Proof of spending in a block-chain system

Chunlei Liu*

Abstract

We introduce proof of spending in a block-chain system. In this system the probability for a node to create a legal block is proportional to the total amount of coins it has spent in history.

1 INTRODUCTION

In 2009, Satoshi Nakamoto [Na] introduced the notion of block-chain as well as the notion of proof of work into P2P cash systems, giving birth to the famous Bitcoin, which is the first P2P cash implemented in practise.

In 2011, the notion of proof of stake was posted in bitcoin forum by a user named Quantummechanic. Various proof of stake systems were then formulated, see, e.g. [KN][BGM][NXT][ML][BPS][DGKR][KRDO]. Unfortunately, all known proof of stake systems seem vulnerable to long-range attacks, see, e.g. [Bu][Pd].

In this paper we present a proof of spending system. In this system the probability for a node to create a legal block is proportional to the total amount of the coins it has spent in history. We shall see that, in the proof of spending system, the adversary trying to build a longest block-chain would earn nothing.

2 TRANSACTION

In this section we recall the notion of transaction in a P2P cash system.

*Shanghai Dengbi Comm. Tech. Co. Ltd., Shanghai, China. 714232747@qq.com
A transaction in a P2P cash system is a collection of the following components: time of the block, address of the payee, amount of payment, unspent transactions, the change, and signature of the payer.

\[
\begin{array}{|c|}
\hline
\text{TRANSACTION} \\
\text{Time:} \\
\text{Payee:} \\
\text{Payment:} \\
\text{Unspent Tx: } \#1, \#2, \cdots, \#n. \\
The change: \\
Payer’s Signature: \\
\hline
\end{array}
\]

A P2P cash system is a P2P system where nodes broadcast valid transactions. A transaction is valid if and only if the sum of the amount of payment and the amount of the change is equal to the total amount of coins in the unspent transactions.

3 BLOCK-CHAINS

In this section we recall the notion of block-chains in a P2P cash system.

A block-chain in a P2P cash system is a chain of blocks.

\[
\text{BLOCKCHAIN: } \text{Block0} \rightarrow \text{Block1} \rightarrow \text{Block2} \rightarrow
\]

A block in a block-chain is a collection of the following components: time of the block, hash of the previous block, new transactions added to the block, address of the block creator, nonce such that the hash of the block begins with a number of zero bits.
A block-chain system is a P2P cash system where nodes broadcast valid blocks. A block in a block-chain system is valid if and only the previous block as well as the transactions added to the block are valid.

4 PROOF OF WORK

In this section we recall the notion of proof of work systems.

Let $D$ be a natural number. A proof of work system with target difficulty $D$ is a block-chain system where block $B$ is legal if and only if its hash satisfies the threshold:

$$\text{nlz}(\text{hash}(B)) \geq D.$$ 

Here $\text{nlz}(\text{hash}(B))$ denotes the number of leading zero bits of hash($B$).

5 TIME TO FIND A POW BLOCK

In this section we review the calculation of the expected number of operations for a computer to find a block which begins with $D$ zero bits.

Throughout this paper, we assume that the time for a CPU to perform an operation is constant and independent of the CPU, and take it as the unit time.

Let $T$ be the time for a CPU to find a new block which begins with $D$ zero bits. One can show that

$$\text{P}(T = n) \approx 2^{-D}(1 - 2^{-D})^{n-1}.$$
It follows that 

\[ E(T) \approx 2^D. \]

Suppose that \( r \) CPUs are working to add a new block to an existing block-chain. Let \( T_i \) be the time for the \( i \)-th CPU to find the next block. One can show that

\[ P(\min\{T_1, T_2, \cdots, T_r\} = n) \approx (1 - 2^{-D})^{r(n-1)} - (1 - 2^{-D})^{rn}. \]

It follows that the expected time for \( r \) CPUs to find a block is

\[ E(\min\{T_1, T_2, \cdots, T_r\}) \approx \frac{1}{1 - (1 - 2^{-D})^r} \approx \frac{2^D}{r}. \]

Therefore the expected time for \( r \) CPUs to build a block-chain of length \( L \) is

\[ \approx \frac{2^D L}{r}. \]

### 6 DOUBLE SPENDING

In this section we review the double-spending attack on a proof of work system.

