SCIS 2016 2016 Symposium on
Cryptography and Information Security
Kumamoto, Japan, Jan. 19 - 22, 2016
The Institute of Electronics,
Information and Communication Engineers

New Cryptocurrency Protocol without Proof of Work

Junichiro Kume* Masayuki Abe*† Tatsuaki Okamoto*†

Abstract: The Bitcoin system has adopted a mechanism called *Proof of Work* (PoW), which contains certain requirements on the block generation. Protocols based on PoW require miners to solve difficult computational puzzles, that cause an issue of wasted electricity. Furthermore, the system has another problem about data storage. A public distributed ledger, called the *block chain*, logs all existing transactions. Miners must store the whole of it to verify the legitimacy of transactions.

In SCIS2015 we proposed a lottery protocol for a cryptocurrency in which a block generator is randomly selected. In this paper we improve the protocol by increasing the efficiency of a method called *follow-the-satoshi* and reducing the size of the public ledger.

Keywords: Cryptocurrency, Bitcoin, Proo of Work, Proof of Stake, Democoin.

1 Introduction

1.1 Bitcoin

Recently, several electronic currencies and payment systems have been considered. However they need a trusted central authority for the money transfer. Bitcoin [1], an electronic cash system without a trusted third party, was proposed in 2008, and since then many follow-up papers have been published. New system proposals and implementations have been discussed in Bitcoin Improvement Protocol (BIP) [18]. Currently, Bitcoin is the first and most popular peer-to-peer cryptocurrency.

Generally, there are following two problems to consider a new electronic currency in peer-to-peer network.
(1) Double spending

Malicious users can easily replicate the money, since an electronic currency is an electronic data. For example, even though Alice sent the money to Bob, Alice remains having the money data. Therefore Alice can send the same money data to anyone else.

(2) Manipulation

As is the case with double spending, malicious users should not be able to tamper with electronic currencies. For example, Alice should not be able to increase by adding changes to her money. Needless to say, she should not be able to tamper with the history of transaction too.

To address these problems, the Bitcoin system has adopted a mechanism called "block chain", that works over the notion of "Proof of Work". Its security requires the majority rule that admits the longest block chain as the only valid chain. The public transaction ledger called block chain includes all existing transactions. The blocks in this ledger are linked in proper linear, sequential order. Each block contains a hash of the previous block and transactions. A transaction is a digitally signed statement, which describes the amount of bitcoins transfered from a sender address to a receiver address. The bitcoin address is an identity in the Bitcoin protocol and is the hash of a public key for the ECDSA signature scheme. Malicious users cannot make illegal money transfers, since people, who are participating in the Bitcoin network, verify the correctness of the digital signatures by the public keys, and check that the input addresses had not previously already been spent. A typical example of transactions is shown in Figure 1.

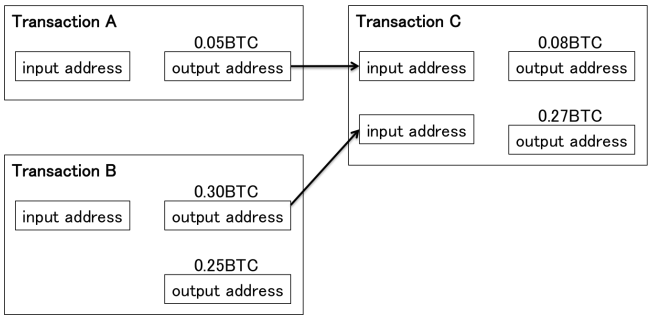

Figure 1. An abstruct example of a transaction

The block chain can be extended to generate a block containing as many transactions as desired by the miner. Generation of a block requires the miner to find a hash value which meets a requirement called *difficulty* [2]. More precisely, the miner must find a "nonce" value so that the hash value (of the block) will contain a run of zeros. This task is called Poof of Work (PoW) and a person who performs PoW is called a *miner*. A miner who successfully generates a block can derive coins as a reward. This difficulty target of PoW is periodically calculated so that blocks are added to the block chain

* Kyoto University
† NTT Secure Platform Laboratories, NTT Corporation

on average every 10 minutes. Therefore it takes a role as managing the total money supply in the Bitcoin system.

