JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 38, 877-892 (2022)
DOI: 10.6688/JISE.202207_38(4).0011

Towards the Rear Attuned Defense Scheme Embark Upon Selfish Mining

SUMMIYA A. PATHAN[1,+] AND NOORULLAH C. SHARIFF[2]
[1,2]*Department of Computer Science and Engineering*
SECAB Institute of Engineering and Technology
Karnataka, 586101 India
E-mail: summias@gmail.com[1]*; cnshariff1@gmail.com*[2]

Bitcoin is the first fully decentralized cryptocurrency. The security features of Bitcoin rely on blockchain technology, which comprises each current as well as the past transactions in the system. In a blockchain when more than one block extends from the same preceding block, the situation is termed as fork or Block race. A selfish miner abuses Bit coin's forks method to gain several unfair rewards. To tackle the issues caused by selfish mining, this work proposes a novel scheme called "Towards the Rear Attuned Defense Scheme". Accordingly, to identify the presence of selfish miners, the Newfangled selfish mining revelation algorithm with a Fork Tenacity Strategy is used here. A weighted fork with a secured fail parameter is established here to compete with the block race caused by the selfish miner. Finally, to ensure reliability, a Slice-up Tender mint consensus protocol is used. Thus, our proposed scheme ensures a better defense against the selfish mining attacks and achieves better time convergence with less electricity fee.

Keywords: cryptocurrency, block race, fork, selfish mining, fork tenacity strategy, weighted fork, secured fail parameter, slice-up tender mint consensus protocol

1. INTRODUCTION

Blockchain technology allows significant transformation for the way parties exchange their digital asset securely without having trust is neither the third party nor a central authority [1]. Owing to the mediator, who involves in the system, the blockchain routinely avoids a single failure and decreases the cost required for the transaction. Moreover, in blockchain technology, security is ensured by a chain of cryptographic puzzles. The cryptographic puzzles are solved by the miner, to earn revenue; thereby a new block of the transaction has been added to the main chain [2]. One of the common blockchain technologies that are currently used by several organizations in the bitcoin blockchain, whereas the blocks are made secured by resolving proof of work [3].

A block in a bitcoin blockchain comprises a timestamp, block reward and number, detailed transaction history, and a hash value that connects the previous block. This makes the bitcoin rank as highest amid the various other crypto-currencies including Lite coin, Ethereum, *etc.* Yet, bitcoin is not an incentive-compatible protocol due to its nature of adapting selfish mining to earn unfair revenue [4].

It doesn't require more mining power for an adversary to obtain unfair revenue. According to [5], the adversary only needs 23, 21% of the hash rate by following the selfish mining strategy to attain the unfair revenue in the Bitcoin network. The perception behind

Received March 2, 2021; revised August 4, 2021; accepted September 24, 2021.
Communicated by Fu-Hau Hsu.
[+] Corresponding author.

selfish mining is to maintain the finding block *i.e.*, new block furtive until their network turns into the longest chain in the Bitcoin network. Moreover, they reveal the block to the public network once if the honest network gets nearer close to the adversary network.

Once the adversary's network becomes the longest chain in a specific bitcoin network, then the adversary can nullify the parties' transaction. It might occur while Bitcoin depends on the tie-breaking protocol. Merely, if two blocks come to the miner, the miner chooses the longest one. Since the revenue from the selfish mining attack is larger than the honest miner, it affects the rational miner to do mining by adapting the selfish mining protocol. Therefore, the strategy is necessary to avert the selfish mining attack in Bitcoin. In [6, 7] the papers discussed the various prior techniques that are used to avoid the Bitcoin selfish mining attack. In addition, [8] provides a detailed study about the various and the most recent types of attacks found within the bitcoin blockchain network. It also provides information about the miners that would spend electricity on solving cryptographic puzzles and they also act as gatekeepers that validate bitcoin transactions of other people. The prior methodologies suffer from high electricity fees due to the existence of dishonest miners.

Furthermore, [9] provides an organized survey about the security as well as privacy features of the Bitcoin network that limits the applications and services of Bitcoin in the real-world scenario. It also states that the introduction of race conditions caused by forking in Bitcoin Blockchain would waste the computational power of honest miners. [10] describes the security concerns caused by the selfish mining tactics underneath the subsistence of mining pools. [11] redesigns the conventional model of the Bitcoin system to tackle the concurrency of the individual mining process, still there exist some complexities related to time convergence and reliability.

