Effective scheme against 51% Attack on Proof-of-Work Blockchain with History Weighted Information

Xinle Yang, Yang Chen and David X. Chen
MOAC Blockchain Tech Inc.
Palo Alto, CA, USA
{xinle.yang, yang.chen, david.chen}@moac.io

***Abstract*—Proof-of-Work (PoW) is a popular protocol used in Blockchain systems to resolve double-spending problems. However, if an attacker has access to calculation hash power greater than half of the total hash power, this attacker can create a double-spending attack or 51% attack. The cost of creating a 51% attack is surprisingly low if hash power is abundantly available. That posts a great threat to lots of PoW blockchains. We propose a technique to combine history weighted information of miners with the total calculation difficulty to alleviate the 51% attack problem. Analysis indicates that with the new technique, the cost of a traditional attack is increased by two orders of magnitude.**

***Index Terms*—Blockchain, 51% Attack, Mining, Double spending attack**

I. Introduction

Developed by Satoshi Nakamoto in 2009 [1], Bitcoin is the first decentralized public ledger system. Since then, a number of similar blockchain-based cryptocurrencies have emerged. Blockchain is a distributed data processing protocol for retaining a public distributed ledger in a Peer-to-peer (P2P) network. Transaction data is recorded in blocks, and these blocks form a linked list (i.e., chain) of blocks. Each node in the network stores and maintains an entire copy of the ledger without requiring a central authority. In blockchain-based cryptocurrencies, each block contains the hash value of the previous block, making it hard to manipulate the transactions within. Normally, a consensus protocol is used to guarantee the data integrity among the nodes of the blockchain P2P network. There are several different consensus protocols used in different types of blockchains [2].

Proof-of-Work (PoW) is the most commonly used consensus protocol in blockchain-based cryptocurrencies. Major blockchains such as Bitcoin and Ethereum are both using different variety of PoW protocol. In PoW protocol, each node is competing to find a nonce value to produce a hash that meets a certain criteria. The difficulty of calculating such a nonce value can be calculated based on the criteria of the hash value. When such a nonce value is found, a block is generated and broadcasted to the P2P network. Depending on different varieties of protocol, peer nodes always accept the longest chain or the chain with the largest total difficulty repeatedly to continuously expand the blockchain. PoW utilizes this mechanism to determine which node has the right to seal a block. This process is also called mining.

In such a mechanism, a peer node with greater computing speed (or sometimes called hashrate power) can calculate nonce value faster than a peer node with less computing speed and thus has a higher probability of getting the right to seal a new block. However, this mechanism has a drawback. A selfish node with hashrate power higher than those of the rest nodes combined can compromise the blockchain system by causing double spending and selfish mining, etc [3] [4]. This is commonly referred to as a 51% attack. Some studies have proposed ways to avoid such kind of attacks. Eyal and Sirer [5] in 2014 proposed a Two-Phase PoW(2P-PoW) solution preventing the formation of a mining pool with huge hash power. In this solution, the second phase PoW requires signature from the private key of the coinbase address. When the second PoW is sufficiently difficult, pool operators have to give out this private key to the pool miners in order to perform a calculation faster than all the peer nodes. Ruffing et al. [6] in 2015 proposed contracts to penalize attackers attempting a double-spending attack. Solat and Potop-Butucaru proposed ZeroBlock [7] in 2016. The mechanism in ZeroBlock requires a block to be accepted by its peers within certain time interval after the timestamp of the block. Otherwise, the block is expired. This mechanism prevents attacker nodes from selfish mining for a long period of time. J. Bae and H. Lim [8] in 2018 proposed a solution to randomly select a certain group of miners to have the right to mine the next block.

In the present paper, we introduce a new solution by utilizing a historical weighted difficulty to determine the total chain difficulty. In such a modified algorithm, a branch of blockchain has a greater total historical weighted difficulty if the miners of such a branch have a higher coverage rate in previous blocks. We demonstrate that, in reality, such an algorithm can increase the money cost and the time cost of 51% attacks by a factor of at least two orders of magnitude.