Moreover, PoW and majority rule prevent malicious users from double spending and manipulation. The attackers attempt to insert the transactions to the longest block chain admitted by the miners, or must create a longer chain which contains an arbitrarily created blocks. In other words they must create blocks by using more hash power than honest parties. To prevent this attack Satoshi Nakamoto argues that honest miners control a majority of hash power in [1].

Recently, Bitcoin usage and the number of real-world stores which accept Bitcoin payments increase. However, the current Bitcoin has two weaknesses to practically operate it as a global electronic currency.
(1) Waste of Electricity

Miners perform PoW to get a reward. Generally, however, the probability that a user mine successfully for oneself is very small. They can be affiliated with a group called "mining pool". In a mining pool they mine together and split the reward coins equally according to the amount of their contribution to mining a block. In order to increase their benefit they purchase high hash rate machine, e.g. ASIC (application-specific integrated circuit), GPU (Graphics Processing Unit) and FPGA (Field-Programmable Gate Array), and use them to compute SHA-256 for PoW. Therefore the Bitcoin system has a problem in which PoW is not economically efficient, since these machines require large amounts of electricity and computing resources [3][4]. Hence, new economical alternatives were proposed in a previous study. We introduce them in Section 1.3.

(2) Huge Storage for Block Chain

In December 2015, the size of the block chain is about 48GB [17], and unfortunately a miner generally has to download the whole of it (i.e., a large history of transactions) to verify the legitimacy of transactions. Moreover the number of transactions increases in every 10 minutes as the number of users gets increased [5]. Hence, various types of software have been developed so as to reduce the file that miners must download. SPV (Simplified Payment Verification) client, e.g. MultiBit, Bitcoin Wallet and Electrum, refers to the block chain on an external server and download the transaction history as many as miner needs. However it isn't peer-to-peer and is based on the assumption a trusted server that has the full block chain exists.

1.2 Our Contribution and Outline

We proposed a lottery protocol [12] in SCIS2015, and can reduce the waste of electricity by using this protocol. The reason is that the block generator is selected by a lottery. Moreover, in this protocol, a winning rate on a lottery depends on the number of "lottery transactions" which require small fees. Thus malicious users must spend a lot of coins to keep winning. More information is described in Appendix.

In this paper, we improve this protocol to mitigate the above weaknesses. The miner in the new protocol stores the all latest account information (without the history of transactions) as a *Merkle tree*, and creates a block that contains the merkle root of it. Introducing the merkle tree reduces the size of the public ledger, and increases the efficiency of the method called *follow-the-satoshi*. We show the more detailed description of the efficiency in Section 3.1. In Section 3.3.2, we specify three instantiations according to the usage of the current Bitcoin and present the scalability with them.

1.3 Related work

・Proof of Stake

Proof of Stake (PoS) [6] was proposed as an alternative to PoW. It is a mechanism in which only a miner who has coins can generate blocks. The block generation probability (i,e., the difficulty in this protocol) is directly related to the amount of coins a miner has, e.g. if a miner holds 1% of the coins then he can generate a block with a possibility of 1%.

Let H be a hash function, PK be a bitcoin address, t be a timestamp, B_{t-1} be a hash of the previous block and *balance* be an amount of coins with respect to PK. A miner can generate a block if

$$H(B_{t-1} + PK + t) \leq 2^{256} \times balance/difficulty.$$

There is another case where the balance is not adopted. It incorporates the concept of coinage as a replacement for a balance. Coinage is that the longer time a miner has coins, the higher probability that he/she will generate blocks. We define coinage as

$$Coinage = balance \times age.$$

This protocol possesses resistances to 51% attack, where the attacker has 51% of all coins and controls the block chain. (1) Attackers' cost would be more expensive to mount this attack. (2) If executing this attack was successfully achieved then the attacker's coins lose their value.