Hence to overwhelm the time complexity issues as well the high electricity charge experienced with the prior methodologies when dealing with the selfish mining tactics, in our proposed work, a novel scheme is initiated. It makes use of a newfangled selfish mining algorithm and a fork tenacity strategy. Moreover, the reliability of our proposed scheme is strengthened with the help of a consensus protocol. Rest our work is presented as follows: Initially, in Section 2, the literature work that related to our proposed work is reviewed. It is then followed with Section 3 describing a detailed view of our proposed scheme differentiating our proposed scheme from the conventional methods. Section 4, presents the simulation results along with the comparison made with the existing work to strengthen our novelty, and finally, Section 5 concludes this paper with our findings.

2. RELATED WORKS

Eyal and Sirer [12] described selfish mining in blockchain, which showed that mining protocols are not incentive-compatible, and selfish miners can compromise the system and receive greater rewards than their due shares. To explain some pros of selfish mining to the malicious miners, they used a state machine. The paper used a random selection method during the fork example to discourage the egotistical miner. But, when dealing with the random block selection system, the honest miner is equally likely to lose his block during the fork, and it does not guarantee that the honest miner wins under race conditions.

A Freshness Preferred approach was introduced by Heliman [13], which works by choosing the block with the new timestamp rather than the old blocks. In FP, it selects the

most recent blocks as defined by their timestamps when a node is faced with blocks of an honest miner and a greedy miner. Although this is an efficient method for spotting selfish mining behavior, the flow of information in a peer-to-peer network is not always fair and unexpected delays in the spread of honest blocks benefit the selfish miner.

Solat *et al.* [14] implemented the Zero block, where miners were compelled to discharge their blocks within an anticipated time interval. If the miners rescind their blocks for selfish mining and don't transmit them within the anticipated time interval, the dummy blocks are generated by the peers in the network, which are then attached to their blockchains. However, when the difficulty parameter was a constant, then for a Zero block it is difficult to uphold the network with varied hash rates. Anticipated block time may result in elevated variability due to changes in the hash frequency, which may invalidate valid blocks under Zero block.

Bahack [15] suggested a fork-punishment rule. Here, with that rule, the first miner who includes a block for k instance in the blockchain had received only half of the revoked benefits. Though, the honest miners who suffered from the collateral damage of this defense were supposed to form another kind of attack.

Shultz [16] suggested that for each solved block, a particular number of dummy blocks are assigned, to prove that the block had been revealed and was witnessed by the public bitcoin blockchain network soon before the miners were ready to work on it. However, they did not provide a mechanism to evaluate whether the number of proofs is adequate to continue working. Neither did they mention how to prevent the selfish miner from generating a dominant number of proofs and releasing them when necessary. In addition, these three defenses require fundamental changes on the block legitimacy and reward allocation policy. Accordingly, the network participants who were not upgraded their clients were unable to comprehend the new protocol.

J. Göbel [17] deals with testing a blockchain, which is affected by selfish mining. For that, initially, they make use of a Markov model to reveal the block hiding strategies. A spatial poison process model was used to learn the mining of a block by the honest community. Finally, a discrete event simulation was used to analyze the propagation delay.

C. Grunspan [18] reviewed the selfish mining tactic in a bitcoin network and compared their profitability over honest mining. They developed a rigorous profitability model for repetition games. Moreover, martingale's techniques and the Doob Stopping Time theorem were used to calculate the anticipated period of the attack cycles. To make the selfish mining tactic profitable, a difficulty adjustment algorithm was used. Thus, with this strategy, the anticipated time before profit for the selfish miner has been calculated.

S. Solat [19] deploys a timestamp-free technique in zero blocks, which in turn exploited the Poisson nature of PoW (Proof of Work) and analyze the nature of bit coin's propagation information. It has been established to prevent the block withholding attack in a bitcoin network. A recommended interval of 60 seconds is given for the publicly available and unpredictable timestamp. A miner prefers the block whose timestamp is fresher when two competing blocks are obtained within 120 seconds. Yet, the emergence of an additional trusted party is incompatible with the decentralized ethos of bitcoin. Finally, it was noted that the tie-breaking defense laws were not applicable when the private chain is longer than the public chain, making the defenses ineffective against resourceful assailants.

R. Zhang [20] discussed the mining algorithm along with the in-block broadcasting mechanism. It is then followed by the main chain selection policy that was used to reduce

the computation as well as communication overhead. A block/in-block used here refers to the hash values of its direct in-predecessors instead of repeating all transactions when there is no conflicting transaction.

