



Proof of Bid as Alternative to Proof of Work

Wai Kok Chan¹(✉), Ji-Jian Chin², and Vik Tor Goh²

¹ Faculty of Informatics and Computing, Multimedia University,
63000 Cyberjaya, Malaysia
wkchan@mmu.edu.my

² Faculty of Engineering, Multimedia University, 63000 Cyberjaya, Malaysia

Abstract. Proof of Work (PoW) protocol for cryptocurrency uses an excessive amount of electricity to secure the network. Many PoW coins do not have sufficient hashing power to secure itself. There are many alternatives to PoW, such as Proof of Stake (PoS), merge-mining etcetera, which uses much less electricity. However, these alternatives have some drawbacks either in terms of security, complexity, and scalability. In this paper, an alternative to Proof of Work (PoW) called “Proof of BID” (PoB) protocol introduced. PoB makes use of existing bitcoin PoW to secure all transactions, thus consuming virtually no electricity. PoB also addresses most of the drawbacks faced by PoW alternatives. We have disclosed a systematic method on how to effectively re-used bitcoin PoW to secure a blockchain with the same level of bitcoin security. A few designs issue to improve the blockchain scalability is given. We have explored various attack scenarios and suggested some remedies.

Keywords: Blockchain · Proof of Work · Proof of Bid · Consensus

1 Introduction

In bitcoin, each miner will merge the “unconfirmed transaction output” (UTXO) into a block. All included transactions are hashed to obtain a Merkle root to form a bitcoin block header. Then, a miner performs sha256 hashes multiple times by changing the 32bit nonce, timestamp and other possible fields on the bitcoin header. The miner would repeat this process until the hash output of the bitcoin header matches the bitcoin current difficulty level. If it failed to do so, the miner would have to change the Merkle root value and repeat the process. Changing the Merkle root value, it may involve adding, removing or changing some of the transactions in the current mine bitcoin block.

As of 8th Sept 2018, the current bitcoin mining difficulty level is 7,019,199,231,177 (<https://bitcoinwisdom.com/bitcoin/difficulty>) and current hash rate is 49,290,360,795 GH/s. A difficulty level of 1 is equivalent to performing a 2^{32} SHA256 hash computation. Throughout 2009, the bitcoin difficulty level is 1. The first bitcoin miner which produces a block with the nonce value that matches the current level of difficulty will be rewarded with 12.5 bitcoin as of 2018. A block is deemed to match the current difficulty level when the hash output of the bitcoin block is lower than the current difficulty target. The mining difficulty will be readjusted every 2016 blocks so that on average, one block

is produced every 10 min. All this computation is useless and consumed an excessive amount of electricity. According to de Vries [1], in 2018 bitcoin network is using 2.55 GW of power which is comparable to countries like Ireland 3.1 GW.

There are many alternatives to “Proof of Work” (PoW) such as Hybrid Proof of Work/Stake (*DASH* coin), Delegated Proof of Stake (*EOS* coin), Delegated Byzantine Fault Tolerance Proof of Stake (*NEO* coin), Proof of Authority (*VeChain* Coin), Proof of Burnt (*XCP* and *SlimCoin*), Proof of Importance (*NEM* Coin), Proof of Devotion (*Nebulas* Coin), Proof of Space (*FileCoin*), Delayed Proof of Work (*Komodo* Coin), Hash Graph (*Hedera* Coin), Directed Acyclic Graph (DAG: *IOTA* Coin), Distributed Hash Table (*HOLOCHAIN* Coin) and Block Lattice (*NANO* Coin). Most of these non-100% PoW alternatives use an alternative consensus protocol that consumes very little electricity. However, these alternatives come with one or multiple disadvantages in terms of security, scalability, complexity and most importantly, an extremely high learning curve for a user to understand. Section 3 will discuss some of the key consensus protocols.

In this paper, we proposed PoB, which uses bitcoin block as an external source of randomness. Thus, virtually no electricity is used. Whenever, the latest found bitcoin block arrived, the whole Bitcoin block is hashed to obtain a 256-bit hash value. The general idea is to have miners bid for the hash value for every BTC block interval. Miner with the closest bid will get the mining reward and will consolidate the block into PoB blockchain. This idea sounds simple, but there are many security, network and consensus issues that need to be addressed before PoB can operate safely. Some of the security issues are Denial of Service Attack, Scalability, Bitcoin Block Orphaning, Fork and Costless Simulation. Some of the consensus issues include bid cost revision, consensus on bids submissions and which chain is valid in the event of a fork.