PPCoin [7] is a typical example of cryptocurrencies using PoS. Let B_{t-1} be a hash of the previous block, $txout_A$ be an unspent transaction output address, and *difficulty* be a constant so that the blocks get generated once every 10 minutes on average. PPCoin's miners can create the next block if

$$H(B_{t-1}, current\_time, txout_A) \leq$$
$$difficulty \times coins(txout_A) \times age(txout_A).$$

In the above formula, variables that users can control are $current\_time$ and $txout_A$. Hence the hash power for the PoS mining (i.e., the number of queries to a hash function) is much smaller than PoW, and this protocol is more economically efficient than Bitcoin.

However, the issue of electrical power remains in that initially coins must be distributed to miners, since the

coin must exist for the PoS mining. To address this problem, PoW is used for the initial money supply in PPCoin.

・Proof of Activity

Proof of Acticity (PoA) [8] is a hybrid of PoW and PoS. $N+1$ stakeholders create a block in this protocol. A miner performs PoW, generates an empty block header, and performs a method called *follow-the-satoshi* N times. This method is given a pseudorandom value (e.g., a hash value of a transaction) as input, and transforms the value into a Satoshi (smallest unit of cryptocurrency), and traces transactions to discover the stakeholder who currently controls this Satoshi [8, Section 3]. $N-1$ miners selected by this method sign the hash of an empty block header and broadcast it. Finally, the Nth stakeholder adds transactions to the empty block header and sign this entire block. Therefore, the number of coins is also directly related to the block generation probability.

・Democoin

A cryptocurrency called Democoin has been proposed and its scalability was also presented by using three sample instantiations in [9]. This cryptocurrency differs from Bitcoin in a variety of ways.

First, the public ledger file that a miner must download is very small in this cryptocurrency. The reason is that the block chain is not adopted, and alternatively all account information called "a full status report" is stored on a storage provider/facilitator. In other words, the system manages the latest account information and does not hold the large transaction history. For instance, single account information that a public key PK has an amount X is

$$P = SIG_{PK}(PK, \#X).$$

SIG_{PK} means that a user computes a digital signature with a signing key SK corresponding to PK. Thus, other users can verify this account information with PK by accessing the cloud storage. A full status report consists of all latest public key information in which each information takes the same form like an above equation. Moreover, this public ledger forms a merkle tree that is a standard cryptographic tool. A merkle tree holding four account infomation is shown in Figure 2.

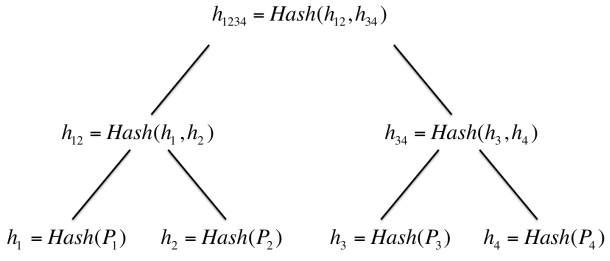

Figure 2. Merkle Tree

The merkle trees' structure allows very efficient checking that a given value was calculated in the tree. In case that we check about a given data, we must have nodes on a path from a given leaf data to the root and their siblings. In the above merkle tree we can check an account information P_3 by having h_{1234}, h_{12}, h_3 and h_4 without h_1 and h_2. Therefore, even though the number of nodes considerably increases, the verifiers' download size is very small to check.

Next, verifiers, who are hierarchical, are randomly decided to verify whether transactions are double-spending or not. The helper verifiers check and send the valid transactions to higher level verifiers. The only top-level verifiers finally store a full status report and payment transactions to a storage provider. Below is a description of Democoi's protocol.

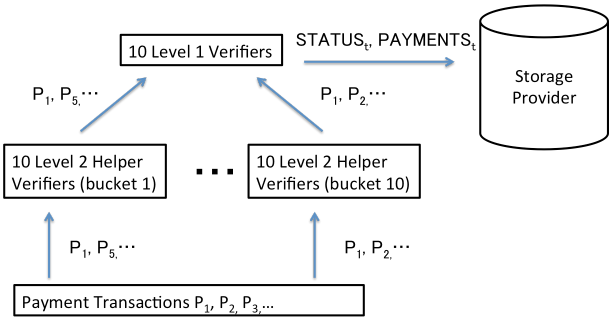

Figure 4. Democoin diagram

The following conditions are required for a verifier. Let H be a hash function, and let vt be an unpredictable beacon which is publicly known at time t. Then, a verifier's public key PK is selected if