R. Zhang [21] This paper develops a proposal for a backward-compatible protection method that outperforms the best defense previously available. Our policy for resolving forks ignores blocks that are not even forked. blocks that integrate links to competing blocks of their predecessors are valued and published on time. As a result, there is a block that is retained. Contributes to neither or both until a competing block is published. As a result, it provides no advantage in the block race.

However, all the prior methodologies fail to tackle the major concerns faced with the selfish mining attack in a bitcoin network. Hence to compete with the time convergence as well as the high electricity fee, there must be a need for innovation in the field of bitcoin network dealing with selfish mining is crucial.

3. NOVEL REAR ATTUNED DEFENSE SCHEME

Selfish mining is a well-known vulnerability in blockchain exploited by miners to steal block rewards. Bitcoin, a peer-to-peer electronic currency is initiated to assist transactions beyond the conventional financial system. However, bitcoin is not an incentive-compatible protocol. This is because; the selfish mining tactic allows a selfish miner to obtain the unfair revenue rewards by hiding the newly generated blocks from the public chain and create a fork generating a centralized chain. Once the threshold and the length of the centralized chain owned by the selfish miner exceeds, the selfish miner unveils the private chain to the public. Once the chain becomes visible the honest miner abandons its chain and adds the private chain owned by the selfish miner.

This would create congestion as well the chance for becoming a straggler. All these issues make the blockchain or bitcoin transaction a time-consuming process with a high electricity fee. To deal with all these concerns the prior defensive methodologies, such as tie-breaking defense, *etc.* are initialized. However, they are unable to defend against resourceful attackers and are less effective when the selfish chain is longer than the public chain results in high time convergence, congestion, and high electricity fee. Anyhow, the malicious selfish miner damages the decentralized structure of the Bitcoin with arise in the propagation time ends with poor reliability. Moreover, the status of each transaction is to be monitor, but still, it is a challenging process.

Thus, to tackle all the aforementioned concerns, a novel scheme called “Towards the Rear Attuned Defense Scheme” is proposed in our work. With this proposed scheme, we categorized the stages of bitcoin transaction into two phases, which include: detection and avoidance of selfish mining. In the detection phase, to detect the selfish miner, we make use of twofold detection assumptions using the Newfangled selfish mining revelation algorithm and Fork Tenacity Strategy. Here the Newfangled selfish mining revelation algorithm works based on analyzing the behavior of blocks. The term behavior represents expected transaction confirmation height and blocks publishing height with the idea of truth state. The expected transaction confirmation height is calculated based on the transaction size, mining fee, and size of the memory pool. Once the behavior is analyzed the honest miner will be assigned to check whether the average expected confirmation height of all

transactions in the target block will be equal to the actual block height. If the height remains equal, then it is assumed as there is no selfish miner in the blockchain else if the height differs, then the presence of a selfish miner is detected.

In addition to this, the fork tenacity strategy is used in our proposed scheme to detect the selfish miner to tackle the issues such as high electricity fees and time convergence even if the blockchain is too long. This is made with a weighted fork using a secured fail parameter; here we initiate a timestamp to generate a new block so that the honest miner can generate a duplicate or competing block even if the selfish miner established a new centralized blockchain by hiding the newly created blocks. This strategy puts the selfish miner into a dilemma. *i.e.*, if the selfish miner keeps the block as surreptitious even after a competing block is released, the secret block doesn't accord to the weight of its chain. If the secret block is released along with the competing block, the subsequent honest block obtains a higher weight by setting proof of having a glance at this block. Thus with both these scenarios, the secret block doesn't facilitate the selfish miner to win the block race. Thus with this strategy, the issue faced with tie-breaking defense is overwhelmed. Finally, to avoid selfish miners a consensus algorithm is utilized to rank the blocks based on the weighted fork and the reliability of our scheme is enhanced with finite state automata.

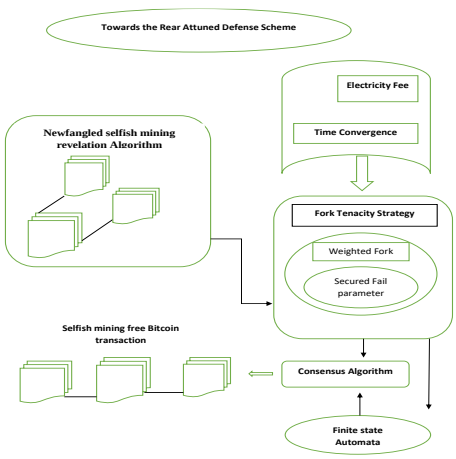

Fig. 1. Block diagram of the proposed method.