In Sect. 2, the related work on alternatives to PoW is described. Section 3 describes some of the preliminaries. The detail description of the PoB protocol is described in Sect. 4. The non-bidding transaction is not described here because these transactions can be modeled similar to bitcoin or other alt-coin transactions. Security issues are discussed, in Sect. 5, followed by the conclusion in Sect. 6.

2 Related Work

In Proof of stake system [2], some signing keys can determine the future blocks. This key may belong to original users who later lost their coin or sold their coin to someone else. These keys can fork another chain which new users cannot distinguish from true ones. At the same time, it is costless to recreate any alternative history which favors the majority of the randomly selected key owner. In addition to that, an adversary may bribe other stakeholders to extend an invalid block which favors them instead of extending a valid block. In Proof of stake [3, 4], periodic check-point in the blockchain is proposed to prevent an attack that attempts to change the history beyond a certain point. However, it is argued in [2], that this is not a distributed consensus.

In Proof of Activity [5], a miner generates an empty block with a certain level of difficulty and then it broadcast the block to N stakeholder. The network selects these N stakeholders. The first N-1 stakeholders check the validity of the entire block, signs it and broadcast it. The Nth (last) stakeholder wraps the blocks with all the transactions and broadcasts it to the network. The mining reward is divided between the N stakeholders and the miner. An attacker who has x -fraction of stakeholders will have a probability of x^N to mining a block under attacker control. The probability to mine a block under an honest network is $(1 - x)^N$. Thus, the attacker must be fast enough to generate with a p fraction of the honest stake is online. An attacker with y fraction of the total stake needs more than $\left(\frac{1}{y} - 1\right) \cdot p^N$ times the hash power of the honest miners to gain an advantage over the network. Thus, the hash power required to performed any attacker is amplified by power N times, making it much better than traditional PoW.

In Proof of Luck [6], a Trusted Execution Environment (TEE) in the Intel SGX platform is used to compute the luck via POLMINE and POLROUND function. In TEE, system time cannot be fake (Proof of time) and the owner can verify that a specific device has done a particular computation (Proof of Ownership). The POLROUND function ensures each device waits for the start of each round before mining the block. The POLMINE function takes the current block header and the previous block to generate a random nonce execute. The largest nonce value will be the winner and the block will be added into a blockchain. However, there is a need to trust the hardware vendor.

Cryptocurrency such as NANO [7], DAGCoin and IOTA are based on Directed Acyclic Graph (DAG). These DAG-based coins have improved performance and security because most of the transaction verification is distributed. In NANO, each account managed its' own transaction and balance via its' blockchain. Few representatives are selected to monitor the network. These representatives will prevent conflicting transactions (double spending) via a voting process. Only blocks with the majority vote are included in the blockchain. In IOTA, each new transaction must choose two previous transactions to verify. In this way, the transaction can be feeless.

PHANTOM [8] proposed a scalable BlockDAG protocol as an improvement over Spectre [9, 10]. Phantom can support faster block generation. In PHANTOM, the user decides the kind of throughput required and set the relevant "k" value. A k-cluster is a subset of blocks in a DAG that is connected to every block but not connected to at most k blocks. A greedy algorithm is used to distinguish honest blocks from a dishonest one. The author suggested that PHANTOM and Spectre should be run together. This ensures faster confirmation time for non-conflicting transactions and also able to detect conflicting transactions such as double-spending. However, the detection time is very slow.

In Proof of Burnt [11], a user can send a trace of bitcoins to a burn address which is un-spendable in exchange for another token coin. A burn address is an address with an unknown private key. However, there is a need to prove that the burner does not

know the private key. Thus in bitcoin, any addresses with the hashes of any script that evaluate to false such as $2 + 2 = 5$ can be used as a burnt address. This script proves that a user had burnt their bitcoin.

Proof of Burnt can be used to boot scrap another coin such as Counterparty XCP. This process ensures a particular coin having some intrinsic value, the initial XCP token ex-changed at around 1000 XCP token per one bitcoin (BTC). Thus, the XCP token has some intrinsic value because it is created by burning bitcoin. This burning process has created some criticism from the community because the actual bitcoin is lost forever. There are alternatives to Proof of Burn which does not destroy any bitcoin. The bitcoin can be locked in the main bitcoin blockchain when it is converted to an alternative coin running in a side-chain [12]. Users can convert back their alternative coin back to bitcoin by destroying/burning their alternative coin in the side-chain. Once these alternative coins are destroyed, an equivalent value of bitcoin is un-locked in the main bitcoin blockchain. However, these alternative coins in the side chain still need to run a consensus protocol and some of them might use merge-mining.

