Luckcoin: An Electronic Cash system With Two-phase-proof-of-work

Sherlock Holmes

Sherlock.Holmes.luck@protonmail.com

[abstract]： Bitcoin first proposed a peer-to-peer electronic cash system, which enables online payment to be sent directly without the need of any third party.To prevent double payments, the Bitcoin network timestamped all transactions by hashing them into an expanding chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work.The emergence of mining pools in the Bitcoin-like network is inevitable, this is due to the fact that the miners of the mining pool have a higher frequency of cooperative mining than non-cooperative mining. In fact, the mining pool can be seen as a single node, which obviously brought huge risks to the Bitcoin network and violated the vision of one-cpu-one-vote proposed by Satoshi Nakamoto.

This paper innovatively proposes a two-phase-proof-of-work mechanism. It makes the frequency of generating blocks when they are cooperative lower than the frequency of generating blocks when they are non-cooperative, that is to say, for the owner of a miner with several nodes, his expected output of these nodes mining for an account less than the expected output of letting these nodes independently mine for their respective accounts. This design makes all nodes in the entire network independent, and most of the CPU's computing power cannot be effectively organized to attack the entire network.

1 Introduction

Since 2009, Bitcoin [1] was born and became the world ’s first point-to-point cryptocurrency. Blockchain technology has also experienced unprecedented

development in commercial applications. Its core lies in the network environment, without the need of any third party, a strong trust can be established directly.

The key function in the Bitcoin blockchain is to pay without the need of any third party, which is based on the proof-of-work mechanism, which is essentially a decentralized clock [2]. In an early email, Satoshi Nakamoto also explained in detail how the proof-of-work mechanism implements a distributed timestamp server on an peer-to-peer basis [3]. In fact, in a distributed system, it is impossible to associating events with time, until Satoshi Nakamoto invented a solution, that is, the proof-of-work mechanism, which made distributed ledger technology possible.

However the mining pool broken Satoshi Nakamoto's "one-cpu-one-vote" vision. In the initial design of Bitcoin, Satoshi Nakamoto did not consider the mining pool. He believes that it is not realistic to run an efficient mining pool among individuals who don’t know each other. Unfortunately, rather than participating independently, most miners join the mining pools, leading to a consolidation of power. In fact the largest mining pool has accounted for more than 50% network’s total mining capacity. The selfish mining attack is also a very destructive attack on the Bitcoin network, and the selfish mining attack will have a lower requirement for the mining power.

In fact, the mining pool is essentially a product of outsourced work. Since the traditional proof-of-work mechanism is only an effective way to resist sybil attacks. It cannot effectively prevent outsourcingwork, which means that the miners will outsource part of their work to other miners for their own interests. This is because the expected output of cooperative mining between miners will be much higher than the expected output of their non-cooperative mining.

In this paper, we firstly proposed a two-phase-proof-of-work mechanism, and theoretically prove that under this new mechanism, the expected output of cooperative mining will be less than the expected output of non-cooperative mining.This will

effectively prevent the formation of the mining pool, which realizeSatoshi Nakamoto's original vision of"one-cpu-one-vote".

2 No puzzle cannot be outsourced

The proof-of-work (POW) mechanism used in Bitcoin-like cryptocurrencies is to incremented a nonce in the block, the POW involves scanning for a nonce that when hashed, such as with SHA-256, the hash begins with a number of 0 bits. Once a nonce is found that the hash of the block begins with the required 0 bits, the block can't be changed without redoing the work. As later blocks linked after it, the work to change the block would require redoing all the blocks after it.

As introduced above, the Bitcoin-like protocol is built around a hard computational puzzle which can be outsourced. Once a miner outsource his work to other miners, he has a great advantage in solving the puzzle compared with honest miners. If there are some kinds of puzzles that cannot be outsourced, we can directly improve Bitcoin's mining mechanism.

If we design a puzzle-solving problem, every attempt to solve the puzzle requires knowing the private key of the account. For example, we can change the puzzle from "Find a block whose hash value is lower than a specific target" to " A block was found, and the hash value of the digital signature of this block is lower than a specific target. ". Such a puzzle-solving problem would prevent the mining pool administrator from outsourcing work to other miners. This is because the mining pool administrator needs share the private key to the outsourcer. However, such a design has the risk of private key, which is undesirable.