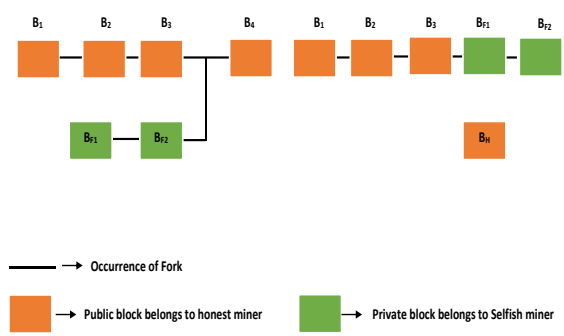

Fig. 2. Selfish mining attack.

3.1 Newfangled Selfish Mining Revelation Algorithm

The time complexity issues as well the high electricity charge experienced with the prior methodologies when dealing with the selfish mining tactics, in our proposed work, a novel scheme is initiated. Where is, the Newfangled selfish mining revelation algorithm is analyzed based on block behavior. With the concept of truth state, the term behavior indicates expected transaction confirmation height and blocks publishing height. The transaction size, mining fee, and memory pool size are used to compute the expected transaction confirmation height. Moreover, the primary aim of the selfish miner is to compute and exempt valid blocks in a private chain to create a fork against an honest miner. The attacker would therefore want the network to move to its longer private chain and to discard the

honest miner's block.

The attacker would like his private chain to be at least one block longer than the main blockchain to persuade the network of a longer proof-of-work and persuade them to move for it to happen. The honest miner, on the other hand, follows traditional mining methods and prioritizes transactions based on their mining fees. To receive both block and fee incentives, He will have to try to achieve as many transactions in the block as feasible. In addition, the honest miner does not withhold his block and makes computer-based transmission to the network on time.

The selfish miner creates two blocks here, BF1 and BF2, and forks the central blockchain to legitimize the block BH of the honest miner. The attack erases 50 percent of the Bit coin's hash power from Nice Hash for 10 minutes. Two rounds match the attack chain. The attacker quantifies the first block, BF1 in the first round, using his hashing power. It then withholds the block and notices that the network accepts the truthful miner's block BH.

The attacker uses the leased hash power in the second round to evaluate the next BF2 block before anyone else on the network. If the block is accumulated, as seen in Fig. 2, the attacker forks the main blockchain with his private chain. As a result, the network switches over to the selfish miner's forked private chain and discards the honest miner's block. In the attack, the selfish miner progresses and wins more rewards than the cost of the attack.

For the blocks at the fork moment, a notion of "truth state" is created to solve certain types of attacks, which in turn are utilized to recognize the actions of the selfish miner. In the data structure of a transaction, we add the parameter of "expected confirmation height". The height of the block in the blockchain is the index number denoting its location in the chain. The new block applies a factor of 1 to the height of the row.

The expected confirmation height is the amount of the index of the future block where the transaction is likely to be mined, depending on the size of the transaction, the mining fee, and the memory pool size. The mining fee and the transaction size give the transaction a preference factor. The preference factor demonstrates a miner's motivation to choose the transaction for his block. Miners are more likely to priorities the transaction for their block if the mining fee is large and the transaction size is small. A repository that caches unconfirmed transactions is the memory pool in the blockchain. If the memory pool size is high, a transaction backlog is created and pending transactions have to wait for mining to take place. A flowchart depicting the steps in a newfangled selfish mining revelation algorithm is defined in Fig. 3.

Algorithm 1: Newfangled Selfish mining revelation Algorithm

Step 1: Start
Step 2: Initialize the value of T_{satate}, G_{state}, and $_{Ivalue}$
Step 3: if I_{value} is less than T_{state} or G_{state} reject BF1
Step 4: Else accept BF1
Step 5: Find product of I_{value} and T_{state}
Step 6: Calculate sum of A_{rev} and P
Step 7: If A_{rev} is greater then 0 accept BF1
Step 8: Else reject BF1
Step 9: End

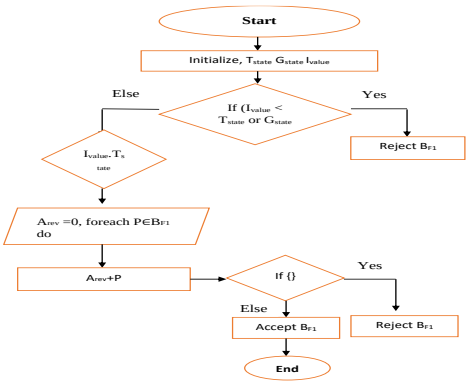

Fig. 3. Flowchart for newfangled selfish mining revelation algorithm.