In SlimCoin [13] Proof of burnt, miner will use real money to buy the SlimCoin. Miner will burn SlimCoins in its mining process to produce a transaction hash. The burn hash is calculated by multiplying a decay multiplier with the internal hash (presumably it is the transaction hash however it is not mentioned in the white paper). Miner with the best hash will get the block reward. Burning existing coin is for mining is almost similar to buying a mining rig in PoW. There are many missing details in the whitepaper. A more detail review may require source code investigation however the Github last commit was done on 7th Nov 2014. <https://github.com/kryptoslab/slimcoin>.

In PoW coin such as NameCoin, Ixcoin, DevCoin, IOCoin and GroupCoin [14, 15] uses the same PoW algorithm as bitcoin. Merge mining is used to secure these coins. In merge-mining, one or more PoW alt-coin is mined together with bitcoin by inserted a scriptSig 44 bytes long containing the block hash of the alt-coin into the bitcoin block. Merge-mining enables PoW alt-coin to improve their hash rate by leveraging on bitcoin mining power. However, a bitcoin miner must first know which alt-coin to merge-mine. Bitcoin miner must find it is profitability and lastly agreed to merge mining it in its mining pool. Thus, not every bitcoin miner performed merged-mining and not every PoW alt-coin will be selected to be merge-mined in a merged mining pool. Generally, bitcoin miners are already preoccupied with many other issues. Since bitcoin miner can merge-mining any child blockchains almost at zero additional cost, it can attack any child blockchain. In 2012, bitcoin mining pool Eligius performed a 51% attack on Coiled-Coin by mining empty blocks [15]. This event had annihilated Coiled-Coin.

3 Preliminaries

Table 1 below shows the variable definition. In our PoB protocol, users can transfer funds to any Coinbase addresses for bidding purposes.

Table 1. Variable definition

| Variable | Description |
|----------|---|
| a_i | bidder with a Coinbase address a_i |
| b_i | the bid value submitted by bidder a_i |
| $t1_i$ | timestamp when the bid is submitted |
| $t2_i$ | timestamp when the blinded bid is revealed |
| F | Fix parameter values for current block such as current block height, previous Block Hash, bidding cost, timestamp etcetera |
| $h1_i$ | $h1_i = \text{blake256}(a_i, b_i, t_i, F)$ This hash value calculated and sent by a_i , it is used to prevents a_i from changing its bid value. Note: b_i is never sent during the bidding process |
| sk_i | Secret key for bidder with address a_i |
| pk_i | Public key for bidder with address a_i |
| B_i | Bid message by sent by a_i $B_i = \{a_i, 0, t_i, h1_i, F\}$; $\text{sign}(B_i, sk_i)$; Receiving full node will accept B_i if $\text{verify}(B_i, pk_i) = \text{true}$; |
| R_i | Bid Reveal Message sent by a_i ; $R_i = \{\text{sign}(B_i, b_i, sk_i), B_i, b_i, F\}$; Receiving full node will accept R_i is valid if $h1_i = \text{blake256}(a_i, b_i, t_i, F) \wedge \text{verify}(R_i, pk_i) = \text{true}$; |
| BidHash | $\text{blake256}(\text{sort}(B_1, B_2, \dots, B_n))$ |
| BidHash' | BidHash value of previous PoB Block |
| τ_i | TraceBid Message sent by a_i ; $\tau_i = \{\text{sign}(\text{BidHash}', sk_i), F\}$; Receiving full node will accept τ_i is valid if $\text{verify}(\tau_i, pk_i) = \text{true}$; |

The hash function used must have the property of collision resistance, preimage resistance and second preimage resistance.