II. 51% attack and cost

First, let us review the 51% attack scheme. Assume the current hashrate is p_o, the attacker accumulates a greater hashrate power p_a with $p_a > p_o$, and utilizes this hashrate power to compute a hidden branch B_a. The attacker performs

double spending in two branches. The attacker then reveals hidden longer branch B_a and invalidates all transactions in original branch B_o. The cost to launch such a 51% attack is,

$$Cost = (P * R) * f * t \tag{1}$$

where P is the token price, R is the block reward, f is the frequency of block generation speed, and t stands for the duration of the attack. For many small blockchains, the cost to perform such an attack is only hundreds or thousands of US Dollars [9].

Another factor that makes the Scenario worse is that hashrate could be easily accessible to anyone who can pay the right price. NiceHash [10] provides an open market for hashrate exchange. Anyone can easily pay with cryptocurrency to rent available hashrate to mine for the target blockchain. So an attacker can accumulate significant hashrate in short period of time to exceed the 51% threshold. The attacker can double spend the token through a centralized exchange. The whole process only takes around 50-500 blocks. After that, attacker can release the rented hashrate and walk away with the profit.

Most recently, Ethereum Classic [11] was attacked. Large amount of ETCs were double-spent in attack branch length ranged from 50 to 150.

III. Historical Weighted Difficulty

We propose a technique to calculate the total difficulty of a certain branch with the consideration of the miners addresses existence frequency in the previous blocks. We call this protocol Historical Weighted Difficulty based Proof-of-Work (*HWD*-PoW) protocol. The assumption is that in an honest blockchain branch, miners of new blocks will most likely be the miners who mined the previous blocks, and the distribution will reflect the ratio in history. Furthermore, in a malicious blockchain branch, distribution of miners of new blocks will most likely be controlled by the attackers, which will be different from the regular distribution of miners in the history. Therefore, when history of miners' distribution is considered, one can easily distinguish an honest blockchain branch from a malicious one.

Under the proposed the mechanism, branch with miners of less representation in the previous blocks will earn less weight in the total difficulty calculation. Therefore, to perform a 51% attack, the malicious miners have two choices: either to mine a much longer branch, or to build up miner representation in the previous blocks to build up the credibility.

Now, let us look at how Historical Weight Difficulty scheme works to defend the 51% attack.

(a) First, record each miner's block generation frequency for history window W:

$$r_i = \frac{(blocks\ mined\ by\ node\ i)}{(total\ block\ number\ generated\ in\ window\ W)} \tag{2}$$

where

$$\sum_{i=0}^{n} r_i = 1 \tag{3}$$

(b) Each block is then signed by miner's private key. By doing this, miners will not be able to counterfeit the miner identity.

(c) When a split is detected, Historical Weighted Difficulty (*HWD*) is calculated for each branch at each peer node in the following way: for any unique miner k in the branch b,

$$HWD_b = HW_b * D_b = \sum_{k=1}^{l} r_k * \sum_{k=1}^{l} d_k \tag{4}$$

where r_k is the block generation frequency in the history window, d_k is the difficulty of block k, and l is the branch length.

Please note that only unique miner's generation frequency is counted. If one miner address mined multiple blocks, it is only counted once. This is to discourage single high hashrate miners, thus encouraging the decentralization of mining. It also increase the difficulty of attack.

(d) Each peer node compares two different *HWD*'s from two branches. The branch with greater *HWD* will be selected.

The immediate result is clear: if the attacker just brings in temporary new hashrate power, but the miner of the new branch is relatively new to the system, the r_i of the blocks is very low, and the corresponding *HWD* will be very small compared to the original branch. No peer node in the original branch will switch to the attacker branch. This scheme can easily defend rent-and-attack case.

If the attacker wants to make nodes to switch his fraudulent branch, he will need to produce higher *HWD* value. The attacker needs to mine in the original branch for a while to make itself included in the history. Therefore, when it switches to hidden branch, its *HWD* will be greater.

Suppose p_a is the attacker miners' hashrate, p_o is the original honest miners' hashrate.

