduction

Bitcoin is a decentralized cryptocurrency created in 2009[1], which recently gained great attention from the economic community as well as the academia. PoW was introduced as a consensus algorithm to ensure the security of the Peer-to-Peer (P2P) blockchain. However, Bitcoin's implementation of the PoW algorithm results in a great amount of wasted electrical energy. In the PoW algorithm, miners compete to solve a cryptographic puzzle with a prize, which is the right to add a block to the blockchain. The cryptographic puzzle is, basically, a search operation using the SHA-256 hashing algorithm to find a target hash, which encodes the difficulty level. The process requires an enormous amount of computations. The difficulty level is adjusted in every 2016 blocks to keep the rate of creating blocks approximately a constant. The increasing difficulty level of the Bitcoin network has resulted in the probability of a single miner mining a block becoming lower, and consequently the appearance of large mining pools. The miners are rewarded according to their hash contribution to the pool. A large number of miners are centralized in pools, where the cost of electricity and equipment are cheap. In addition, as the network grows, the difficulty of the cryptographic puzzle increases, which increases the energy waste.

Due to the great number of computations, special computing machines are designed to make computation more efficient. Several studies [2] [3] have proposed more efficient custom and non-custom hardware for Bitcoin mining. High-performance graphical processing units (GPUs) and Application Specific Integrated Circuit (ASIC) processors are dedicated chips specialized for Bitcoin mining, are used to manufacture the mining machines. According to the current market rates, the cost of an ASIC mining machine varies from 3000 USD to 12000 USD ($Bitmain.com$). These mining machines compute the required nonce fast if compared to PCs. However, the increased use of these mining machines stimulates the difficulty of mining over the time as shown in $Bitcoinity.org$[4].

In addition to improving the performance of mining hardware, several alternative consensus algorithms have been introduced to address the high energy consuming problem caused by PoW. These algorithms can be divided into two types : (1) Proof-of-Stake (PoS) based algorithms - which do not require a high amount of computational power. (2) Different variations PoW with slight modification as well as combinations of Pow and PoS.

In PoS based algorithms, the distributed consensus is achieved by various combinations of random selection with ”stake”. The voting power of a participant on the validity of a block depends on their stake. The form of stakes varies in different implementations. Peercoin[5] uses ”coinage” as the stake to mint new coins. Coinage is derived from multiplying the number of coins with the number of days it has been held. Once the stake of the coins is used for minting a new coin, their coinage will be reset to zero. The maximum age of staking is set at 90 days to prevent extremely old coins dominating the minting. Tendermint [6] uses the solution of Byzantine General Problem to achieve the consensus. According to the Tendermint protocol, validators are the users who have locked their coins through a bond transaction. Their voting power is related to the number of coins they locked. These algorithms attempt to achieve consensus without a significant amount of computation power. However, there are several new issues arise from PoS based algorithms. For example, the "Nothing at Stake" problem happens when a user votes on all forks to get the maximum benefit and it is not a completely decentralized solution.

In most of the second type, the algorithm either extends PoW by adding a new scheme on top of PoW or combines a variation of PoS. Proof-of-Activity (PoA) [7] is an algorithm that combines PoW and PoS. Instead of generating the whole block, in PoA, hash power is only used to generate the block-header, thus, the process requires few computations. Peercoin[5] implements a hybrid of PoW and PoS. In the early stage of the network growth, Peercoin uses PoW to generate most of the blocks. As the difficulty increases and the mining reward decreases, PoS is used for minting new coins. This transition will result in eventually phasing out PoW. The difficulty is adjusted based on the coinage. The older the coin, the easier to meet the target hash and generate a new block. This type of algorithm aims to alleviate the high energy consumption problem caused by PoW. However, the combined approach is not efficient as expected. For instance, as of January 2014, Peercoin usage equaled to approximately 30% of Bitcoin's energy consumption. In addition, as mining reward of Bitcoin halved in July 2016, the miners moved from Bitcoin to Peercoin since both cryptocurrencies employ the same hashing algorithm (SHA256). The Peercoin network's hashing power has increased 13 times as a result.

In order to address the high energy consumption issue, we propose a novel consensus algorithm, called Proof of Contribution (PoC). Similar to Peercoins' consensus algorithm, our algorithm combines PoW and PoS. Energy efficiency is achieved by the PoS component of the algorithm. In contrast to the coinage in Peercoin, the stake is based on the honesty of the miner. It is calculated as the number of times the miner adds a valid block to the blockchain, represented as *Success Times*. The success times coefficient determines the miners mining difficulty. Each miner has a different difficulty level. The higher the score of miners' success times, the easier to mine a block. Instead of solely relying on hashing power for a higher probability of mining blocks, PoC provides another way to increase the miners' likelihood of finding a block by integrating with success times. Miners can verify the success times and difficulty of other miners by viewing the transactions on the blockchain. A blacklist is maintained by miners to deter misbehavior and make the network resilient to attacks. As a punishment for misbehavior, all the difficulty reward of the miner are withdrawn once its address is blacklisted. PoC motivates honest behavior while discouraging misbehavior. However, important aspects of blacklist are not discussed such as, how to define a misbehavior and how to detect attacks.