4 PoB Protocol

4.1 General Overview

PoB uses the latest bitcoin block as an external source of randomness which nobody can predict. A miner must maintain a bitcoin full-node status and a full copy of the PoB blockchain for PoB mining operation. PoB Block time is the same as Bitcoin. When a new bitcoin block is discovered, the miner will hash the whole Bitcoin block with blake256 [16] hash function to obtain a 256-bit BTCHash.

$$\text{BTCHash} = \text{Blake256}(\text{FutureBTCBlock});$$

A few blocks before the calculation of BTCHash, each bidder/miner submits a bid to a few random “Peer-to-Peer” (P2P) full nodes. These bids propagate to the whole network via GOSSIP [17] protocol. A bid will consist of the predicted BLAKE2 hash value (256 bits) of a future full bitcoin block. In order to simplify the protocol, each Coinbase address can submit only one bid per block interval. The winning miner is chosen by this self-explanatory tie-breaking pseudocode function.

```
Winner = CoinbaseAddress(LowerBidValue(Nearest(AllBids,BTCHash)));
```

If there are multiple miners submitted the same bid then the miner with the lowest Coinbase address will win. There is only one winner per PoB Block. Thus, the blockchain will not be bloated. In this PoB protocol, we make four assumptions

- (i) Every node is synchronized to the “Network Time Protocol” (NTP) server right to seconds.
- (ii) The broadcast message delivery via gossiping is reliable where all miners will receive the sent message within 15 s.
- (iii) More than 50% of the participating miner is honest. However, dishonest miners can collude together by adding or deleting certain bidding information to their advantage.
- (iv) The (Time Stamp Authority) TSA server service [18, 19] is reliable and the expired keys are still available for historical verification.

4.2 Bidding Transaction Control

Bid Filtering and Fair Bidding. PoB protocol will accept a maximum of 100 bid submission per block interval so that only a small fraction of the blockchain belongs to bid transaction. Adversaries can acquire many coin base addresses to submit as many bids as possible. Thus, a bid filtering algorithm is required to prevent excessive bid flooding. For each block, each bid transaction will go through these two functions, as shown in the pseudocode below.

```
BidderString = XOR(BidderAddress, Blake256(LastBTCBlock), LastBTCHeight);
AcceptBid = mod(BidderString, 1000) + (PastStatistic(BidderAddress) x 100);
```

The first 100 bids with the lowest AcceptBid value will be accepted. If the AcceptBid value is the same, then the lowest coin base address is considered. Thus, a miner with the most fund can monopolize the mining process. In order to prevent that each miner will retrieve every bid transaction Coinbase address from confirmed block PoB (1000L + 1) to PoB(1000L + 1000) where $L \in \{0, 1, 2, \dots\}$. The retrieved Coinbase address will be hashed into a counting BLOOM filter. The function prototype PastStatistic (BidderAddress) will return the number of times each address had bid for every 1000 PoB blocks. Thus, the value of AcceptBid increased tremendously for repeated bidders. Thus, Coinbase addresses which bided in the past have a lesser chance to bid. Every 1000 blocks, the BLOOM filter is reset.

Bidding Reward Carried Forward. The winning bidder takes all the transaction fees (\$TX, excluding the bids reward) within a single block. The total Bid (\$TotB) reward consists of the total bid (\$Bid) by all bidders in the current block and carried forward bid (\$C/F) from the previous block. If all reward is paid within a single block, a miner may place as many bids as possible using different Coinbase addresses.

If the \$TotB is more than \$10, then half of the bidding capital will be carried forward as a reward for the next block. This method will prevent a monopolistic miner from grabbing all the reward. The miner is indirectly forced to bid for a more extended period to minimize loss. Due to the high \$C/F value, other miners may be incentivized to place their bid. Figure 1 show the pseudocode for mining reward (\$Reward) paid to the miner in each block.

```

$TotB = $Cur + $C/F;
If $TotB < $10
    $Reward = $TotB + $TX;
Else
    $Reward = $TX + ($TotB)/2;
    $C/F = $TotB/2;

```

Fig. 1. Mining Reward Pseudocode

Bidding Cost Adjustment. For each block interval that received more than 100 bids, the bidding and deposit cost will be increased by $x_1\%$ to reduce the number of eligible bidders. The initial bidding cost per miner is \$1 plus \$19 deposit. The bidding and deposit cost will be reduced by $x_2\%$ if there are not more than 100 bids per block received for x_3 consecutive blocks. The optimal value for x_1 , x_2 and x_3 is still under investigation. The cost adjustment pseudocode is shown in Fig. 2.

```

If NoOfBid > 100
{ BidCost = BidCost x (1 + x1%);
  DepositCost = DepositCost x (1 + x1%);
  Count = 0;
}
Else
{ Count++;
  If (Count == x3)
  { BidCost = max(MinBidCost, BidCost x (1 - x2%));
    DepositCost = max(MinDepositCost, DepositCost x (1 - x2%));
    Count = 0;
  }
}
}

```

Fig. 2. Bidding Cost Revision Pseudocode

Aggressive cost adjustment algorithm can result in zero eligible bidders for the next block interval. There may be a possibility that PoB protocol is used for alt-coins or coin

running in a side-chain. Initially, there may be no activity for months or years. Thus the side-chain or alt-coin operator must be the miner of last resort. In each block interval, if there is no bid, no block can be generated even though there are some transactions not related to bidding. Thus, one free mandatory bid must be made available to side-chain or alt-coin operators so that mining can continue. When the number of transactions grows, new miners will join.