$$H(PK, vt) \leq p. \quad *$$

Fairness of this verifier selection depends on the unpredictable beacon and a public key. Anyone can create public keys easily, hence anyone (and attackers) can create public keys meet an above condition by consuming vast amounts of CPU power. In that case there is a problem in that the probability of a successful attack that inserts illegal transactions and drops arbitrary valid transactions in a public ledger file. Four solutions are discussed to address this problem in the same paper, all of them are not practical, though. Moreover there is the assumption that unpredictable beacons exist is strong.

To update this public ledger at round t (e.g. t = 10 minutes), verifiers must download it and check valid payments at round t+1. For instance, a payment PAY of an amount X which transfers from a public key PK to a public key PK' at time t+1 is

$$PAY = SIG_{PK}(PK, PK', \#X, t+1).$$

The public ledger at round t+1 is stored and calculated by PAY_{t+1} and it at round t.

Verifiers periodical update of the file also results in a faster payment time than Bitcoin's. Bitcoin users generally must wait for a block, which contains their own transactions to be more than 3-depth because the block's chain may not be the longest chain. Hence, a

Bitcoin user should wait about 30 minutes after generating a block that contains own transactions. In this protocol, on the other hand, a user only has to wait one round (i.e., the time for the transaction to be reflected in the public ledger).

・SybilAttack

To consider a new cryptocurrency protocol in peer-to-peer network, it is necessary to prevent the *Sybil Attack*. The attacker can control a large fraction of the nodes in a system by creating a large number of fake identities. An identity is a public key (a bitcoin address) in Bitcoin system. Thus, anyone can easily create multiple identities, and control a large number of parties. To prevent this attack, resource-based defenses, location-based defenses and social network approaches [10] are proposed. Proof of Work is a type of resource-based defenses. Moreover, a new framework for polling in peer-to-peer network by using CAPTHA as a prevention of sybil attack has proposed in [11].

2 Lottery Protocol

This proposed protocol in [12] performs a random lottery among miners to decide a block generator. The lottery in which each miner can select (not) to participate is basically performed every 10 minutes as well as Bitcoin's difficulty. If a miner want to participate in the lottery, then a miner must broadcast a small fee transaction, called the *lottery transaction*. Note that this fee is normal output, not *transaction fee* which is paid for a miner as a reward. This fee is periodically computed as the same amount of coins (e.g. 0.01 coins) and its output address is decided follow-the-satoshi. Since there is this fee, 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, and is discussed in Appendix. Informally, a block generation process of this protocol is following.

1. Each miner decides an output address by the follow-the-satoshi, and broadcast the lottery transaction at a fixed time.

2. Miners derive a hash value from adding the all hash values of the valid lottery transactions.

3. An input address (a public key) of the closest the hash value to the calculation result is elected in the lottery.

4. The miner has the address generates a block that contains a hash of the previous block, the timestamp, lottery transactions and payment transactions.

3 Improvement the lottery protocol

In this paper, the efficiency of follow-the-satoshi subroutine and the size of the public ledger are much-improved. In order to achieve these effects, new lottery protocol requires that a block has a merkle root of a miner's full status report, which is introduced as all account information in Section 1.3. This merkle root at round t is computed from the previous merkle tree, payment transactions and lottery transactions at round $t-1$. Hence, to create a correct block, the miner have to keep his own full status report up to date, and have to prove that his full status report is correct by broadcasting the his block. If the merkle root is correct, then the others follow the block. Introducing a merkle root is based on the idea that the longest block chain manages authenticated transactions among Bitcoin users. The block information is shown in Figure 3.