Suppose that the adversary wants to spend the same coin in two transactions, say \( \text{Tx}_1 \) and \( \text{Tx}_2 \). Suppose that \( \text{Tx}_1 \) is first built in a longest block-chain, say \( C_1 \). This means that the adversary has spent the coin successfully in \( \text{Tx}_1 \). In order to spend the same coin in \( \text{Tx}_2 \), the adversary must include \( \text{Tx}_2 \) into a block, add it to a shorter block-chain, say \( C_2 \), and does its best to make that block-chain the longest some time later. As no rational node will help the adversary in building block-chain \( C_2 \), the adversary must build \( C_2 \) alone. Therefore it is very difficult for \( C_2 \) to outpace \( C_1 \) unless the adversary has more CPUs than the honest party.

### 7 PROOF OF STAKE

In this section we recall the notion of proof of stake systems. There are various kinds of proof of stake systems. Here we present the simplest one, which we call it a proof of balance system.
The proof of balance system with target difficulty $D$ is a block-chain system where block $B$ is legal if and only if
\[
\text{nlz}(\text{hash}(B)) \geq D - \log_2(1 + \text{bal}(\text{creat}(B); \text{chain}(B))).
\]
Here creat(·) is the creator of a block, chain($B$) is the block-chain before $B$, and bal($A; C$) is the balance of node $A$ in block-chain $C$.

8 SECRETE MINING

In this section we review the secrete mining attack on a proof of balance system.

Suppose that nodes $A_1, A_2, \cdots, A_r$ are working to add a new block to block-chain $C$. Let $T_i$ be the time for node $A_i$ to find the next block. One can show that
\[
P(\min\{T_1, T_2, \cdots, T_r\} = n) \\ \approx \prod_{i=1}^{r}(1 - 2^{-D}\text{bal}(A_i; C))^{(n-1)} - \prod_{i=1}^{r}(1 - 2^{-D}\text{bal}(A_i; C))^n.
\]

It follows that the expected time for the $r$ nodes to find a block is
\[
E(\min\{T_1, T_2, \cdots, T_r\}) \approx \frac{1}{1 - \prod_{i=1}^{r}(1 - 2^{-D}\text{bal}(A_i; C))} \\ \approx \frac{2^D}{\sum_{i=1}^{r}\text{bal}(A_i; C)}.
\]

Suppose that the adversary with initial balance $B_a$ wants to earn the mining rewards by building a block-chain secretly at first and publish it when it becomes the longest. Suppose that the reward to the creator of a block is $R$. Then the expected time for the adversary to build a secrete long block-chain of length $L$ in which it spends no money is
\[
\approx \sum_{i=1}^{L} \frac{2^D}{B_a + iR} \\ \approx \frac{2^D \log L}{R}.
\]

But the expected time for the honest party with initial balance $B_h$ to build a long block-chain of length $L$ in which the adversary spends no money is
\[
\approx \sum_{i=1}^{L} \frac{2^D}{B_h + i(R - F)}
\]
\[ \approx \frac{2^D \log L}{R - F}, \]

where \( F \) is a lower bound for the total transaction fees of a block in the block-chain. It follows that, in a long run, the secret block-chain would outpace the block-chain maintained by the honest party. The same philosophy can be applied to attack other kinds of proof of stake systems.

9 PROOF OF SPENDING

We now present a proof of spending system.

The proof of spending system is a block-chain system where block \( B \) is legal if its hash satisfies the threshold:

\[ \text{nlz}(\text{hash}(B)) \geq D - \log_2(1 + \sum_{K \in \text{chain}(B)} \text{spen}(\text{creat}(B); K)), \]

where \( \text{chain}(B) \) is the block-chain before \( B \), \( \text{spen}(A, K) \) is the total amount of coins spent by \( A \) in transactions in \( B \).