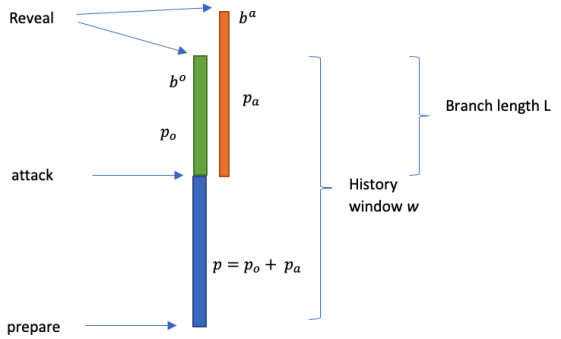

Fig. 1. Attacker tries to attack branch b_o with branch b_a.

The attacker needs to spend hashrate p_a in the original branch for a time duration t. The optimum spending of p_a is to mine for $w/2$ period and then switch p_a to hidden branch at attack point show in Fig. 1. In order for attack to be successful, we have

$$p_a > p_o \tag{5}$$

As a result, the revealed malicious branch b_a will be longer than original branch b_o. At the reveal time, *HWD* of the original branch is,

$$HWD_o = D_o * \sum_{k=0}^{l} r_k^o = D_o * (\frac{1}{2} - \delta) \tag{6}$$

where δ is the minimal marginal difference accepted by peer node.

So, *HWD* of the malicious branch is

$$HWD_a = (\frac{1}{2} + \delta) * \frac{w-l}{w} * D_a \tag{7}$$

Therefore, we have,

$$(\frac{1}{2} + \delta) * \frac{w-l}{w} * D_a > (\frac{1}{2} - \delta) * D_o \tag{8}$$

Because D_a and D_o are close to each other in such a condition, we can simplify the equation to

$$(\frac{1}{2} + \delta) * \frac{w-l}{w} > (\frac{1}{2} - \delta) \tag{9}$$

and we derive to

$$\delta > \frac{l}{4w - 2l} \tag{10}$$

To summarize, the minimal cost of the attacker is to prepare mining for $(w-l)$ duration with hashrate $p_o * (1/2 + l/(4w - 2l))$.

For a typical attack, l needs to be around 50-500 blocks to allow token withdrawal from a token exchange. From below section, the change of miners is not very frequent. We can easily set $w > 1$ month to increase the history weight. With $w = 100,000$ blocks, the robustness against attack is increased by over 100 times.

Although we cannot totally avoid 51% attacks, we can dramatically increase the money cost and time it takes to prepare a potential attack by at least two orders of magnitude. Also, because attacker needs to spend quite a long time to prepare, the attack is much less likely to happen as the long period of time causes a lot of opportunity cost and uncertainty.

IV. Additional improvement schemes

Orthogonal to the above-mentioned *HWD* scheme, additional improvement schemes can be applied to further increase the cost of attack.

The first approach is to post a cap of $r_i < r_c$, which means single miners can never be counted more than r_c even it generates more blocks within the history window. This approach will encourage a more diversified mining pool. At the same time, the miners could purposely split the mining hashrate into multiple miners to make sure each one is below r_c. This will circumvent the diversification requirement. However, this is still useful, as it increase the cost for attackers to maintain multiple miner accounts.

The second approach is to post an overlap requirement between two split branches. To minimize the cost of attack, an attacker will focus its hashrate to the malicious branch. The overlap requires the new branch poses a certain amount of miner overlap with the old branch. Under such a condition, in order for a branch switch to happen, one needs to satisfy not only the *HWD* condition,

$$HWD_a > HWD_o \tag{11}$$

but also, the overlap between two sets of miners need to be greater than s,

$$\{r_i\} \bigcap \{r_k\} > s \tag{12}$$

where r_i are the miners' frequency of original branch, and r_k are the miners' frequency of attacker branch.

Meaning, the system discourages sudden hashrate power switches from one set of miners to another distinct set of miners.

With such an enhanced requirement, if $s = 0.25$, an attacker needs to have three times of the current hashrate within time duration w. This means an attacker needs to keep some hashrate in the original branch, and twice more in the malicious branch to compensate for the effect. Therefore, by introducing such a requirement, we double the time cost and triple the money cost to perform an attack to the original Historical Weighted Difficulty algorithm. For higher s, attacker needs to spend even more hashrate than the minimal case.

V. *HWD* algorithm

In this section, we will present the algorithm to perform *HWD* based branch selection.