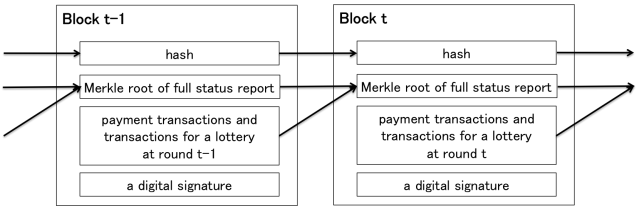

Figure 3. Our Block Chain

Note that a merkle tree might be different from another merkle tree that even contains same leaves. The reason for this difference is because the leaves data are not stored in a fixed order. Thus, we index each address between 0 and the sum of all the unspent coins, and store accordingly to the order.

3.1 Increasing the efficiency of new follow-the-satoshi

Follow-the-satoshi is described in Section 1.3. There is an implementation of this method was written in Python and queries the SQLite database of the Abe block explorer [13, 14]. However this is somewhat slow, since in each hop the system must query the database via tree lookups that run in $\mathcal{O}(logK)$, where K denotes the total number of transactions. Hence the worst case time bound is $\mathcal{O}(KlogK)$. Another implementation written in C++ was proposed in [8, 15]. In this implementation each newly minted coins are partitioned into intervals, where each interval corresponds to a bitcoin address who currently controls those coins. Hence the system can remove all the intermediate traversal paths. The worst case time bound is $\mathcal{O}(logJ) \leq \mathcal{O}(logK)$, where J denotes the total number of intervals.

PoA miners derive a pseudorandom value from an empty PoW header and do follow-the-satoshi with the value. In our protocol, we derive a pseudorandom index value (a bitcoin address) from input addresses and a fixed time for a lottery. Since tracing transactions is made redundant, we can get from the latest all account information in $\mathcal{O}(1)$. This efficient follow-the-satoshi is referred to as the new follow-the-satoshi in the rest of this paper.

3.2 Block generation process

Below is a description of how blocks are generated in the our 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 for 0.01 coins (as a lottery fee), the hash of the previous block and a fixed timestamp. And the miner decides to whom send the 0.01 coins by invoking new follow-the-satoshi, and signs the lottery transaction.

2. At the next lottery time, the miner broadcasts the lottery transaction in peer-to-peer network.

3. Each miner collects the lots that were broadcasted in a fleeting moment, and verify the correctness of the digital signatures and the new follow-the-satoshi, and check whether the transaction contains unspent input addresses and the appropriate time-stamp or not. Then the miners derive a hash value from adding the all hash values of the valid lottery transactions, and find a lottery transaction as an elected person who has the closest hash value to the calculation result.

4. The miner who was elected by a lottery acquires the right to generate a block. A block contains a merkle root of his full status report, lottery transactions at this round, a merkle tree that includes as many payment transactions as desired by the miner, a timestamp, and his own signature for the hash of this entire block. After generating the block, the miner broadcasts it in the peer-to-peer network.

5. The others who received it can check the block to verify whether it includes double spending transactions or not, and whether a merkle root of the full status report is valid or not. Otherwise, miners follow one of the processes in [16, Section 2.3]. These processes are how to perform a lottery again, and depend on the implementation of the alternative Bitcoin system.

3.3 Comparison with Democoin

3.3.1 Verifiers

In Democoin, the verifier selection depends on a substantially unpredictable value and public keys. Thus the paper proposed four solutions to prevent sybilattack (or corruption) by malicious users as follows:

> One possibility is to elicit an entry fee, or a (pro-rated) yearly fee from each public key in the system. A second possibility is having the probability of turning a public key PK into a verifier key depend also on the amount of money... A third possibility is having a separate entity, call it the verifier registration authority (VRA), who certifies (anonymously or not) the public keys eligible to be selected as verifier keys. A fourth possibility is to have a mixture of verifiers: for example: (a) fixed set of verifiers (possibly none) as in Spreadcoin; (b) a set of dynamically selected verifiers (possibly none); and (3) a set of registered over-time verifiers (possibly none).

In our protocol there is not such problems, since the miners take on the role of verifiers as well as Bitcoin's miners. No matter how many bitcoin address (public keys) sybil attacker creates, he can not participant / monopolize in a lottery by using them as long as the addresses has no coins. Furthermore this idea follows in the distributed approach that is a basic principle of Bitcoin.