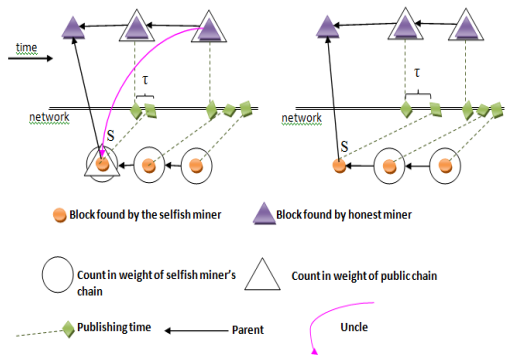

Fig. 4. Multi-choices offered to a selfish miner during selfish mining.

As follows, an adaptive intruder can still bypass detection; (1) Include transactions with the predicted future block time in the first block to minimize the gap between the height of the last block and the expected mean confirmation height of the first block transactions; (2) To obtain a better truth state than the honest miner, have less or no transactions in each block (T State_XSn and T State < G State).

To address the issues, we propose a future state parameter G State in our algorithm, which evaluates if the selfish miner intended to include transactions corresponding to a future block in the current block. If the selfish miner performs this, the G State value would be smaller than zero, revealing the existence of transactions in each block. Also, Fig. 3 illustrates if the G State value is less than zero, the private chain is denied. To address the second strategy, we compare the number of transactions in the blocks of honest and greedy miners. If the average number of transactions in the selfish miner's blocks is less than the average number of transactions in the honest miner's block, it will be revealed that the selfish miner attempted to obtain a falsely higher state of reality by publishing empty blocks. When such fraudulent behavior is detected, our algorithm refuses the private chain.

3.2 Fork Tenacity Strategy

The fork tenacity strategy is used in our proposed scheme to detect the selfish miner to tackle the issues such as high electricity fees and time convergence even if the Blockchain is too long. This is made with a weighted fork using a secured fail parameter *K*; In a block race, the current working condition is that once if the length of a chain is greater than the other chains that are less than *K* blocks, then the miner has mine the longest chain. If the length of both the public and private chains is the same, then the miner will choose the chain with the largest weight. If the largest weight is achieved by multiple chains simultaneously, the miner selects one amid them randomly.

3.2.1 Secured fail parameter

The hashes of all of a miner's uncles should be included in the working block. Also, our weighted fork-resolving policy (FRP). Since two competing chains always have a common prefix, only consider the last part of the chains in our weight calculation, excluding

the common prefix. The number of in time blocks plus the number of in time uncle hashes embedded in these in time blocks is the weight of a chain from the perspective of a miner. A miner's local perspective is used to determine if a block is in time. In the same mining sequence, Fig. 4 depicts two alternative selfish miner decisions. Both chains on the left graph have the same weight of three. Even though the honest miners only have two blocks, the second block has the hash of its uncle *S*, who is published on time. Because the selfish miner does not broadcast *S* in time, both chains are weighted two in the right graph. FRP that is weighted. In a block race, a miner mines on the longest chain if it is longer by at least *k* blocks; otherwise, the miner picks the chain with the biggest weight; and if the largest weight is obtained by multiple chains at the same time, the miner chooses one at random. Here, the parameter *k* is introduced as a fail-safe option. When $k = 1$, our defense is reduced to a tie-breaking defense: in the case of a tie, honest miners would mine on the heavier chain. The first rule of weighted FRP does not applicable when $k = \infty$. It can be seen from Fig. 4, where the weighted FRP puts the selfish miner into a dilemma.

The selfish miner has two choices when an adversary of the first secret block *S* is released: if it publishes *S*, it will be an uncle of the next honest block; if not, honest miners will render it as a late block. *S* could not only add to the weight of the selfish chain in any way. Moreover, since the second selfish block is mined before the first honest block, it is unlikely for this uncle's hash to be embedded before it. The latter block is therefore guaranteed to only add to the equal chain. Consequently, our protection lowers the incentive of the selfish miner to hold back an exposed stone. This security is perfectly decentralized. Here, backward compatibility retains the current rules of block validity and makes a smooth transition to reward distribution policy; non-miners who are unable to improve their customers still seem to be compliant. To bring our security into motion, miners and the most publicly reachable network members need to change.