Below is the pseudo code to calculate the *HWD*,

Algorithm 1 Calculation of *HWD*

```
function HwdCalculation(W, B) ▷ Where W - array of historic blocks window, B - array of branch blocks
    HW = 0
    d = 0
    w = Length(W)
    l = Length(B)
    Let R[1...l] be new arrays
    for i = 1 to l do ▷ Calculate miner appearance frequency in historic blocks window
        R[i] = 0
        for j = 1 to w do
            if Miner(W[j]) == Miner(B[i]) then
                R[i] += 1
            R[i] /= w
    for k = 1 to Length(B) do ▷ Sum Historic Weight
        HW = HW + R[k]
    for k = 1 to Length(B) do ▷ Sum branch difficulty
        d = d + Diff(B[k])
    HWD = HW * d
```

VI. Real data statistics from Ethereum

We picked a well-known PoW blockchain platform Ethereum. We analyzed block information of first 6,000,000 blocks, which is about 3 years mining history. The first task is to find out the distribution of miners with significant hashrate.

The analysis shows that the miner distribution has a strong correlation with past history. At block #2,000,000, #4,000,000, #6,000,000, the following 360 blocks were analyzed. Each miner's weight in the 360-block and weight in the 2M-block is shown in tables blew:

TABLE I
ETHEREUM BLOCK MINERS FROM 2,000,001 TO 2,000,360

Miner	Weight	Amount	Weight_in_2M
0x2a65.....8226	0.272222	98	0.239315
0x61c8.....0bd9	0.177778	64	0.049251
0xbcdf.....41d1	0.163889	59	0.023924
0xea67.....8ec8	0.116667	42	0.036458
0x4bb9.....1b01	0.063889	23	0.057752
0xa42a.....e84e	0.055556	20	0.001293
0x52bc.....e3b5	0.038889	14	0.147223
0x1a06.....58f1	0.030556	11	0.004212
0x6879.....01da	0.025000	9	0.023600
0xd138.....a31c	0.005556	2	0.000053
0xf3b9.....c2fb	0.005556	2	0.008294
0xa027.....e88f	0.005556	2	0.012287
0x1654.....d5de	0.002778	1	0.001272
0x186a.....b0f2	0.002778	1	0.000001
0x2cb7.....6402	0.002778	1	0.000004
0x30b6.....4e6d	0.002778	1	0.002430
0x40ce.....f821	0.002778	1	0.000954
0x5979.....e584	0.002778	1	0.000116
0x6caf.....a46d	0.002778	1	0.001050
0x7a14.....0b95	0.002778	1	0.001233
0x9148.....a49d	0.002778	1	0.000021
0x94ce.....a2f7	0.002778	1	0.000944
0x9558.....7211	0.002778	1	0.011990
0xadd8.....db02	0.002778	1	0.000039
0xd3d0.....ee9d	0.002778	1	0.001598
0xdc3f.....e455	0.002778	1	0.000340

[a]Total miners' weight from previous 2 million blocks is 62.57%.

TABLE II
ETHEREUM BLOCK MINERS FROM 4,000,001 TO 4,000,360

Miner	Weight	Amount	Weight_in_2M~4M
0x829b.....a830	0.283333	102	0.012240
0xea67.....8ec8	0.236111	85	0.169016
0x1e99.....0341	0.138889	50	0.079728
0xb293.....0347	0.086111	31	0.032167
0x52bc.....e3b5	0.075000	27	0.045849
0x2a65.....8226	0.072222	26	0.167880
0xc0ea.....2949	0.033333	12	0.072130
0x4bb9.....1b01	0.016667	6	0.058859
0xf3b9.....c2fb	0.013889	5	0.015585
0x9435.....7805	0.008333	3	0.003338
0x9633.....a11c	0.008333	3	0.005184
0x73b8.....7fea	0.005556	2	0.007535
0x8727.....87a5	0.005556	2	0.000331
0xa42a.....e84e	0.005556	2	0.033483
0xa4aa.....7f0d	0.005556	2	0.003277
0xa9a9.....51fc	0.002778	1	0.000911
0xa027.....e88f	0.002778	1	0.001355

[a]Total miners' weight from previous 2 million blocks is 70.89%.