3.3.2 Scalability

If the number of users (bitcoin addresses) increase, then the full status report increases also. Thus, miners must need to store more data in his storage. In this section, we consider the data miners must store, and use the analyzing techniques for the full status report introduced in [9] as a reference. Three instantiations shows Democoin's scalability in [9], and they assume that a single authenticated payment (or a status report record) is about 100Bytes. Let N be the number of users (bitcoin addresses), let T be transactions every 10 minutes, and let STATUS be the size of a full status report. Democoin's Urban instantiation is :

N = 300K, T = 1,000, STATUS = 30MB.

Compared to Democoin, we must add the above number of transactions every 10 minutes to transactions for a lottery. Fortunately the size of the full status report doesn't change, i.e., the size is equal to the Democoin's size because output addresses of lottery transactions are selected from the public keys has existed. More precisely, lottery transactions does not spend to a newly created address. Let L be a number of transactions for a lottery, and the results are following.

(1) The Urban Instantiation
N = 300K, T = 1,000+L, STATUS = 30MB
(2) Regional Instantiation
N = 3M, T = 10,000+L, STATUS = 300MB
(3) International Instantiation
N = 30M, T = 100K+L, STATUS = 3GB

That is, in the Urban Instantiation, users and transactions numbers are slightly larger than those the usage of Bitcoin in February 2015. Thus, in our instantiations, we transform them according to the current Bitcoin system (in December 2015). Referring to Block chain info [17], they are following.

- Number of Users: 400,000.
- Number of transction per block: 1,400.

We present current three instantiations according to this information.

(1) The Urban Instantiation
N = 400K, T = 1,400+L, STATUS = 40MB

(2) Regional Instantiation
N = 4M, T = 14,000+L, STATUS = 400MB
(3) International Instantiation
N = 40M, T = 140K+L, STATUS = 4GB

4 Conclusion

In SCIS2015 we proposed a lottery protocol [12] in which the block generator is selected by a lottery. Since the miners must spend a small fee, Syibil attackers can not control a large fraction in lottery transactions, and the economical cost does not motivate an 50% attacker. In this paper we improved this protocol so that the public ledger miners must store is smaller than the size of Bitcoin's block chain. More specifically, we can reduce the block chain size from about 48GB to about 40MB. Furthermore the follow-the-satoshi method becomes very efficient through the introduction of a full status report, which contains latest all account information. Hence, we can mitigate the waste of electricity and the huge storage for the block chain by using this protocol.

References

[1] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Consulted, 1(2012):28, 2008. https://bitcoin.org/bitcoin.pdf

[2] Bitcoin wiki: Difficulty https://en.bitcoin.it/wiki/Difficulty

[3] NateAnderson. Mining Bitcoin stakes power, but is it an "environmental disaster"? April 2013. http://arstechnica.com/business/2013/04/mining-bitcoins-takes-power-but-is-it-an-environmental-disaster/.

[4] Bitcoin Computation Waste, http://gizmodo.com/the-worlds-most-powerful-computer-network-is-being-was-504503726, 2013.

[5] Malte Moser and Rainer Bohme. Trend, Tips, Tolls: A Longitudinal Study of Bitcoin Transaction Fees. Financial Cryptography and Data Security (2015) http://fc15.ifca.ai/preproceedings/bitcoin/paper_8.pdf

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

[7] Sunny King and Scott Nadal. PPcoin: Peer-to-peer crypto-currency with proof-of-stake. http://peercoin.net/assets/paper/peercoin-paper.pdf

[8] 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 Networ- ked Systems, NetEcon 2014. http://eprint.iacr.org/2014/452.

[9] Sergey Gorbunov and Silvio Micali. 2015. Democoin: A Publicly Verifiable and Jointly Serviced Cryptocurrency. https://eprint.iacr.org/2015/521.pdf

[10] H. Yu, M. Kaminsky, P. B. Gibbons, and A. D. Flaxman. Sybilguard: defending against sybil attacks via social networks. Expanded Technical Report IRP-TR-06- 01, Intel Research, Pittsburgh, Pittsburgh, PA, Jun 2006. http://www.pittsburgh.intel-research.net/people/gibbons/papers/sybilguard\-tr.pdf

[11] Giulia Alberini, Tal Moran and Alon Rosen. Public Verification of Private Effort. https://eprint.iacr.org/2014/983.pdf

[12] Junichiro Kume, Masayuki Abe, Tatsuaki Okamoto. Lottery Protocol for Cryptocurrency. SCIS2015

[13] Tobey, J. 2011. Abe: block browser for bitcoin and similar currencies. https://github.com/jtobey/bitcoin-abe.