Regarding to the puzzles that cannot be outsourced, we can also refer to the paper by Miller, Andrew, Elaine Shi [4] who constructed a set of puzzles that cannot be

outsourced, but their solutions can be pre-computational, that is, before mining, the solution of the puzzles can be generated in batches in advance, which obviously violates the design of the proof-of-work mechanism.

In fact, the design of the proof-of-work mechanism requires that the puzzles must have features that are difficult to compute and easy to verify.The ease of verification means that the process of solving the puzzle can be split into several sets of verification tasks in parallel, which means that there is no puzzles that cannot be outsourced.

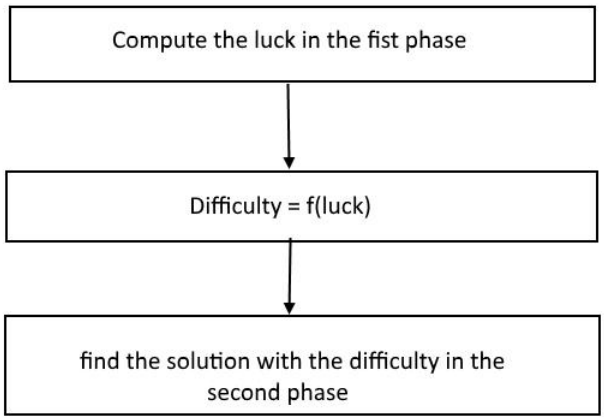
3 Two-phase-proof-of-work mechanism

In this paper, we introduce the luck in the mining process, that is, the participating nodes mined by their own luck, which means that the miners will compute the luck according to their own address and the hash of the previous block during each round of mining, the higher the luck, the less difficult the mining; the lower the luck, the more difficult the mining.

We note that if the computation of the luck is low energy consumption, it still cannot prevent the malicious miner from finding the best one in a large number of addresses in a round (this behavior is equivalent to sybil attack), and then he choose the best address participates in mining, this is unfair to other miners. In order to prevent this malicious behavior, we introduce a lightweight computation when computing the luck, in fact, this is another proof-of-work. The proof of work in the early research was mainly to resist sybil attacks, and later in hashcash [5], this anti-spam system also adopted a similar idea.

In our system, we introduced a two-phase-proof-of-work. The proof-of-work in the first phase mainly focus on the luck; the proof-of-work in the second phase focus on

finding the solution of the puzzle that match the target.



It should be noted that the work of computing the luck is lightweight, this is relative to the number of nodes in the entire network. When the number of nodes in the entire network is small, in order to effectively resist cooperation, the consumption of computing the luck will be lower than the consumption of finding the puzzle's solution of the luckiest node. When the number of nodes in the entire network is sufficient, in order to ensure the security, the consumption of computing the luck may be much higher than the consumption of finding the puzzle's solution of the luckiest node.

Since computing the luck consumes the work of the miners, a single miner has no incentive to computing the luck in batches so as to select addresses with smaller luck to generate blocks.

4 Cooperation vs Non-cooperation

we prove through a mathematical model: the two-phase-proof of work mechanism can effectively resist cooperative mining between nodes.

First, we assume that the node's luck is $l \in L$, where l is uniformly distributed in the space $L = [a,b]$. For each l, the expected mining difficulty is expressed as $f(l)$ with the expected block generation time, where $f(l)$ is a decreasing function.

For a miner who has m nodes, the expected time of generating a block with the strategy that these m nodes mining for one address (not considering the scheduling time of cooperation between nodes, in fact, the expected time of generating a block for cooperative mining will be longer):

$$E_1(m) = \frac{1}{m}\sum_{l\in L} f(l)\frac{1}{|L|} = \sum_{l\in L} f(l)\frac{1}{b-a}\frac{1}{m}$$

For these m nodes, without cooperation, they mined independently for m addresses, the expected time of generating a block is:

$$E_2(m) = \sum_{l\in L} f(l)\Pr(l_{\max} = l) = \sum_{l\in L} f(l)[\Pr(l_{\max} < l+1) - \Pr(l_{\max} < l)]$$

Under the assumption that the block generation times of these m nodes are independently distributed, the above expression can be simplified to

$$E_2(m) = \sum_{l\in L} f(l)[\left(\frac{l-a+1}{b-a}\right)^m - \left(\frac{l-a}{b-a}\right)^m]$$

Comparing $E_1(m)$ and $E_2(m)$, the larger l is and the faster $f(l)$ decreases, it can satisfy $E_2(m) < E_1(m)$.

This means that for miners who have multiple nodes, the expected output of these nodes for independent mining will be higher than the output of letting them cooperate

in mining.