10 SECURITY

In this section we analyse the security of a proof of spending system.

Suppose that nodes \( A_1, A_2, \cdots, A_r \) are working to add a new block to block-chain \( C \). Let \( T_i \) be the time for node \( A_i \) to find the next block. One can show that the expected time for the \( r \) nodes to find a block is

\[ \mathbb{E}(\min\{T_1, T_2, \cdots, T_r\}) \approx \frac{2^D}{\sum_{i=1}^{r} \sum_{K \in C} \text{spen}(A_i; K)}. \]

Suppose that the adversary wants to build a long block-chain in shortest time. His best strategy is to transfer to himself in every block a fixed amount of coins, say \( S \), such that the amount of transaction fees is equal to the block-creating reward. Then the expected time for the adversary to build a long block-chain of length \( L \) in which it spends no money is

\[ \approx \sum_{i=1}^{L} \frac{2^D}{iS} \]
\[ \approx \frac{2^D \log L}{S}. \]

Nevertheless, even if the adversary succeed in building a longest block-chain, he earns nothing at all!
11 VARIATION

We now present a variation of the proof of spending system presented in §9.

In this section, the proof of spending system with target difficulty \(D\) and time parameter \(e\) is a block-chain system where block \(B\) is legal if its hash satisfies the threshold:

\[
\text{nlz}(\text{hash}(B)) \geq D - \log_2(1 + \sum_{K \in \text{chain}(B), \text{age}(K; B) \leq e} \text{spen(creat}(B); K)),
\]

where \(\text{age}(K; B)\) is the number of blocks between \(K\) and \(B\).

12 GENERALIZATION

We now present a generalization of the proof of spending systems presented in §9 and §11.

Let \(w(\cdot)\) be a nonnegative decreasing weight function, \(v(\cdot)\) a nonnegative increasing speed function, and \(f\) a positive increasing function. The proof of spending system corresponding to \((w, v, f)\) is a block-chain system where block \(B\) is legal if its hash satisfies the threshold:

\[
\text{nlz}(\text{hash}(B)) \geq D - f(\sum_{K \in \text{chain}(B)} w(\text{age}(K; B)) \cdot v(\text{spen}(\text{creat}(B); K))).
\]

References

[BGM] I. Bentov, A. Gabizon, and A. Mizrahi, Cryptocurrencies without of proof of work, CoRR, abs/1406.5694, 2014.

[BPS] I. Bentov, R. Pass, and E. Shi, Snow white: Provably secure proof of stake, http://eprint.iacr.org/2016919, 2016.

[Bu] V. Buterin, Long-range attacks: The serious problem with adaptive proof of work, https://download.wpsoftware.net/bitcion/old.pos.pdf, 2014.

[NXT] The NXT Community, NXT whitepaper, https://bravenewcoin.com/assets/Whitepapers/NxtWhitepaper-v122-rev4.pdf, 2014.

[DGKR] B. David, P. Gaži, A. Kiayias, and A. Russell, Ouroboros proos: An adaptively-secure semi-synchronous proof of stake protocol, http://eprint.iacr.org/2017573, 2017.
[KN] S. King, and S. Nadal, Ppcoin: Peer-to-peer crypto-currency with proof of stake, https://ppcoin.net/assets/paper/ppcoin-paper.pdf, 2012.

[KRDO] A. Kiayias, A. Russell, B. David, and R. Oliynykov, Ouroboros: A provably secure proof of stake block-chain protocol, In J. Kakz and S. Shacham, editors, CRYPTO 2017, Part I, vol. 10401 of LNCS, 357-388, Springer, Heidelberg, 2017.

[Mi] S. Micali, ALGORAND: The efficient and demacratic ledger, CoRR, abs/1607.0134, 2016.

[Po] A. Poelstra, Distributed consensus from proof of stake is impossible, https://download.wpsoftware.net/bitcion/old.pos.pdf, 2014.

[Na] S. Nakamoto, A peer-to-peer cash system, http://bitcoin.org/bitcoin.pdf, 2008.