4.3 Bidding Details

Block Numbering: Bitcoin Block G and PoB Block G are defined as $BTC(G)$ and $PoB(G)$ respectively. PoB genesis block $PoB(G)$ is created at $BTC(G)$ where $G \in \{1, 2, \dots\}$, $K \in \{4, 5, 6, \dots\}$ and $(K-3) > G$. PoB and Bitcoin have the same block height. When $BTC(K)$ is found, $PoB(K)$ is created with one block delay. The Bid data in $PoB(K-3)$, BidRevelation data in $PoB(K-2)$ and TraceBid data in $PoB(K-1)$ determines the winning miner for $BTC(K)$. The winning miner will then construct $PoB(K)$ consists of Bids for $BTC(K+3)$, BidRevelation for $BTC(K+2)$ and TraceBid for $BTC(K+1)$.

Bitcoin Block Orphaning: Bitcoin has an orphaning rate of 0.5%. Miner will bid three blocks in advance to prevent PoB from bidding on an orphan block. Most of the orphaning process already resolved by then. However, this enables adversaries to create three future blocks in advance. In order to prevent this, the latest PoB block $PoB(K)$ must include the hash of the latest BTC block $BTC(K)$. In case $BTC(K)$ is orphaned by $BTC(K')$, there is a side pointer in $PoB(K)$ that points to the whole orphaned $BTC(K)$ block and the hash of $BTC(K')$. Thus in PoB, there is no blockchain reorganization. Figure 3 shows the block diagram.

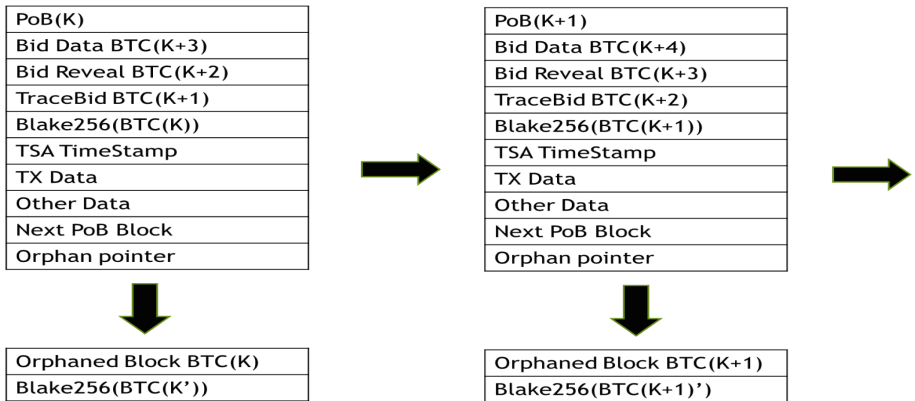


Fig. 3. Block Diagram for PoB Block.

Bidding Process: There are three separate phases in each bidding process for one BTC block. The three phases are the bidding phase in $BTC(K)$, the Bid Revelation Phase in

BTC(K+1) and the TraceBid phase in BTC(K+2). All bidding transactions must be confirmed in the blockchain before bids revelation. All bid revelations must be confirmed before the TraceBid process. The TraceBid process prevents adversaries from forking the PoB chain and also used to resolve other consensus issues. Upon completion of the TraceBid process, when BTC(K+3) is found, the winning miner is determined and reserved the right to create PoB(K+3). Upon discovery of BTC(K+4), PoB(K+3) is created when BTC(K+4). Table 2 shows the bidding process for one bidding cycle. In the actual bidding process, the bidding process overlaps with bid revelation and the TraceBid process.

