SCIS 2015 The 32nd Symposium on Cryptography and Information Security
Kokura, Japan, Jan. 20-23, 2015
The Institute of Electronics, Information and Communication Engineers

Lottery Protocol for Cryptocurrency

Junichiro Kume* Masayuki Abe*† Tatsuaki Okamoto*†

Abstract: The Bitcoin system has adopted Proof of Work (PoW), which manages the total number of blocks and prevents double-spending attacks. Since the protocol based on PoW requires miners to solve difficult computational tasks, a problem arises in terms of wasted electricity. Hence, some alternatives to PoW protocols were proposed to avoid wasting computational resources. A leading example of the alternatives is Proof of Stake (PoS), where miners that possess more coins gain an advantage to create new blocks. Proof of Activity (PoA) is a hybrid of PoW and PoS. In the PoA protocol, the follow-the-satoshi procedure is performed to select block creators.

In this paper, we propose a new alternative of PoW for cryptocurrency. The proposed protocol wastes much less energy than PoW and fairer than PoA in such a sense that it randomly determines a creator of the block by performing a lottery. Furthermore, this protocol has the property that an economical cost does not motivate attackers under certain conditions.

Keywords: cryptocurrency, Bitcoin, Proof of Work, Proof of Stake, Proof of Activity.

1 Introduction

In the Bitcoin system, the public transaction ledger called block chain manages transactions. The block chain can be extended to generate a block containing transactions. Generation of a block requires miners to brute-force the lower hash value than the Bitcoin's difficulty. This work is called Proof of Work (PoW). A miner who successfully generates a block can derive coins as a reward.

However, PoW is not economically efficient. It requires large amounts of electricity and computing resources. Hence, new economical alternatives were proposed.

We propose a new cryptocurrency protocol called the lottery protocol. We show that it more economically efficient than PoW and fairer than Proof of Activity (PoA) [1].

In Section 2.1, we present how to generate blocks in proposed protocol. In Section 2.2, we present the probability that the block chain forks as a result of a time lag. In Section 2.3, we present the probability that the block chain forks based on a lottery and methods to prevent this from occurring. In Section 3, we present attack methods and how to secure against each attack or not secure. In Section 4, we present features of this protocol and discuss operational issues.

1.1 Related work

Proof of Stake (PoS) [2] has a mechanism in which only a miner who has coins can generate blocks. This protocol incorporates the concept of coinage: the longer the time a miner has coins, the higher the probability that he will generate blocks. However, there is a problem in that initially coins must be distributed to miners.

PoA is a hybrid of PoW and PoS. This protocol performs a method called follow-the-satoshi N times, and N+1 stakeholders create a block. This method is given a pseudorandom value as the input, and transforms this value into a satoshi (smallest unit of cryptocurrency), and traces transactions to discover the stakeholder who currently controls this satoshi [1,Section3]. Therefore, the number of coins is directly

* Kyoto University
† NTT Secure Platform Laboratories, NTT Corporation

related to the probability that blocks are generated in this process.

Proof of Space (PoSpace) [3][4] based on data storage was presented as an alternative to PoW. This study and Proof of Location (PoL) [5] that do not incorporate a computational effort or an amount of coins are novel approaches. The proposed protocol is related to these alternatives that use a new concept.

1.2 Motivation

Bitcoin is implemented in a peer-to-peer network, and it was designed so that all participants are equal where everyone can become a miner. However, the current Bitcoin system based on PoW is highly-advantageous to the people who have more computing resources. Additionally, a subspecies based on PoS or PoA is advantageous to people who have more coins.

In this paper, we propose the lottery protocol that does not depend on the amount of computational resources (without PoW) and is fairer than PoA with a new concept.

2 Lottery Protocol

The proposed protocol performs a random lottery among miners to decide a block creator. The lottery is basically performed every 10 minutes. Each miner can select not to participate in the lottery.

In the protocol, a miner must pay a small fee to participate in the lottery. In Section 2.1, we assume that this fee is 0.01 coins. Since there is this small fee in this protocol, the attacker must pay more coins to win the lottery. This economical cost that is a large amount of fees does not motivate an attacker.

The destination of the small fee is decided by invoking follow-the-satoshi. Hence, miners who have even a little coin have a chance to gain profits from performing follow-the-satoshi. This distribution gives coins to more people. However, a miner that has more coins receives more coins to gain from invoking follow-the-satoshi.

If we can implement a global clock, then one person cannot broadcast a large of number of lots. A global clock is needed to conform the time that each miner will broadcast a lot. We use this clock to implement a way in which that each miner can receive lots for only a fleeting moment. We can change this time based on the size of the turnout.

2.1 Block generation

Below is a description of how blocks are generated in the proposed protocol.