The main contribution of this work is that we proposed the novel consensus algorithm, which alleviates the high energy consumption and improves mining efficiency of the Bitcoin network. This paper is organized as follows: In Section 2, we discuss the issues associated with the Bitcoin mining difficulty. In this section, we also specify several well-known malicious mining behaviors. In Section 3, we introduce the characteristics and terms associated with the PoC algorithm. Then we describe the concept of PoC and how PoW is modified to change the mining difficulty. In Section 4, we explore different attack scenarios and evaluate the performance of the new algorithm compared to the original PoW algorithm. In section 5, we simulate PoW and PoC algorithms and analyze the energy consumption for Bitcoin mining. In section 6, we conclude the paper by briefly summarizing our work and indicating the direction of future work.

II. Preliminaries

A. Bitcoin mining

In the original PoW algorithm, miners try their best to solve a cryptographic puzzle by calculating the hash of a block header twice with the SHA-256 function. Solving this puzzle is referred to as Bitcoin mining. Each miner uses a different number called a nonce as a random part of block header to yield a hash. The block header contains a field named "Bits" that store the mining difficulty, which ensures the validity as well as the generation of a block approximately in every 10 minutes. The Bit is a 32-bit integer that compresses the actual hexadecimal target. It consists of two parts: as in the formula 1, the first two digits of the hexadecimal are used as exponent and the remaining bits are the coefficient denoted as C. All users in the Bitcoin network use a global difficulty denoted as D. The legitimate block must have to get a hash value lower than the target which we can get from D by applying the formula 1.

$$Target = C * 2^{(8*(exponent-3))} \tag{1}$$

For example, in the block number 421133 the value of difficulty is 402990845 and in hexadecimal is 0x180526FD. So the coefficient is 0x0526FD and the exponent is 0x18. According to formula 1, we get the result is:

$$Target = 337661 * 2^{168} = 0x000...0000526FD0...0 \tag{2}$$

The value of difficulty represents how hard the block to be generated, which compared to the first block created by Satoshi Nakamoto. In order to keep the generation of blocks to maintain the same transaction time, the difficulty is automatically adjusted according to the total hash rate of all the participating miners in the previous 2016 blocks.

$$D_{new} = D_{old} * (\frac{Actual\ Time\ of\ Last\ 2016\ Blocks}{1209600\ seconds}) \tag{3}$$

When the difficulty target is at the maximum value 0x1D00FFFF, the minimum difficulty is 1. The maximum difficulty does not effectively exist, because when target equals 0 the resulting difficulty would be infinite. However, we can roughly estimate that the maximum difficulty is 2^{224} when the target is set to 1. In the PoC algorithm, we use success times to reduce the difficulty.

B. Malicious behaviors

Double spending attack and selfish mining attack are two well-known malicious behaviors in Bitcoin. The verification scheme in the PoW algorithm is specially designed to counter double spending[1] [8], but research shows that the 10-minute transaction confirmation period does not meet the need of fast trading while the shorter time increases the risk.

Double spending happens when a malicious client tries to reuse the same coins that have already been spent. In a normal transaction sender and receiver generate a new key pair and sign the transaction using their keys. However, in the double spending attack, the malicious attacker (sender) tries to generate a parallel chain, in private, containing an alternative version the transaction which has already been sent to another receiver (vendor). The vendor waits until the transaction is added to a block and z blocks have been linked after it. The attacker's potential progress will be a Poisson Distribution.

$$\lambda = z\frac{q}{p} \tag{4}$$

where p denotes the probability that an honest miner finds the next block, q stands for the probability that the attacker finds the next block. The attacker uses a "helper address" to regain the coins, for which the vendor has already provided a service. As advocated in [9], Bitcoin developers proposed two detection strategies, namely: using a listening period and inserting observer, to help client users find double spending behavior.

The selfish mining attack was first mentioned and discussed on Bitcoin forum[10] and then Ittay Eyal and Emin Gun Sirer of Cornell[11] improved the analysis. It is a way that a dishonest miner could finesse the protocol and win more blocks than their percentage of the overall hash rate. They analyzed the benefits of selfish mining in eight different scenarios. In order to maximize their reward, it is advantageous for a miner to reveal a new block as soon as it is found. Selfish miners, by contrast, hide the block on a private chain and publish it only when there is no more lead. According to [11], the power of selfish miner is denoted by α and γ is the proportion of the honest node's hashing power to work on the selfish miner's chain during the competition.

$$\frac{1-\gamma}{3-2\gamma} < \alpha < \frac{1}{2} \tag{5}$$

According to formula 5, if $\gamma = 1$, every miner propagates all the branch of the same length he received which in turn yields a threshold of $\alpha = \frac{1}{4}$. If $\gamma = 0$ or $\gamma = 1$, the threshold for success of block withholding behavior is $\alpha \geq \frac{1}{3}$ or $\alpha \geq 0$, separately. However, currently, there is no effective method for detecting selfish behavior. Thus, there is no definite proof that selfish mining is taking place or not, in the current Bitcoin network. The number of abandoned (orphaned) blocks and timing of successive blocks are two distinct network signatures of selfish mining. The problem with the orphaned block is that it is difficult to get a definite count. In particular, Matt Springer [12] has finished a fascinating timing gap analysis but the results were just a statically fluke. Since it is too easy to hide the true owner of a block, the ownership of successive blocks is not a good sign of selfish mining. However, a PoC implementation would encourage miners as well as mining pools to reuse the addresses, in order to receive the difficulty reward. Thus, ownership of the miners can be tracked to allow the network to be more trustworthy. Importantly, although miners anonymity becomes more transparent, the anonymity of normal users transacting on blockchain remain pseudo-anonymous as in the PoW implementation.