A state is represented as a 6-tuple (B_H, B_M, $Diff_w$, luck, last, published). B_H and B_M denote the total length of the honest and selfish chain, respectively. $Diff_w$ is defined as the weight difference between two chains.

$$Diff_w = W_H - W_M \tag{1}$$

The Boolean value of luck indicates whether an honest uncle has a hidden, non-late block. This block is what we refer to as the lucky block. There should be at most one lucky block because if there are two, it will require the first lucky block to be released or turn it into a late block by the uncle with greater height. There are multiple alternative final values: H or M, representing the block miner that was mined in the last stage. Finally, the number of selfish blocks released denotes released.

Fig. 5 describes the states as (2, 3, 2, 1, s, 1) during the block race. The lucky block is the last selfish block owing to the reason thatS2 is already late. During this state, a hide action releases no more block; selecting even releasesS2, the ensuing temporary state earlier than mining is (2, 3, 2, 1, s, 2); releasing the entire chain has matched, the ensuing state earlier than mining is (2, 3, 0, 0, s, 3). The luck value is updated to 0 because the lucky block is no longer secret. Thus, with both these scenarios, the secret block does not help the selfish miner to win the block race. Thus, with this strategy, the issue faced with tie-breaking defense is overwhelmed. Moreover, the Byzantine Generals Problem (BGP) is mostly solved by consensus algorithms in blockchain. BGP is a well-known problem in computer science that deals with distributed system consensus. Through the case of block-

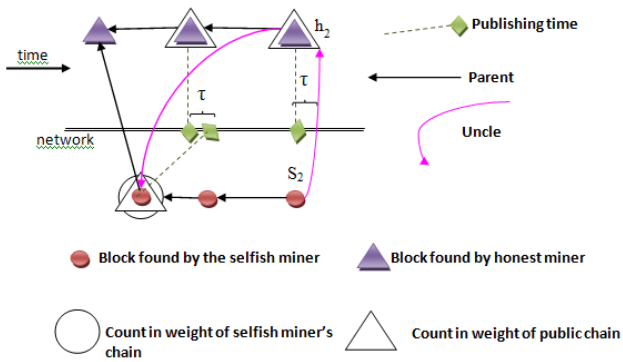

Fig. 5. Block race.

chain systems, the entire network must agree on every message (transaction) transmitted in the network, even if some nodes (blockchain agents) are corrupt or malicious.

Finally, a consensus algorithm along with finite state automata is used here in our proposed work to ensure the reliability of our proposed work against the security risk faced by the bitcoin network. Accordingly, Slice-up Tender mint consensus protocol is used which combines both the work nature of a consensus algorithm and finite-state automation.

Thus, our proposed framework effectively transacts the bitcoin thereby avoids the occurrence of selfish mining with less electricity fee and with less.

4. RESULTS AND DISCUSSIONS

This section ensures the efficiency of our proposed work by analyzing the results and comparing those proposed results with the prior methodologies.

4.1 System Specification

OS: Windows 7 64 bit
Processor: Intel(R) Pentium (R) CPU G2030 @ 3.00GHz
RAM: 4.00 GB
Software: Python

4.2 Performance Analysis

Here in our proposed work, the selfish miner has been detected and avoided using the Newfangled selfish mining revelation algorithm and Fork Tenacity Strategy respectively. For that, the behaviors of the blocks are analyzed. the average number of blockchain splits per 24 hours $b(t)$ as a function of communication delay t, averaged throughout all nodes in the network, the time delay varies from 1 millisecond to 100 seconds. The axes are logarithmic on both sides. The average split rate was almost linearly proportional to the average communication delay when a straight line was fitted to the log-log plot, yielding $b(t) = 0.2508t^{0.9695}$. This section ensures and enhances the performance of our proposed scheme using analyzing our proposed scheme using various performance metrics, some of them are exposed as follows,

From Table 1, it is clear that when the expected communication delay was 10 seconds, on average 2.34 splits are observed per 24 hours. As the average communication delay increased, the number of splits increased and the time until the splits were resolved. Let us consider, during the average communication delay of 0.001012 second, the fork exhibited in our proposed scheme is 0.001015, similarly, for 0.005968seconds the average of Bitcoin blockchain splits occurs is about 0.005119 and so on.