5 Luck vs power

We consider a scenario where a new node A is added to a subnetwork N based on a two-phase-proof-of-work mechanism. It has two strategies: A joins N in a cooperative manner, or A joins N in a non-cooperative manner. When A joins N in a cooperative manner, A's contribution to N is reflected in the increase in A's computing power; When A joins N in a non-cooperative manner, A's contribution to N is reflected in A's luck. If the effect of increased luck on network N is significantly greater than the effect of increased computing power on network N, we can prove that the non-cooperative mining strategy between nodes is superior to the cooperative mining strategy between nodes.

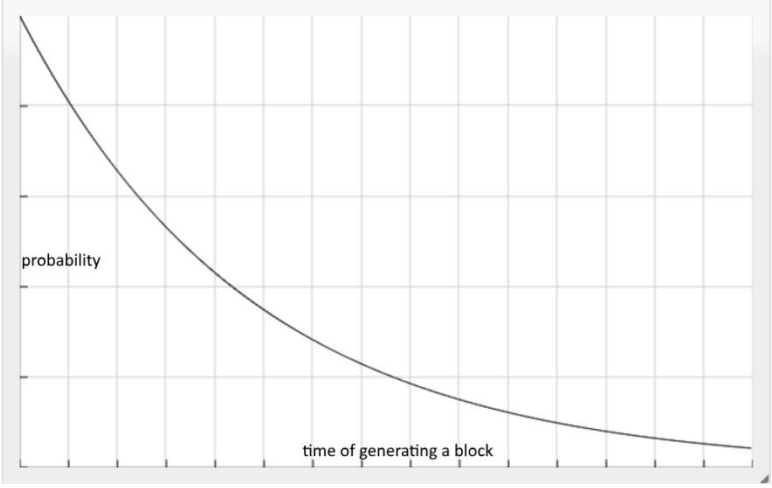
From another perspective, the increase or decrease of computing power is the result of the subjective behavior of miners, and the luck cannot be predicted and controlled by miners in advance. The system is more fair and just, which is exactly what we pursues.

6 Analysis of forks' Probability

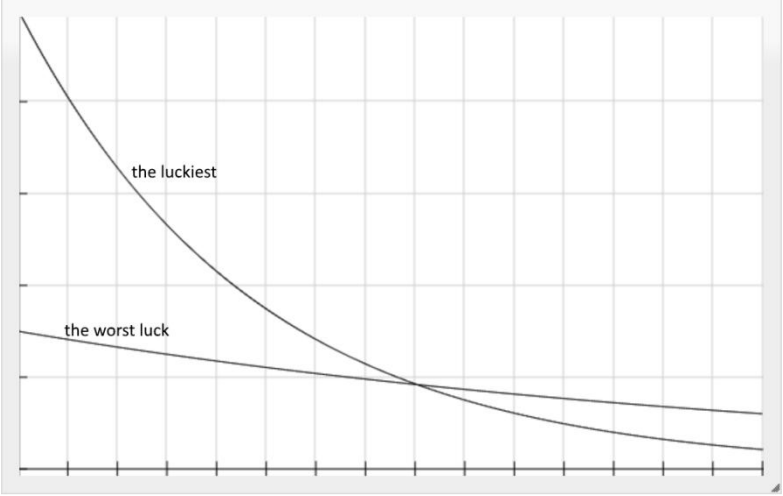
An obvious shortcoming of the proof-of-work mechanism is fork. If there are two or more miners solved cryptographic puzzles almost simultaneously in a short time, the chain will fork. Forking is harmful to the system, and it is the root cause of uncertainty in the system. For the two-phase-proof-of-work mechanism, forks are still inevitable. Below we make a brief comparative analysis of the fork probability under these two mechanisms.

Since the proof-of-work is represented by finding a solution to the puzzle, we can

express the block generation time as the number of nonce attempts t, assuming that the probability of a single attempt is p, then the probability of the block generation time T is $\Pr(t=T)=p.q^{T-1}$, where $q=1-p$.



For the two-phase-proof-of-work, under the condition of the same block generation time, the probability of the block generation time of the node with the best luck being T is $\Pr(t=T)=p.q^{T-1}$. The probability of the block generation time of the node with the worst luck value being T is $\Pr(t=T)=p_1.q_1^{T-1}$, where p_1 is much larger than p.



It can be clearly seen from the above figure that under the same block generation time, the probability of the fork produced by the two-phase-proof-of-work mechanism is

much smaller than the probability of the fork produced by the proof-of-work mechanism.