III. Protocol

The PoC is an extension of the Bitcoin protocol. Each miner in PoC is required to verify the difficulty of the miner who releases a new block using the blacklist since each miner has a different difficulty. This extra work is insignificant compared to the energy consumption on hashing.

A. Success Times

The success times (denoted by c) is a crucial concept in PoC. It is the number of times the miner has added a valid block to the chain. Whenever a valid new block is added by a miner, the miner's success times increases by 1. The value of success times can be obtained from the public ledger by viewing coinbase transaction for each miner in the network, it is similar to getting the balance of an account. A coinbase transaction is a unique type of Bitcoin transaction (as shown in Figure 1) that can only be created by a miner who found a new block. Coinbase transaction has no input and the output is the sum of block rewards (which halves every 21000 blocks and is currently at 12.5 BTC) and all the transaction fees included in the block. Miners choose to use one or more addresses to receive the rewards[13], so that each address's score $c \in R^+$.

B. Proof-of-Contribution

PoC extends PoW by introducing success times, which is a mechanism for adjusting miner's mining difficulty. PoC maintains the same difficulty adjustment mechanism of Proof-of-Work. If the network difficulty decreases, the protocol will decrease the difficulty to ensure the same

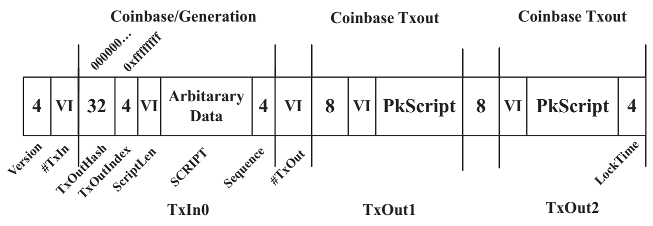


Figure 1. Bitcoin transaction structure [13].

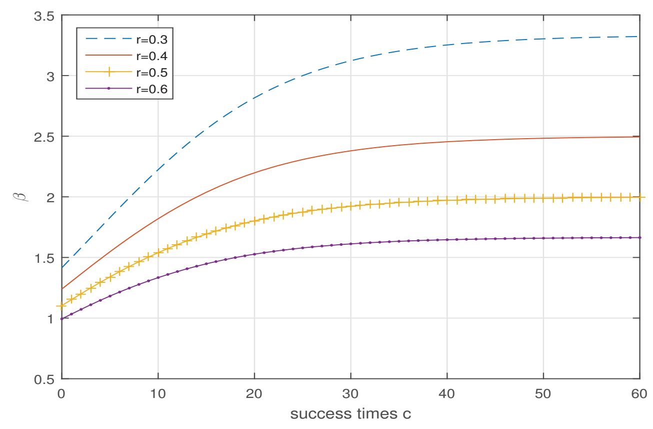


Figure 2. For a given r, the coefficient β with the change of success times c.

average transaction time. Because the major change of PoC is adjusting miner's mining difficulty, it would be easy to implement the modification in the real Bitcoin network. As different miners have different success times, a difficulty factor β is introduced to adjust the miner's difficulty according to miners' success times c.

$$\beta = \begin{cases} \frac{1}{r+e^{-(c-\mu)/\gamma}} & c \geq 1 \\ 1 & c = 0 \end{cases} \tag{6}$$

The equation 6, r determines the upper bound convergence of the function. By adjusting r, the maximum reward for each miner can be changed. The maximum difficulty reward is the reciprocal of r. As shown in Figure 2, μ affects the position of the mean. We set μ equal to -9 to ensure that β is greater than 1. The value of γ determines the growth rate near the mean; the smaller the γ is, the faster the growth speed is. We set $\gamma = 10$ to ensure that the miners can get their roughly maximum reward after mining 50 blocks. The equation is based on the logistic scale in order to control the increment and to ensure that each bonus increases while the increment decreases.

In the protocol, miners with the global difficulty D mine for a period of time t. We combine the D and the difficulty factor β to get the current difficulty D^c:

$$D^c = D \cdot \beta^{-1} \tag{7}$$

The equation is monotonically decreasing, which guarantees that the difficulty will continue to decrease with the accumulation of c. As β increases, D^c will be smaller than the current global difficulty D. The basic work of the miners remain unchanged, including the collecting, verifying, packaging of transactions into the block, and calculating the hash according to the different difficulty. The miners with high success times get the benefit of an easier cryptographic puzzle. This does not mean a miner with low success times cannot increase their likelihood of finding a block. Like in PoW, they can acquire more hashing power to increase their chances and compete with miners who have lower difficulties. In the other word, the maximum gains from seniority are capped by adjusting the β of the protocol. This should prohibit possible consolidation by a single miner with maximum seniority and give new miners a chance to eventually catch up to older miners. By decreasing the difficulty, the amount of computation required to achieve the consensus of the network is reduced. There are various attacks on the Bitcoin network, we focus on how to store blacklist in this paper.