1. Each miner waits until the gap between the time stamp of the previous lottery and the current time is 10 minutes. While waiting for the next lottery, the miner prepares his public address, 0.01 coins, and the hash of the previous block hash, timestamp, and public address. Each miner decides to whom send the 0.01 coins by invoking follow-the-satoshi with the hash as input.
2. At the next lottery time, the miner broadcasts a lot that contains the hash of the previous block, timestamp, and transaction to send 0.01 coins to the person who was decided by follow-the-satoshi and his own signature for the hash value of this lot.
3. Each miner collects lots that were broadcast as payments for the lottery in a fleeting moment, and adds all the hash values of each lot. Then the miners find the lot that has the closest hash value to the calculation result, and the person who has the public address of the transaction in the lot is elected by the lottery.
4. The miner who was elected by lottery acquires the right to generate a block. A block contains transactions that were used in the lottery, a Merkle root that includes as many transactions as desired by the miner, a timestamp, and his own signature for the hash of this entire block. The miner broadcasts it after generating the block.
5. Each miner can check the block to verify whether or not it includes double spending transactions. If the block includes such a transaction, then each miner follows one of the processes in Section 2.3. These processes describe how to perform a lottery again and depend on the implementation of the alternative Bitcoin system.

2.2 Forked block chain by based on a time lag

According to the blocks generating process described above, branches in the block chain occur due to a time lag, i.e., if Alice has 50 lots and Bob has 55 lots then Alice's lottery result is different from Bob's. If this protocol is implemented in a synchronous network, this branch does not occur. However, a realistic lottery network is an asynchronous network.

This problem will be spontaneously solved since the number of lots that each miner collects exhibits a distribution similar to a mound (time-dependent

dispersal). For example, a lot that was broadcast early in the 10-min period becomes easily widespread while a lot that was broadcast just prior to the end of the time period exhibits poor dispersal. Hence, the lottery result shows the same distribution. Thus, if the majority of miners enact a lottery according to this distribution then a lot is determined as a result. Furthermore, honest miners follow the result due to a majority rule that the longest block chain is correct.

2.3 How to perform a lottery again

In the Bitcoin system, each miner checks a block to prevent attackers from creating a block that contains double-spending transactions. However, since the block creator is selected by lottery, each miner cannot perform a check in this protocol. Hence, a block that the selected miner generates can potentially contain double-spending transactions. Honest miners do not follow the poisoned block, although we must consider it. To address this problem, we consider a ranking list and repetition of the lottery every 10min.

2.3.1. Ranking list

This list is a ranking based on the closeness to a lottery result. If a block contains double-spending transactions then the miner is passed over. Each miner repeats this check until each miner finds the correct block according to this list. So it is not necessary to repeat this lottery again, since we can easily implement this check process using the ranking list. For example, the specific idea is that miners check in order of distance from a calculation result until miners finds a correct block.

2.3.2. Every 10 minutes

This process is simply to perform a lottery again every 10 minutes. Thus, if n lotteries fail then there is a lag of 10(n+1) minutes between the previous lottery timestamp and the next lottery.

However, we note the Denial-of-Service attack [Section3.2], since the probability of stopping the system has a direct relationship with the rate of poisoned blocks.

3 Attacks on Lottery Protocol

3.1 Intentional lottery control

The goal of this attack is to select the attacker's own lot by lottery. In the proposed protocol, the only person who most recently broadcast a lot has an advantage in controlling intentionally a lottery. This is because the final hash value is determined by adding the hash values of all lots and it depends on the last broadcast value. However, a lot that was broadcast just prior to the end of the fleeting moment exhibits poor dispersal, and it is difficult that an attacker most recently broadcasts a lot in the short time period. In the case of an attack that attackers send more lots during a lottery, we show to address this issue more detail in Section3.3.

3.2 DoS attack

This attack intends to stop the service based on this protocol, and is divided into two classes.

1. Attacks using blocks

One way of this class is that if an attacker is elected to a lottery in the legitimate procedure, then the attacker doesn't create a block for 10 minutes. The other way is that attackers generate many blocks that include double-spending transactions. The honest miner must wait due to these attacks until the correct block is generated. However, if the majority of miners are honest, then these attacks are effectively prevented by the ranking list [Section2.3.1]. If many persons are participated in these attacks and generate many poisoned blocks, then these attacks are awfully similar to the 50% attack.

2. Network DoS attack

This attack is intended to prevent honest miners from sending lots. To perform this attack successfully, attackers must broadcast many lots in amounts greater than the receiver's capacity. However, it is difficult, because the time that miners send and receive is limited in only a fleeting moment. Otherwise, this attack becomes successful.

3.3 50% attack

In the Bitcoin system, a 50% attack is to obtain >50% of the total hash power. Attackers can certainly gain possession of the block chain through this attack. For example, an attacker can refuse to include transactions in the blocks that he generates, unless the transactions comply with the attacker's policy.

However in the proposed protocol, we consider that a 50% attack has two attack paths. We set up parameters in the proposed protocol make it how to secure against each attack. We can use the following setup and an economical approach in the case of <50% attack.

Setup

A miner gets R coins as a reward for a block generation. A small fee that a miner must pay to broadcast a lot is c coins. The total number of lots in a lottery denotes N . A miner (or attackers) has p % of the total amount of coins, and broadcast x lots in a lottery. The value of xR/N is defined as the expected value of a block generation. The value of pcN is a benefit gained from performing follow-the-satoshi. The total of participation fees that attackers pay denote xc coins.