7 Unpredictability of the number of nodes

For the proof-of-work mechanism, the computing power of the entire network is determined by the difficulty, which is a publicly determined variable.
For the two-phase-proof-of-work mechanism, the change in the difficulty is affected by two factors which are the number of participating nodes and the distribution of computing power. Since the distribution of computing power is completely unpredictable, the number of participating nodes is a parameter that cannot be computed.

In fact, for the two-phase-proof-of-work mechanism, the number of nodes is an important indicator of network security. A malicious attacker who wants to run an attack on the network must accurately evaluate the number of nodes in the entire network in order to make an attack cost budget and make an actual attack deployment.

8 Resist selfish mining attacks

The traditional view is that Bitcoin's mining protocol is compatible with incentives and can resist collusion attacks from minority groups, but scholars such as Eyal [6] believe that Bitcoin's mining protocol is not compatible with incentives. The author proposes a new mining strategy, this strategy can allow a few mining pools to gain more than their honest mining agreement, this strategy is called selfish mining.

Selfish mining is a mining strategy for Bitcoin's proof-of-work mechanism. The simple explanation is that miners choose not to announce the block after mining, and

continue to mine until they meet their own interests. Compared with 51% computing power attacks, selfish mining attacks have a stronger feasibility. As long as a mining pool has more computing power than other mining pools, it can be implemented normally.

Compared with the proof-of-work, a miner can successfully run a selfish mining attack as long as he has the maximum computing power, which may have a computing power far lower than 51% of the entire network's power. In the two-phase-proof-of-work mechanism, since the output of cooperative mining between miner nodes is less than non-cooperative mining, if a miner wants to successfully run a selfish mining attack, it must effectively control 51% nodes of the entire network, this will be more difficult.

It is also very important to explain that in the two-phase-proof-of-work mechanism, it is assumed that there are m nodes in the entire network (except the attacker). When an attacker has $m+1$ nodes, he wants to run a selfish mining attack, in each turn the expected time of generating a block is only about $\frac{1}{m^2}$ unit time less than the normal generating time, that is, under the assumption of $m = 1000$, it needs to selfishly mine 1000000 blocks for catch up with the mainnet, and as m increases, more blocks are needed to catch up with the mainnet.

9 Ideal incentive scheme

Both the Bitcoin-like proof-of-work mechanism and the two-phase-proof-of-work mechanism proposed in this paper are competitive consensus. The feature of the competitive consensus is that there is only one winner per round. The higher fluctuations in probability make small miners dare not participate in mining, the rest are large mining teams.

An ideal incentive plan should be to reward as many honest participants as possible in small amounts. This is similar to the lottery. A lottery is not only a grand prize for each round, but there are many other prizes such as the first prize, second prize, third prize ...

However, in the bitcoin-like proof-of-work mechanism, the computing power is the only proof of the honesty of miners. If the computing power is the only criterion to reward miners, it will inevitably lead to a monopoly of computing power. For the two-phase-proof-of-work mechanism, luck is important in each mining process, and it is a parameter that is not affected by computing power or any other factors. According to the luck, several participants are selected and given rewards. This will greatly reduce the risk of probability fluctuations, which means that miners do not need to form an alliance, and small miners can also participate in mining and make profits.

The luck project will introduce some ideal incentive scheme according to project progress and community development.

10 conclusion

In this paper, we propose a two-phase-proof-of-work mechanism and implement an electronic payment system that does not require any third party. Compared with the electronic payment system based on the proof-of-work mechanism, the expected output of cooperative mining between miners is smaller than the expected output of non-cooperative mining, which can effectively prevent mining pools and realize the vision of one-cpu-one-vote based on luck mining. In addition, we analyzed the security of the two-phase-proof-of-work mechanism from many aspects such as fork probability and selfish mining.

[references]

[1] Bitcoin: A Peer-to-Peer Electronic Cash System.

[2] Blockchain proof-of-work is a decentralized clock.

https://grisha.org/blog/2018/01/23/explaining-proof-of-work/

[3] https://satoshi.nakamotoinstitute.org/emails/cryptography/11/

[4] Miller，Andrew，Elaine Shi. Nonoutsourceable Scratch-Off Puzzles to Discourage Bitcoin Mining Coalitions. Acm Sigsac Conference on Computer & Communications Security. 2015.

[5] A. back. Hashcash – a denial of service counter-measure. 2002.