Table 2. One bidding cycle time line

| Second | Event | Timing |
|-------------------|---|--------|
| 0 s | BTC(K) is found | T1 |
| 0–200 s | Miners submit their bids for BTC(K+3) directly | |
| 0–300 s | Submitted bids are relayed to all miner | |
| 300–400 s | Miner sort all bids from the smallest Coinbase address and calculate the BidHash Send the BidHash to all Miner | T2 |
| 400–BTC(K+1) | All miner reach consensus on all bids | |
| 600 s (estimate) | BTC(K+1) is found | T3 |
| 600–800 s | Miner reveals bid for BTC(K+3) directly | |
| 600–1000 s | Reveal Bid are relayed to all miner | |
| 1000 s–BTC(K+2) | All miner reach consensus on Reveal bids | T4 |
| 1200 s (estimate) | BTC(K+2) is found | T5 |
| 1200 s–1300 s | Miner sign the TraceBid for BTC(K+3) based on BidHash received in BTC(K+4) stored in PoB(K+1) | |
| 1200 s–1400 s | Signed TraceBid is relayed to all miner | |
| 1400 s–BTC(K+3) | Miner reach consensus on which blockchain to follow | T6 |
| 1800 s (estimate) | BTC(K+3) is found. Winning Miner is determined. Winning Miner will create PoB(K+3) when BTC(K+4) is found. The cycle repeats | T7 |

4.4 Bidding Consensus

Network Issues: A miner can cause consensus problem by sending B_i, R_i, τ_i messages during the last few seconds. In consequence of that, GOSSIP’s protocol does not have sufficient time to disseminate the info to all full nodes. A full node is a node that maintains the full copy of the blockchain and may optionally participate in mining. Thus, miners have 200 s to send their B_i or R_i messages directly and 100 s to send τ_i messages. Full nodes that relay miners’ messages to other nodes will have an additional

100 s. There is a mechanism to distinguish direct messages from miners and relay messages from full nodes.

According to data [20] available from 1st January 2017 until 5th April 2017, the average time to propagate a block and transaction to 50% and 90% of the node is shown in Table 3 below. Therefore, 200 s for message and block propagation should be sufficient.

Table 3. Block and transaction propagation speed

| 50% of block | 90% of block | 50% of transaction | 90% of transaction |
|--------------|--------------|--------------------|--------------------|
| 2.398874 s | 11.77294 s | 3.447737 s | 14.26984 s |

Consensus on Submitted Bid: Miner receive bid transactions $\{B_1, B_2, \dots, B_n\}$ from other miners. All miners must reach a consensus on all the submitted bid. Each miner sort all bid transaction from the lowest Coinbase address and perform a Blake256 hash on the sorted message as shown in the preliminaries section.

Each miner signs the BidHash value, the number of bids received from a unique Coinbase address (NoOfBidder) and time stamp it before relaying it to other full nodes. The majority of the miners should have only one BidHash value from other peers if the message delivery is reliable and most miners are honest. The purpose of the BidHash value is to prevent illegal modification to the bid list by any miner. This BidHash value is also used to reach a consensus on the bidding list so that the correct winner can be determined later. In each block interval, each miner will monitor the number of unique Coinbase addresses received (NoOfBidderPerBidHashValue) for each BidHash value. If more than 50% of the miner received the same BidHash value, then consensus is achieved. It is achievable because more than 50% of the miner is honest and the network is reliable. Miner which calculated a different BidHash value should stop mining immediately and resynchronize it is PoB blockchain on the next block before it can mine again.

5 Security Issue

5.1 Costless Simulation

An adversary can reconstruct the blockchain with a closer or exact bid starting from a known block or genesis block to their advantage since all the past bitcoin blockchain information is available. These problems are called “Costless Simulation” or “Nothing At Stake Problem” which happened mainly in PoS protocol.

Miners must send every PoB block to a TSA server for timestamping to prevent these problems. Hash responds from the TSA server must be included in PoB blockchain. Thus, whatever data created inside the PoB cannot be changed and proven to be created at the specified time. Most TSA server certificate provided by VeriSign, Thawte and other vendors has a validity of one year upon subscription. In every 25,000 PoB block (approximately six months), all previous hashes and TSA timestamp will be

reshashed together and signed with a new TSA certificate. This forms a chain of timestamp signatures. In every 25,000th PoB block, PoB will include this new hash. Thus, the TSA services can prevent any adversaries from creating an alternative history base on existing available data.