C. Blacklist

We propose to embed a blacklist into blockchain to record addresses involving in malicious behaviors and it will be referenced when validating the blocks. It is the miners' responsibility to detect attacks and malicious behaviors in the network. Once an address is detected participating in an attack, it will be added to the blacklist by the miner. Whenever a block is found by a miner whose address has been blacklisted, the other miners ignore that block. The purpose of the blacklist is to prevent malicious addresses from receiving the difficulty reward, thereby reducing incentives for malicious behavior. Several kinds of research have focused on studying the malicious behavior[14][15], however much work has yet to be done on how to store blacklists and how to verify whether a transaction should be blacklisted. There are various attacks on the Bitcoin network, we focus on how to store blacklist in this paper.

The blacklist is composed of public and private information. The snippets of the public blacklist are stored in each block, and all the blocks together form the full version of the public blacklist. There are two places in the blockchain where data can be stored instead of funds. One of them is the Arbitrary Data field in the Coinbase transaction shown in Figure 1, but the data field only can be up to 180 bytes in length. The second way is to record data in transaction outputs. The data can be embedded to the output with OP_RETURN opcode. This transaction will be ignored by normal users. Although each output contains up to 80 bytes, we can use multiple OP_RETURN outputs in a single transaction. The most commonly used Bitcoin address format is 25 Bytes, each opcode output can store three addresses, as shown in Figure 3. It is easy for users or miners

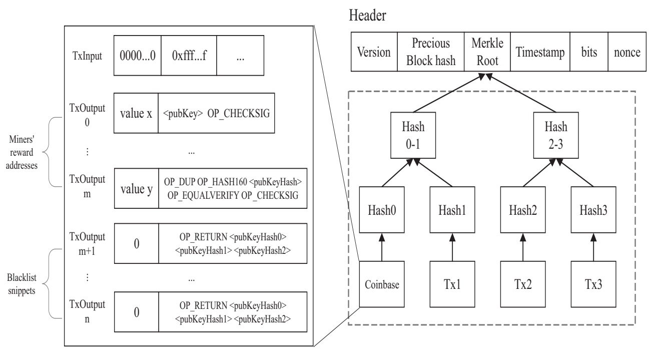
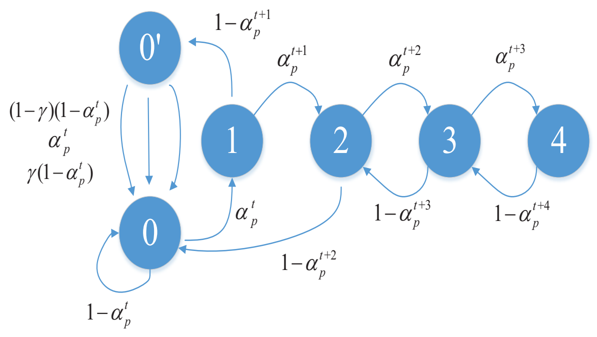


Figure 3. The extend structure of Bitcoin with blacklist.



$$\begin{tikzcd}
0' \arrow[d, bend left] \arrow[d] \arrow[d, bend right] & 1 \arrow[l, "1-\alpha_p^{t+1}"'] \arrow[r, bend left, "\alpha_p^{t+1}"] & 2 \arrow[r, bend left, "\alpha_p^{t+2}"] \arrow[dll, bend left, "1-\alpha_p^{t+2}"] & 3 \arrow[r, bend left, "\alpha_p^{t+3}"] \arrow[l, bend left, "1-\alpha_p^{t+3}"] & 4 \arrow[l, bend left, "1-\alpha_p^{t+4}"] \\
0 \arrow[loop left, "1-\alpha_p^{t}"] \arrow[ur, bend right, "\alpha_p^{t}"'] & & & &
\end{tikzcd}$$

Figure 4. The state machine with mutative transition frequencies based on Proof-of-Contribution. The selfish pool starts with t times where $t \geqslant 0$.

to parse the public blacklist. Another method to embed data into blockchain is to store the hash of a document in a transaction, also known as Proof-of-Existence[16]. Since it proves that an instance of the file exists at some point time, it can be used as proof of ownership without relying on a central certification authority.

In the Bitcoin network, users transfer coins via transactions, which is a data structure that contains inputs and outputs. The outputs record the coins that the users received. If the current owner creates a new transaction and tries to use coins, he must specify where the coins come from. Several academic papers [17][18] proposed two heuristics (Multi-input Transactions and Shadow Address) for grouping addresses controlled by the same user. Based on the two heuristics, we add the corresponding detection code for the client. Using these heuristics, each miner can generate a local blacklist (the private blacklist) based on initial information from the public blacklist. Every miner using the same code to parse public blacklist will get the same private blacklist. If the miner has no further information about the addresses in the public blacklist, then the private and the public blacklists are same.