[14] Abe Block Explorer https://blockexplorer.com

[15] blockparser https://github.com/killerstorm/blockparser/blob/master/cb/fts.cpp

[16] Bitcoin wiki: coinbase https://en.bitcoin.it/wiki/Coinbase

[17] Bitcoin Block Chain Info, https://blockchain.info/ja/ , December 2015

[18] Bitcoin Improvement Proposals https://en.bitcoin.it/wiki/Bitcoin_Improvement_Proposals

A Analysis of Lottery Protocol

A.1 Intentional lottery control

The goal of this attack is to select the attacker's own lot by a lottery. In our protocol, the only person who most recently broadcast a lot has an advantage in intentionally controlling a lottery. This reason is that the final hash value is calculated by adding the hash values of all lots, and depends on the latest broadcasted lot. However, it is difficult that an attacker most recently broadcasts a lot in the short time period. This is because miners have fairly network communication costs / resources, and a lot that was broadcasted just prior to the end of the fleeting moment exhibits poor dispersal among miners. Thus, to gain a profit the attacker must simply send more lots in a lottery. We show and address this issue more detail in Section A.3.

A.2 Resistance to DoS attack

This attack intends to stop the service, and is divided into two types.

1. Attacks using blocks

This attack type is that if an attacker is elected to a lottery in the legitimate procedure, then the attacker does not create a block for 10 minutes or generates many blocks that include double-spending transactions. The honest miner must wait due to these attacks until another miner generates a correct block. However, if the majority of miners are honest, then these attacks are effectively prevented by the ranking list in [12, Section2.4.1]. If many persons are participated in these attacks and generate many poisoned blocks, then we can consider that these attacks are awfully similar to the 50% attack.

2. Network DoS attack

This attack is intended to prevent honest miners from sending lots. Generally, if the number of participants increase, then protocols in peer-to-peer network become more secure against such a DoS attack. To mount this attack successfully, attackers simply broadcast many lots in amounts greater than the receiver's capacity. However, it is difficult, because all miners and attackers have common network communication cost. Thus the attacker must invest network resources in some way according to the number of honest miners.

A.3 50% attack

In the Bitcoin system, 50% attack is to obtain >50% of the total hash power, and the attackers can certainly gain possession of the block chain through this attack. For instance, an attacker can refuse to include transactions in his blocks, unless the transactions comply with the attacker's policy. However, in our protocol, the attacker can not always monopolize the block chain, because a block generator is randomly selected by a lottery. In this regard, we argue that our protocol is more secure for the 50% attack than Bitcoin protocol.

We can divide 50% attack into two types. One is a approach in which the attacker generates many lots or bribes miners so that their lots are >50% of the all lots in a lottery. The other approach is the attacker has >50% of the total coins. However, this attack costs too much coins. In other words, it is difficult that attackers buy coins from others or keep winning in more lotteries to get >50% coins. Thus our protocol has resistance to this 50% attack as well as PoS. To analyze these attacks, we define variables in the following.

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 broadcasts 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 new follow-the-satoshi. The total of participation fees that attackers pay denote xc coins.

Analysis

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

The second is to obtain >50% of the total coins. This means that the attacker has p% (>0.5) coins and can increase pcN. Hence, if this attack becomes successful, attackers can gain profits (pcN) to only wait for invoking new follow-the-satoshi by the others miners. However, since the honest miners ordinarily gets coins as a block generation reward in the legitimate procedure, attackers must participate in lotteries to maintain 50% coins. Moreover, even though after this attack, if honest miners keep winning lotteries, then we can decrease the attacker's fraction of the total coins.