The first one is to hold >50% of the lots in a lottery. This means that attackers increase x and x/N is >0.5. However, if the more participants (N) or the more fees (c) are given, then they consume a large number of coins (xc) and it is difficult to attain control. For example, if $N = 5000$ and $c = 0.01$ coins then, $x = 2500$ and $xc = 25$ coins. If $0.5R + 50p$ is less than 25 coins, then this attack is not beneficial for attackers. Hence, it is possible to reduce this attack by increasing the miners and setting a reward and fees.

The second is to obtain >50% of the total coins. This means that p is >0.5 and the value of pcN increases. Hence, attackers gain profits to only wait. However, if attackers only wait for invoking follow-the-satoshi, then the honest people gets >50% of the total amount of coins, i.e., attackers must participate in lotteries. And attackers can easily let x/N become >0.5, because they have >50% of the total coins. It is difficult to reduce this attack's advantage. However, it is also difficult that attackers buy coins from others or win in more lotteries to hold >50% coins.

3.4 Bribe attack

In the proposed protocol, one person generates a block. So his probability for obtaining a benefit decreases as more people participate. Therefore, we consider the idea of a mining pool to share the profits with more than one person.

Attackers send a bribe to many people, and one of the attackers wins the lottery. If the profit is higher than the amount of the bribes, then this attack is successful.

This attack means that attackers simply increase lots (x) in a lottery or stockpiles of coins (p). However, this attack is very critical for the system where one person has only one vote and does not depend on stockpiles of coins.

4 Discussion

4.1 Basic concepts

In the proposed protocol, we used a random lottery. Additionally, we considered a new concept such as PoSpace or PoL that does not use PoW, and decided to use a time constraint and an economics approach in our proposal. We explain these approaches.

Time constraint. A global clock lets miners receive lots in only a fleeting moment. This clock implements a time constraint. This time constraint curbs the number of lots that one person can send. But it is based on an assumption that a communication time in a peer-to-peer network is fair for all miners. If there are people who have a network in which miners send more lots to more people, then the person has an advantage.

Economics approach. We use this approach that each miner must pay a small fee to generate a block. If more people are participated in the lottery, then attackers must pay a large amount of coins to get a reward. The most important aspect of this approach is that this fee is a bit for the individual, but the total of fees is large sums of coins. Hence, if fewer miners, then the attackers have an advantage of attacks.

4.2 One person one vote

To implement the system that is fair to all miners and doesn't give particular miners that have large amount of computational resources an advantage, we must implement a mechanism that one person has only one vote. It is difficult to implement it using only the cryptographic primitives in peer-to-peer network. Of course, implementing a trusted third party, such as an Election Commission, can resolve this problem. However, this assumption is inconsistent with the design for implementation in a peer-to-peer network. In the proposed protocol, a time constraint takes a role as the limiter. We hope that other solutions that can be easily implemented than this.

4.3 Initial distribution

An issue is left to address. It is an initial money supply. In the Bitcoin system, a first block of a block chain is called genesis block. This block's reward is the first coins. However, the system is based on the assumption that miners who want to participate in a lottery have coins. Hence, the system must distribute coins to the interested parties in some way. Of course,

assuming a centralized party, can resolve this problem too. However, we demand a solution that is executed in a peer-to-peer network. We propose a way that an initial miner performs PoW for only the first block. This approach is presented in [2] to use PoW and PoS. To increase participants, we consider that an initial miner sends these coins to miners who want to participate in the system or sells coins.

5 Conclusion

In this paper, we proposed a protocol to perform a lottery among miners. And we argue that the proposed protocol is fair, because miners can participate in a random lottery under certain conditions. And this protocol does not depend on the total amount of computational resources. Additionally, a small fee is an economical cost for attackers. However, it is difficult to implement that one person has only one vote in a peer-to-peer network without a trusted third party.

References

[1] Bentov, I., Lee, C., Mizrahi, A., and Rosenfeld, M. 2014. Proof of Activity: Extending Bitcoins Proof of Work via Proof of Stake. In Proceedings of the ACM SIGCOMM 2014 Workshop on Economics of Networked Systems, NetEcon 2014. http://eprint.iacr.org/2014/452.

[2] Bitcoin wiki: Proof of Stake https://en.bitcoin.it/wiki/Proof_of_Stake

[3] Dziembowski, S. and Faust, S. and Kolmogorov, v. and Pietrzak, K. 2013. Proofs of Space. http://eprint.iacr.org/2013/796.

[4] Ateniese, G. and Bonacina, I. and Faonio, A. and Galesi, N. 2013 Proofs of Space: When Space is of the Essence. In 9th Conference on Security and Cryptography for Networks, SCN 2014. http://eprint.iacr.org/2013/805.

[5] MrBea. 2014. Proof of location. Bitcoin Forum thread. https://bitcointalk.org/index.php?topic=520126.0;all