IV. Analysis of Proof-of-Contribution under Attack

The PoC protocol reduces the miner's relative mining difficulty over a period of time. Although the absolute value of the difficulty becomes higher as the equipment improves and the calculation increases, it is still enough to motivate the miners to use a fixed address for rewards. This approach does not seriously affect user's privacy while providing additional benefits since mining addresses are usually used only for mining. A miner can either choose to mine alone or to join a pool. For pool mining, miners have to provide their computing power to the pool and receive a more stable reward[19]. In most cases, miners use a fixed address to get the rewards. The address based difficulty reward mechanism would discourage miners from joining the pools unless the pools implement a fair mechanism to distribute the difficulty rewards among its members. Otherwise, the miner's fair share of difficulty reward would be retained by the pool, only reducing the overall difficulty of the pool. Another benefit is the increased cost of malicious behavior. Mining can be considered as a random event since the calculated hash is independent of each other. Block finding when solo-mining with a constant hash rate h is a Poisson process with $\frac{h}{2^{32}D^c}$ as the rate parameter. If mining for time t results in $\frac{ht}{2^{32}D^c}$ blocks on average. We can say further that number of blocks found follows the Poisson distribution with $\lambda = \frac{ht}{2^{32}D^c}$, for which the miner is rewarded with Bitcoin in an amount we denote as B. The expected payout for a miner is $\lambda = \frac{htB}{2^{32}D^c}$.

According to the PoC protocol, the cost of launching an attack depends on how many success times the miner has. Mining under the PoC protocol becomes unprofitable once the expectation of cost is equal to or exceeds $\frac{(\beta-1)htB}{2^{32}D^c}$ based on the distribution where $\beta \geq 1$.

A. Selfish Mining Attack

Selfish mining is an attack on the integrity of the Bitcoin network. As discussed in the preliminary section, it happens when one miner, or a mining pool, does not release and distribute a valid solution to the rest of the network. It results in unfair competition for honest nodes.

Although there is no reliable method for effectively detecting selfish behavior, there are three indicators of selfish activity and two of them will be very effective in PoC protocol. It shows that the network resilient to these attacks is comparable to PoW. They are a number of abandoned (orphaned) blocks, the timing of successive blocks, and the ownership of successive blocks. The problem with measuring the number of orphaned blocks is that it is difficult to get a definite count. Since it's quite easy to hide the true owner of a block by changing the mining address, so the ownership of successive blocks is not a good sign of selfish mining in Bitcoin.

In PoC, it is assumed that the honest miners and the pools will prefer using the same address to receive difficulty rewards in the system. Also, we assume that the selfish pool has mining power of α_p^0 on state zero without success times. When the selfish pool creates blocks successfully with same address and gets rewards, the mining power will become $\alpha_p^c = \frac{\beta_p h_p}{\sum_{i \epsilon S} \beta_i h_i}$. This is obviously a monotonically increasing function. It would be very difficult for an attacker to, simultaneously, cloak the identity and hide timings of successive block creation.

We detail the cost of each event below:

1) If the selfish pool initiates an attack using a different address to avoid the detection as in Bitcoin. If the attack fails, the more blocks they create, the more computing power they will lose. The total cost is $\frac{(\beta_p - 1 - r) h_p t B}{2^{32} D^c}$ where the success times is $c \geqslant 1$. In the current implementation of Bitcoin, this cost is relatively low compared to PoC.

2) In the second scenario, if the selfish pool success in creating an attack using different addresses. Although each address will receive a certain difficulty rewards, as long as there is one of the addresses is judged by joining the selfish mining all the incentives of the addresses' owner will be invalid. There will heavy penalties in the future.
3) In the third scenario, where selfish pool succeeds in finding blocks with the same address. The ownership will be a better indicator ideally coupled with the timing of successive blocks. One attacker is found misbehaving, miners could reject the blocks from attacker's addresses which are subsequently recorded on the blacklist. The paper[11] details eight situations of selfish mining and calculates the revenue of the pool from state probabilities and transition frequencies. The state machine model of the proposed system is shown in Figure 4. The states of the system represent the lead of the selfish pool. State 2 means that there is a lead of two unpublished blocks in selfish miners' private chain. Although the new α makes the attack cheaper to perform, releasing blocks with the same address means it's easier to detect the attack behavior. The attacker faces the compound problem of a loss of rewards due to failed attempts and an increased likelihood of detection of selfish behavior when moving from state 2 to state 0. Miners can reject blocks created by same address with creation time under a certain threshold and blacklist the address.

B. Double Spending Attack

The double spending attack refers to the case where a malicious client can redeem and re-use a coin which has already been spent [1] There are multiple scenarios of a double spending attack. One scenario is that the attacker could include the malicious transaction in his parallel chain and generate a block faster than the current chain to take control of the chain and launch double spending with his computing power. For the first scenario, we conclude that the reduction of puzzle difficulty will not increase the possibility of launching a double spend attack since the change of λ, which is the attacker's potential progress, is small. Another scenario is that the attacker can send the same coin to an address that is affiliated with the attacker to launch the attack. Current detections of double spending attack are mainly by using a listening period and introducing an observer [8].

As a detection mechanism for double spending, [8] proposed to use a listening period, where the service provider waits for few seconds before delivering the service to the sender. During this period, the vendor checks every transaction to verify that the received coins have not been used in another transaction. Introducing an observer is usually affiliated with using a listening period. Observers would relay all transaction it receives to the vendor. Since it takes few seconds for every transaction to propagate to all the nodes on the network, it is highly likely that both vendor and its observers would receive the same transaction from the attacker in this period. [8] analyzes the prospects of these two detection strategies and proposed a new efficient way based on the propagation of the double-spend transactions. The main idea of this technique is, although inefficient, that whenever a node receives a transaction, it checks whether the coin has been used. It does so by searching the memory pool to find if there is a transaction with the same source address and amount but with a different output address. If the node finds one, it ignores the transaction and continues to process other transactions in the memory pool, otherwise, it accepts the transaction.