Adversaries can hijack a PoB blockchain project by mining the genesis block first before everyone else. Thus, before launching a PoB blockchain, the first ten blocks must be secretly mined with zero incentives to the initial operator. The public announcements should come later. After the 10th block, the public can view the blockchain information to decide whether it is fair to mine or secure to use the blockchain. The initial operator may need to mine with zero incentives for an extended period because miners need some time to come in.

The preliminaries section shows that in each bidding message B_i , the current PoB Block includes the hash of the previous PoB block as an input. Even though an adversary can reconstruct their blockchain starting from a known block, the adversary cannot produce the correct signature for transactions not owned by them. Thus, an adversary cannot recreate the history that includes transactions that do not belong to them.

5.2 Denial of Service Prevention

Bid Transaction Filtering and Bid Cost Revision: As mentioned in Sect. 4.2, there is a bid transaction filtering mechanism to filter out excessive bid submission sent by miners. In case, the miners continue to send an excessive number of bids, the cost per bid is revised upward to reduce the number of eligible miners. Miners who monopolized the current bidding process will have their bid capital tied down as 50% of the bid reward is carried forward as the next block reward. These miners may need mine for an extended period to recover their investment. Thus further securing the PoB blockchain. In order for a miner to continuously monopolize the bidding process, there must exist a lot of funds movement transaction between Coinbase address. Chain Analysis can easily detect these activities and an early alarm can be triggered. All Coin-base addresses that are related to these activities can also be directly identified for possible blacklisting in the future.

Deposit for Bidding: Initially, before any bidding cost revision, when one miner places a bid, \$20 will be charged. If the bid value is revealed correctly, \$9 will be refunded. This indirectly forces miners to reveal their bids and ensure bids consistency. The final \$10 is refunded after miners signed the TraceBid process. The refund will be forfeited if the miner is found to have double-sign their PoB transaction. The refund process is locked for 20 blocks to ensure sufficient time to check for fork and double-signing. This deposit locking mechanism ensures honest miners' actions.

5.3 Fork Prevention

As shown in Fig. 4, an adversary can fork the PoB blockchain at PoB(K-3) by submitting a Bid(K') instead of Bid(K). The adversary then forcibly makes a fork at PoB (K-3) which includes Bid(K'). In this case, the TraceBid process comes into the picture.


| BTC/PoB BLK | | K-3 | K-2 | K-1 |
|-------------|----------|-----------|--|-------------|
| NORMAL | TRACEBID | K-2 | K-1 (HONEST MINER SIGN HERE) | K |
| | REVEAL | K-1 | K | <u>K+1</u> |
| | BID | K | <u>K+1</u> | K+2 |
| ADVERSARIES | TRACEBID | K-2 | K-1 (ADVERSARIES MINER SIGN HERE) | K' |
| | REVEAL | K-1 | K'  | <u>K'+1</u> |
| | BID | K' (FORK) | <u>K'+1</u> | K'+2 |

Fig. 4. PoB fork prevention scenario.

An adversary miner that submitted a different set of bids will have a different BidHash value. Honest miners check the BidHash value they received in Bid(K) stored at PoB(K-3). It is found to be Bid(K) instead of Bid(K'). Thus, at PoB(K-2), honest miners will sign their TraceBid(K-1) with BidReveal(K). Miners who sign on multiple chains will have their deposit forfeited. Since adversaries cannot control more than 50% of the miner from the previous block, it cannot fork the PoB chain beyond PoB(K-3). The fork chain at PoB(K-3) will be orphaned after PoB(K-2) is created. In bitcoin, the longest chain is considered as a valid chain in case of any blockchain fork. In PoB, if a fork happened, the chain with the most number of TraceBid signature is considered as a valid chain.

If the number of TraceBid signatures is precisely 50% for both chain, then each miner must remove all the duplicate transactions from Bid(K) and Bid(K'). In the remaining unique bid list, the bid list with the lowest AcceptBid value is considered as a valid chain. If the AcceptBid value is the same, then the lowest Coinbase address is considered. If there are more than two forks at PoB(K-3), then there exists a dangerous prolong network partition among all miners. This problem is a hard problem even for bitcoin itself. Miners can indirectly detect network partitioning especially when its' discovered that a significant portion of RevealBid or TraceBid is missing in their PoB Block.

6 Conclusion

In this paper, we have described a PoB protocol that can secure the transaction by leveraging on bitcoin PoW. PoB can replace bitcoin PoW if there exists a random source of information that is publicly available, cheap to verify, with sufficient entropy

and periodic. In addition to that, the randomness source must be expensive to reproduce and can output a precise value. So far, we only found bitcoin as the source of randomness.