Table 1. Average number of bitcoin blockchain splits per 24 hours.

The average number of blockchain splits per 24 hours	Average communication delay (seconds)
0.001015	0.001012
0.005968	0.005119
0.007258	0.005112
0.01099	0.007848
0.014112	0.009898
0.017233	0.011948
0.020355	0.013998
0.023476	0.016048
0.055975	0.038097
0.097219	0.094766
0.138463	0.151435
0.179707	0.208104
0.220951	0.264773
0.262195	0.321442
0.303439	0.378111
0.344683	0.413478
0.385927	0.491449
1.02312	0.521853
2.78125	0.932859
4.54213	1.355327
6.30135	1.772064
8.01614	2.188801
9.82145	2.605538
11.5815	3.022275
13.3455	3.439012

Table 2. Average revenue earned by miners per hour.

Hash rate	Revenue per miner (bitcoins/hour) earned by a dishonest (selfish) miner	Revenue per miner (bitcoins/ hour) earned by an honest miner
0.151487	0.095745	0.14728
0.201673	0.108511	0.146325
0.252788	0.118997	0.142657
0.301115	0.126748	0.137627
0.351301	0.132675	0.130334
0.398699	0.136322	0.123488
0.451673	0.140881	0.112567
0.499071	0.144529	0.101193

Fig. 6, describes the average number of bitcoin blockchain splits within 24 hours, which has been analyzed through examing the average communication delay. Here the communication node is averaged over all the nodes in the network. With our proposed

work, the detection of the selfish miner is made by examining the number of forks in terms of communication delay.

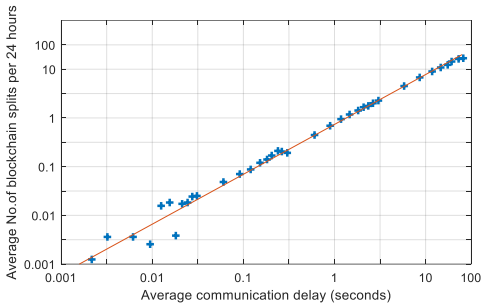

Fig. 6. Average number of bitcoin blockchain splits per 24 hours.

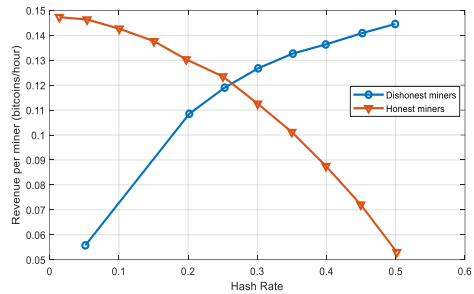

Fig. 7. Average revenue earned by miners per hour.

Fig. 7, and Table 2 reveal that with the rise in the number of dishonest miners, the honest miners have earned under the expected average of 0.15 bitcoins per hour. This enables an affordable way for the honest miners to determine the existence of a pool of miners implementing the selfish-mine strategy.

Table 3. Average block mining rate.

Hash rate	Total	Total confirmed	Honest confirmed	Dishonest confirmed	Honest fair share	Dishonest fair share
0	6	6	6	0	6	0
0.05	6	5.6	5.5	0.054573	5.678764	0.3
0.1	6	5.4	5.06	0.280252	5.340272	0.6
0.15	6	5.1	4.6	0.557215	5.053171	0.9
0.2	6	5	4.1	0.851273	4.800151	1.2
0.25	6	4.8	3.6	1.162317	4.427793	1.5
0.3	6	4.6	3.18	1.542008	4.191437	1.8
0.35	6	4.4	2.62	1.870201	3.887619	2.1
0.4	6	4.2	2.06	2.198285	3.582884	2.4
0.45	6	4.09	1.5	2.54368	3.295621	2.7
0.5	6	4	1	3	3	3

Fig. 8 and Table 3 describe this by exhibiting the performance of both the dishonest pool and the honest miners in terms of the number of blocks they mined that end up in the main branch. This reveals the information about the average number of blocks mined per hour by the pool, by the honest miners. Moreover, a constant say 6 blocks per hour as an average block mining rate. From the above figure, it is clear that whenever a pool gets started to implement a selfish mine, then the pool as well as the honest miners are worse off. Here the total number of blocks is always less than the number that would have been incorporated if dishonest mining were not present.