In PoC, we established a blacklist containing all addresses that have a history of participating in attacks. The vendor or observers could view this blacklist and check if the address is blacklisted. If so, the vendor could directly refuse to receive the payment from the address or the vendor. The observer could also be more cautious about a potential malicious behavior.

V. Experiments

We design tow simulation experiments. Considering that there are fewer miners in the early stages of the network, we add a fixed number of miners to the network at each time. When the network matures, the difficulty of mining is increasing too high, so that the number of miners added at each time is not fixed. Because the cost of mining increases at the same time, miners will calculate their Return-on-Investment(ROI) before joining the network.

A. Fixed Potential Entrants

Simple Mining. We developed a software implementation to simulate the miner's behavior. Simulation provides a convenient way to represent the behavior of the miners and run the network without actually spending the computation power. In order to simplify the hash calculation, our model replaced it with a random number generated from an even distribution. The upper bound of a random number depends on the mining difficulty D as in the following formula.

$$nonce < \frac{ht}{D} \tag{8}$$

Where h is the hash rate held by the miner and t is the number of the miner's attempts to engage in mining. When the random number is less than the target value, it is the mining success.

New Arrival. As of the 10th December 2017, the Bitcoin network's total hash rate is equal to 13523349TH/s. The initial hash rate is far less than the actual environment in the simulation experiment, so we simulate timing with new miners carrying fixed computational power to join the network. The new miner will work on the longest chain in the current records. If a miner finds more than one chain with the same length as the longest chain, then the miner chooses one randomly with equal probability. It is noteworthy that even if a new miner and existing miner have the same hash rate, the new miner's likelihood of winning a block would not always lag behind the existing miner who has already won the difficulty reward. As shown in Figure 6, when the new potential entrants, the probability of winning the puzzle for old miners will be reduced accordingly. Since the reward function is convergent, the new miner will eventually have the same rewards as the old miner. In addition, whenever some a miner quits the network, the probability of winning for each miner is changed according to the proportion of their hash rate.

Difficulty Update. In the experiment, we simulated the PoW algorithm and the PoC algorithm and analyzed the changes to the difficulty coefficient. In the PoW algorithm, we set a fixed period to denote as T to update the difficulty coefficient. For example, when T = 20, a new mining difficulty is calculated based on the total number of attempts t, it takes to generate the last 20 blocks. The formula is as follows:

$$D_{new} = D_{old} * \frac{expect\_t * T}{\sum t} \tag{9}$$

where expect_t is the average number of attempts that we expect for mining. If the sum of the actual number of trials is greater than the desired, the difficulty will be reduced, thus increasing the range of random numbers and improving the probability of success. In the PoC algorithm, each miner needs to update the β to get the individual mining difficulty using the formulas (6)(7). A miner's contribution determines the difficulty of mining in the future.

Figure 5 shows the experimental results, we set the initial difficulty to 10000. Whenever a new block is found 10 new

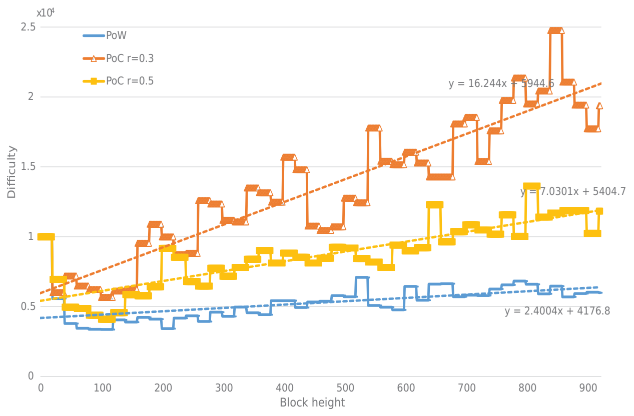


Figure 5. r determines the reward ceiling. This is the relationship between the global difficulty and the height of the block with PoW and PoC algorithm.

miners, who hold a constant hash rate, join the network. At the beginning of mining, the difficulty quickly falls to around 5000 due to lack of computation power in the network. And then as the new miners continue to join, the total hash rate increases. It will lead the global difficulty continues to rise.

The difficulty in PoC algorithm rose more than PoW, dotted lines represent the change trends. If a new miner wants to join it compete with someone with a higher difficulty, the cost of getting the same revenue will be higher. This effectively discourages miners from purchasing additional hashing power to increase the earning. On the other hand, the advantage of mining is capped by β, it guarantees that the new miner will eventually have the opportunity to catch up with the old miner. Although the difficulty is different between the PoC and PoW, the total hash rate of the network is the same. In this simulation that adds a fixed hash rate to the network, energy consumption does not save, but PoC will make the network move to the next stage faster.