Using PoB may have some side-effect such as mining on an orphan BTC block. Some remedies are described to remove these side-effects. Many attack scenarios similar to PoS are explored and remedies are suggested. Our PoB protocol does not need any specialized hardware such as Proof of Luck method. PoB protocol does not have any problem with Long-Range Attack and “Nothing At Stake Attack” as compared to Proof of Stake. Thus, we believe that many blockchain applications can use PoB protocol and it is a strong contender to many Proof of Stake and PoW coin. At the same time, PoB is indirectly leveraging on the security provided bitcoin PoW. Thus, it is far more secure than most of the consensus protocol. PoB does not need to coordinate with bitcoin miner to perform any merge-mining. The information given in this paper should be sufficient for a prototype implementation and further exploratory work.

References

1. de Vries, A.: Bitcoin’s growing energy problem. *Joule* **2**(5), 801–805 (2018)
2. Poelstra, A.: On Stake and Consensus (2016). <https://download.wpssoftware.net/bitcoin/pos.pdf>
3. Bentov, I., Gabizon, A., Mizrahi, A.: Cryptocurrencies without proof of work. In: Clark, J., Meiklejohn, S., Ryan, P.Y.A., Wallach, D., Brenner, M., Rohloff, K. (eds.) FC 2016. LNCS, vol. 9604, pp. 142–157. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53357-4_10
4. Snider, M., Samani, K., Jain, T.: Delegated proof of stake: features & tradeoff. *Multicoins Capital* (2018)
5. Bentov, I., Lee, C., Mizrahi, A., Rosenfeld, M.: Proof of activity: extending Bitcoin’s proof of work via proof of stake. *IACR Cryptology ePrint Archive 2014*, p. 452 (2014)
6. Milutinovic, M., He, W., Wu, H., Kanwal, M.: Proof of luck: an efficient blockchain consensus protocol. In: *Proceedings of the 1st Workshop System Software Trusted Execution (SysTEX)*, pp. 1–6 (2016)
7. Colin, L.M.: Nano: a feeless distributed cryptocurrency. *Network*. <https://nano.org/en/whitepaper>
8. Sompolinsky, Y., Zohar, A.: PHANTOM: a scalable BlockDAG protocol. *IACR Cryptology ePrint Archive 2018*, p. 104 (2018)
9. Sompolinsky, Y., Lewenberg, Y., Zohar, A.: Spectre: a fast and scalable cryptocurrency protocol. *IACR Cryptology ePrint Archive 2016*, p. 1159 (2016)
10. Sompolinsky, Y., Zohar, A.: Secure high-rate transaction processing in Bitcoin. In: Böhme, R., Okamoto, T. (eds.) FC 2015. LNCS, vol. 8975, pp. 507–527. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-47854-7_32
11. https://en.bitcoin.it/wiki/Proof_of_burn. Accessed 11 May 2019
12. Sidechains, Drivechains, and RSK 2-Way peg Design. <https://www.rsk.co/noticia/sidechains-drivechains-and-rsk-2-way-peg-design>. Accessed 11 May 2019
13. P4Titan. SlimCoin.: A Peer-to-peer Crypto-Currency with Proof-of-Burn. Mining without powerful hardware, 17 May (2014)
14. https://en.bitcoin.it/wiki/Merged_mining_specification. Accessed 11 May 2019

15. Judmayer, A., Zamyatin, A., Stifter, N., Voyiatzis, A.G., Weippl, E.: Merged mining: curse or cure? In: Garcia-Alfaro, J., Navarro-Arribas, G., Hartenstein, H., Herrera-Joancomartí, J. (eds.) ESORICS/DPM/CBT -2017. LNCS, vol. 10436, pp. 316–333. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67816-0_18
16. [https://en.wikipedia.org/wiki/BLAKE_\(hash_function\)](https://en.wikipedia.org/wiki/BLAKE_(hash_function)). Accessed 11 May 2019
17. https://en.wikipedia.org/wiki/Gossip_protocol. Accessed 11 May 2019
18. Adams, C., Cain, P., Pinkas, D., Zuccherato, R.: Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP). RFC 3161, August 2001
19. Pinkas, D., Pope, N., Ross, J.: Policy Requirements for Time-Stamping Authorities (TSAs), RFC 3628, November 2003
20. <http://bitcoinstats.com/network/propagation>. Accessed 11 May 2019