The total weight of 360-block miners has a strong correlation with the distribution in the history window, even if the history we used is 2 million (M) blocks long (about one year). The result is shown in Fig. 2. This supports our assumption that the participation rate of honest miners is relatively stable. Therefore, miners' historical weight is a valuable information

TABLE III
ETHEREUM BLOCK MINERS FROM 6,000,001 TO 6,000,360

Miner	Weight	Amount	Weight_in_4M~6M
0xea67.....8ec8	0.322222	116	0.259396
0x5a0b.....9c4c	0.133333	48	0.108651
0x829b.....a830	0.127778	46	0.210382
0x52bc.....e3b5	0.116667	42	0.125527
0xb293.....0347	0.105556	38	0.098562
0xf3b9.....c2fb	0.036111	13	0.023992
0x2a65.....8226	0.027778	10	0.035354
0x1ca4.....be1a	0.019444	7	0.000904
0x52e4.....f13c	0.013889	5	0.007205
0xd958.....4012	0.013889	5	0.000263
0xb75d.....22f5	0.011111	4	0.005698
0xd438.....1807	0.011111	4	0.000594
0x6a7a.....9b1f	0.008333	3	0.008874
0x70ae.....e21d	0.008333	3	0.002358
0x35f6.....738d	0.005556	2	0.000325
0x4a07.....a82b	0.005556	2	0.002980
0xe4bd.....0649	0.005556	2	0.003712
0x0019.....99e8	0.002778	1	0.000784
0x4bb9.....1b01	0.002778	1	0.015724
0x914d.....1dcd	0.002778	1	0.000168
0x92e3.....b549	0.002778	1	0.000822
0x9435.....7805	0.002778	1	0.007071
0xb8f8.....5453	0.002778	1	0.000668
0xcc16.....e610	0.002778	1	0.002227
0xd100.....4fce	0.002778	1	0.000470
0xd380.....636d	0.002778	1	0.000002
0xd9cf.....06e3	0.002778	1	0.000108

[a]Total miners' weight from previous 2 million blocks is 92.28%.

we can utilize to fight against 51% attack. We also observed that correlation of new miners with previous 2M blocks is strengthened, as it indicates that Ethereum mining is going towards centralization.

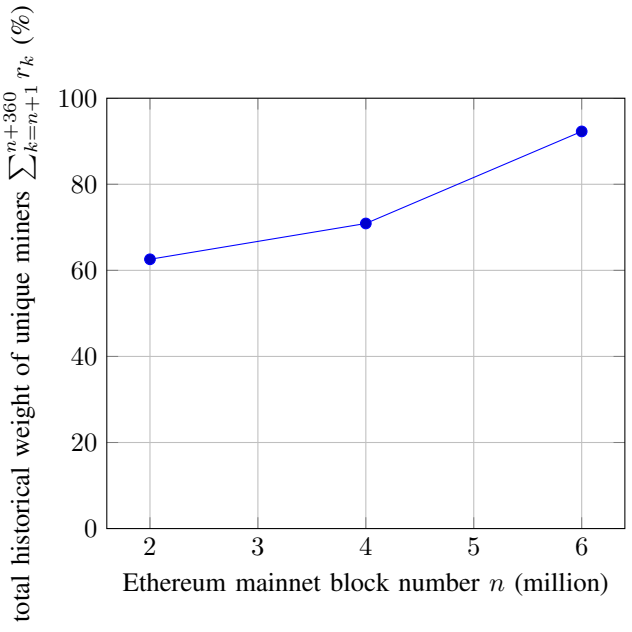

Fig. 2. Total historical weight *HW* of unique miners of the 360 new blocks from previous 2 million blocks.

We further created a simulation to perform 51% attacks at

the point of block #2M, #4M and #6M, and computed the result for window length of 1M and 2M. To make it close to real case, we randomly flagged 51% of miner in certain length back from attack point to be dishonest, which represents the attack preparation period. From the dishonest list, we picked the top 360 miners as the attacking branch generator. If the *HWD* of attacking branch is higher than those of the original branch, we marked it as a successful attack.

In the ideal case, the preparation should be at least longer than the specified window. However, given the correlation of distribution, the preparation period was slightly shorter. We ran the simulation multiple times for each window to get the average accumulated cost. Here we ignored the mining reward halving effect over the time. Result is below Fig. 3.