Edge Case. We analyzed the effects on the hash rate in two extreme cases in PoC. In the first case, no new miner is introduced to the network. The probability of getting a block when there is only one miner is 1, and the hash rate h eventually converges to βh. When there are two or more miners with same hash rate h at the beginning, each miner eventually gets an equal probability and the hash rate converges to βh. The second case, the network is open for new miners. In our experiment, a new miner joins the network whenever a miner finds a new block. As new miners continue to join, the existing miners' hash rate reach close to βh. The probability of mining a new block decreases for new miners. Figure 6 shows these probabilities under PoW and PoC.

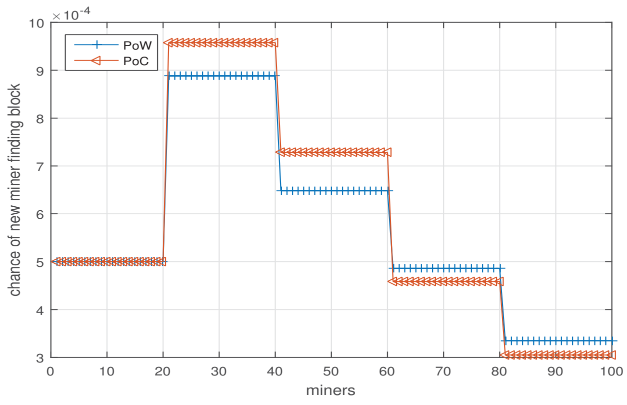


Figure 6. Once one miner find a block there is new miner join the network. As the number of miners increases, the chance of new miner finding a block in the PoC and PoW algorithm shown in the graph.

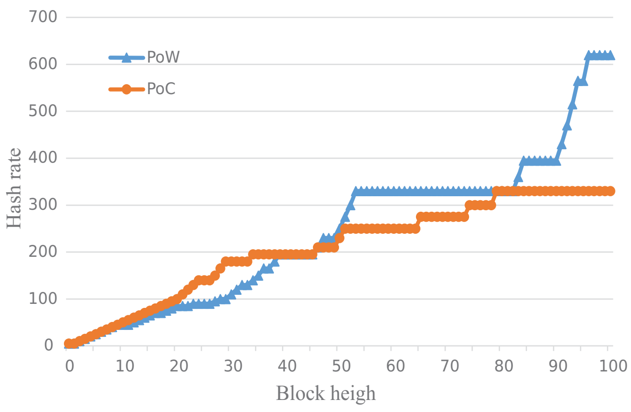


Figure 7. New miners will calculate their ROI before join the network. This graph show the comparison of hash rate changes in PoW and PoC networks.

B. ROI Based Joining

Profitability Calculator. Bitcoin mining calculator is used to calculate mining profitability for miners mining in Bitcoin[20]. The estimated expected Bitcoin earning is based on an equation using the value list in Table I. This is an easy statistical calculation do not take into account the exchange rate fluctuation and a pool's efficiency. In our experiment, we randomly selected Bitcoin price changes over time as a sample of Bitcoin price fluctuations. Potential entrants will consider the operational cost for a period of time must lower than the mining profit during that time. If they use the MiningProfit equation to calculate the annual profit is a loss, the will not join to be a miner.

Table I
THE EQUATION FOR MINING PROFIT.

function	MiningProfit	
input		
	HashRate	eg. 9460.00 GH/s
	Power	eg. 2600.00 Watts
	PowerCost	eg. 0.1 /kWh
	Difficulty	eg. 3839316899029.67
	BlockReward	eg. 12.5
	BitcoinToDollar	eg. 8864.99 USD
	HardwareCosts	eg. 0.00 USD
onput		
	$\mathbf{p}$	Annually profit
begin		
(1)	$hashTime = \frac{Difficulty*2^{32}}{HashRate*10^9}$	
(2)	$powerCostPerYear = 365.25 * 24.0 * Power/1000.0 \; *PowerCost$	
(3)	$blockPerYear = \frac{365.25*24*3600}{hashTime}$	
(4)	$coinPerYear = BlockReward * blockPerYear$	
(5)	$powerCostPerYear = 365.25 * 24.0 * Power/1000.0 \; *PowerCost$	
(6)	$revenuePerYear = coinPeryear * BitcoinToDollar$ *// we don't account for HardwareCosts*	
(7)	$\mathbf{p} = revenuePerYear - powerCostPerYear$ return $\mathbf{p}$	
end		

Proof-of-Work and Proof-of-Contribution financially reward miners for their work. Mining, therefore, remains attractive as long as the financial reward for mining exceeds the financial cost of mining. This mechanism allows for the accumulation of a large and diverse set of miners. The underlying assumption is that if this large and diverse set of miners come to a consensus, the rest of the network can trust that consensus. However, as the number and diversity of miners increases, the marginal benefit of adding more miners decreases. we can surmise that the ideal level of mining difficulty is one where:

- Mining remains financially rewarding for all miners in the network
- Mining becomes prohibitively expensive once the network acquires a sufficiently large and diverse number of miners

PoC provides such a system, by using a different mining difficulty for different miners. Potential entrants to the mining network will calculate their ROI before deciding to mine. As long as the ROI is too low to justify entering the market, the network will not add new mining capacity. The high difficulty for new entrants will keep the ROI low to decrease the likelihood that the network will add unnecessary new miners. Meanwhile, the low difficulty for old miners will keep them active and thus ensure that the network maintains a large and diverse set of miners. This contrasts with a simple adjustment of the difficulty for Proof-of-Work. A difficulty adjustment in Proof-of-Work will merely increase the difficulty for all miners, meaning that in order to stay at the same average transaction time, the network should increase its mining capacity. Proof-of-Work has a comparative disadvantage that as long it is profitable for existing miners to mine, it remains equally profitable for new entrants to mine. That is, until the addition of new miner. As shown in Figure 7. when the same block is generated in the network, the hash rate is lower in the

PoC network and it will be efficient in the future.

VI. Conclusion

Bitcoin mining turns out to be a profitable business, but it wastes a great amount of energy on computation. In order to address the issue of higher computational requirements, we proposed a new algorithm called ”Proof-of-Contribution”, which is based on Proof-of-Work and Proof-of-Stake. Miners’ honesty is represented as success times is used as ”stake” to adjust the difficulty in PoC mining process. The PoC algorithm is friendly to honest miners and penalizes malicious behavior. Experiments have shown that in the same case of calculations, the global difficulty will go up faster in PoC. In the other words, as the network grows, the cost of adding new hash power increases. It encourages miners to be honest and refrain from misbehaviors instead of buying more machines. It is achieved by integrating a PoS component, called success times. The simulation takes ROI into account and the results have shown that PoC increases mining efficiency.

The concept of the blacklist is also used to maintain a record of misbehaving nodes. This concept can be useful for other research related to bitcoin such as cybercrime. We will research some details about this part and other malicious behaviors in the future.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (No.91118002).

References

[1] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

[2] Barkatullah, J., & Hanke, T. (2015). Goldstrike 1: Cointerra’s first-generation cryptocurrency mining processor for bitcoin. IEEE micro, 35(2), 68-76.

[3] Dev, J. A. (2014, May). Bitcoin mining acceleration and performance quantification. In Electrical and Computer Engineering (CCECE), 2014 IEEE 27th Canadian Conference on (pp. 1-6). IEEE.

[4] Bitcoinity.org. (1 Dec. 2018). Bitcoin mining difficulty. Retrieved from http://data.bitcoinity.org/

[5] King, S., & Nadal, S. (2012). Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. self-published paper, August, 19.

[6] Kwon, J. (2014). Tendermint: Consensus without mining. Retrieved May, 18, 2017.

[7] Mizrahi, I. B. C. L. A., & Rosenfeld, M. (2014). Proof of Activity: Extending Bitcoin?s Proof of Work via Proof of Stake.

[8] Karame, G. O., Androulaki, E., & Capkun, S. (2012, October). Double-spending fast payments in bitcoin. In Proceedings of the 2012 ACM conference on Computer and communications security (pp. 906-917). ACM.

[9] Bitcoin Wiki. (1 Dec. 2018). Bitcoin Myths. Retrieved from https://en.bitcoin.it/wiki/Myths#Point_of_sale_with_ bitcoins in.27t_possible_because_of_ the_10_minute_wait_for confirmation

[10] Bitcoin.org. (1 Dec. 2018). Selfish mining. Retrieved from https://bitcointalk.org/index.php?topic=325792.0

[11] Eyal, I., & Sirer, E. G. (2014, March). Majority is not enough: Bitcoin mining is vulnerable. In International conference on financial cryptography and data security (pp. 436-454). Springer, Berlin, Heidelberg.

[12] Matt S. (1 Dec. 2018) Is Bitcoin Currently Experiencing a Selfish Miner Attack? Retrieved from http://scienceblogs.com/builtonfacts/2014/01/11/is-bitcoin-currently-experiencing-a-selfish-miner-attack/

[13] Bitcoin Wiki. (1 Dec. 2018). Bitcoin Transaction. Retrieved from https://en.bitcoin.it/wiki/Transaction#Generation

[14] Mser, M., Bhme, R., & Breuker, D. (2014, March). Towards risk scoring of Bitcoin transactions. In International Conference on Financial Cryptography and Data Security (pp. 16-32). Springer, Berlin, Heidelberg.

[15] Karame, G. O., Androulaki, E., Roeschlin, M., Gervais, A., & ?apkun, S. (2015). Misbehavior in bitcoin: A study of double-spending and accountability. ACM Transactions on Information and System Security (TISSEC), 18(1), 2.

[16] Anuel A., & Esteban O. (1 Dec. 2018). Proof of Existence. Retrieved from http://proofofexistence.com

[17] Spagnuolo, M., Maggi, F., & Zanero, S. (2014, March). Bitiodine: Extracting intelligence from the bitcoin network. In International Conference on Financial Cryptography and Data Security (pp. 457-468). Springer, Berlin, Heidelberg.

[18] Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G. M., & Savage, S. (2013, October). A fistful of bitcoins: characterizing payments among men with no names. In Proceedings of the 2013 conference on Internet measurement conference (pp. 127-140). ACM.

[19] Rosenfeld, M. (2011). Analysis of bitcoin pooled mining reward systems. arXiv preprint arXiv:1112.4980.

[20] Coinwarz.com. (1 Dec. 2018). Bitcoin Mining Calculator. Retrieved from https://www.coinwarz.com/calculators/bitcoin-mining-calculator