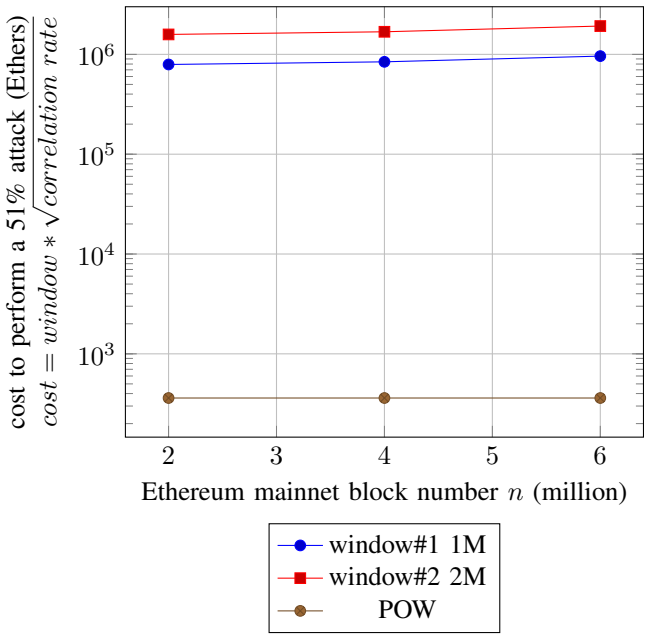

Fig. 3. Comparing the cost of perform a 51% attack, *HWD*-PoW(1M windows, 2M windows) vs. PoW

The result indicates that with *HWD* scheme, the cost of attacking the Ethereum Mainnet blockchain is increased by more than 100 times if we set the window size to 1M or 2M blocks. In real application, the window could be shorter to speed up the process. We suggest window size is at least 100k, making the cost of attack 100x more expensive.

VII. Conclusion

In this paper, we proposed an approach to increase the cost of a successful 51% double-spending attack on Proof-of-Work types of Blockchain protocols. The proposed approach utilizes the frequency rate of miners in history blocks and calculates the total Historical Weighted Difficulty to determine if branch switch is needed. We demonstrated in three real Ethereum Mainnet scenarios that the cost of attack is increased by more than 100 times with *HWD*-PoW scheme.

Our *HWD*-PoW scheme could be applied to all other PoW based blockchains. This can improve smaller blockchain's security drastically with easy integration.

Acknowledgment

This work was supported by the MOAC Foundation and MOAC Blockchain Tech Inc.

References

[1] S. Nakamoto,"Bitcoin: A peer-to-peer electronic cash system," 2008.
[2] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," Int. J. Web Grid Serv., 2016.
[3] M. Conti, S. Kumar E, C. Lal, and S. Ruj, "A survey on security and privacy issues of Bitcoin," arXiv preprint arXiv:1706.00916, 2017.
[4] Eyal, I. and Sirer, E.G. (2014) "Majority is not enough: Bitcoin mining is vulnerable", Proceedings of International Conference on Financial Cryptography and Data Security, Berlin, Heidelberg, pp.436–454.
[5] I. Eyal and E. G. Sirer, "How to disincentivize large Bitcoin mining pools," 2014.
[6] T. Ruffing et al., "Liar, liar, coins on fire!: Penalizing equivocation by loss of Bitcoins," ACM Conf. Comput. Commun. Secur., Oct. 2015.
[7] S. Solat and M. Potop-Butucaru,"ZeroBlock: Preventing selfish mining in Bitcoin," arXiv preprint arXiv:1605.02435, 2016.
[8] J. Bae and H. Lim, "Random Mining Group Selection to Prevent 51% Attacks on Bitcoin," 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), Luxembourg City, 2018, pp. 81-82. doi: 10.1109/DSN-W.2018.00040
[9] PoW 51% attack cost, https://www.crypto51.app
[10] NiceHash - Largest Crypto-Mining Marketplace to sell or purchase hash power, https://www.nicehash.com/
[11] Deep Chain Reorganization Detected on Ethereum Classic (ETC), https://blog.coinbase.com/ethereum-classic-etc-is-currently-being-51-attacked-33be13ce32de