Fig. 9 reveals the time consumption of our proposed model during the process of the transaction. From this graphical representation, it is clear that the time consumption required to transact the bitcoin is gradually decreased.

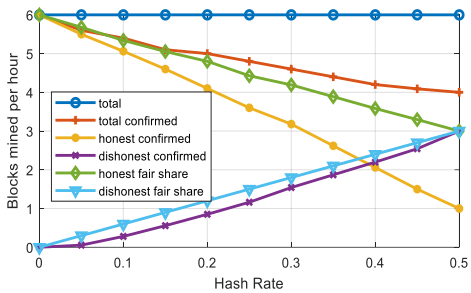

Fig. 8. Average block mining rate.

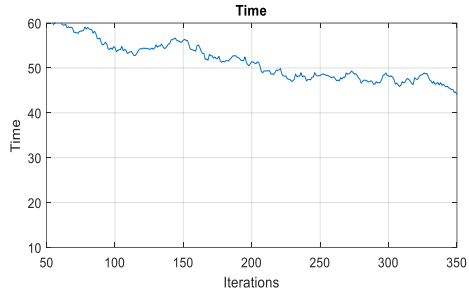

Fig. 9. Performance in terms of time consumption.

Fig. 10 states the performance of the proposed framework regarding the number of bitcoin transactions performed by the miners. During the time, 150-200 minutes, the number of a bitcoin transaction is seemed to be low say between 400-500 numbers of transactions. But, the number of transactions reaches its maximum during the time interval between 250-300 minutes.

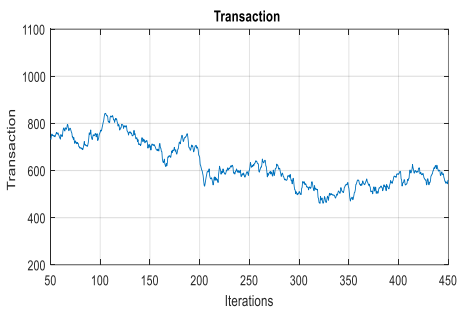

Fig. 10. Performance in terms of No. of transaction.

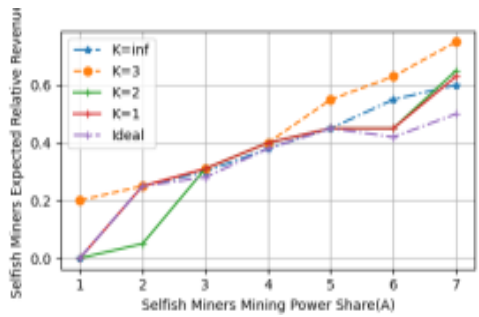

Fig. 11. Relative revenue of the selfish miner within our defense.

In all four settings of Fig. 11, the profit threshold, minimum α to gain unfair rewards, is larger than 0.25. The relative revenue for $\alpha = 0.48$ is 0.764, 0.684, 0.642, 0.622 when $k = 1, 2, 3$ and ∞, respectively. The effectiveness of our defense increases as k grows.

An interesting result is that when $k = 1$, our defense can prevent a malicious miner with more than 50% of mining power from taking over the network. In Fig. 4, the selfish miner with 55% of mining power only earns 76.3% of block rewards. When blockchain integrity is more important than partition recovery time, this variant of our protocol can be useful.

4.3 Performance Comparison

This section describes the various performance of the proposed method comparing with the results of previous methodologies and depicting their results based on various metrics. For the comparison, the optimal selfish mining approach with no defense is used as the baseline. Apart from our defenses, others are used such as uniform tie-breaking, Optimal tie-breaking, and Ideal. Also, that's an imaginary defense known as optimal tie-breaking, in which the selfish miner loses every tie. This defense, in which timestamps are issued with unlimited granularity, can be considered the strongest form of freshness pre-

ferred. From Fig. 12, the relative revenue for uniform tiebreaking and optimal tie-breaking when $\alpha = 0.48$ is 0.837 and 0.731, respectively. Where α is revenue. The numbers become 0.891 and 0.831 if we set the truncating threshold to 160. The difference is for the reason that the block races with a resourceful attacker typically preceding for dozens of blocks in these defenses. Neither defense has any effect for $\alpha > 0.5$. Our defense has the best performance for all α values except when $\alpha = 0.3$ and 0.35. The performance of our defense can be boosted by together with a trusted timestamp server or using the local time to discover potential selfish miner's blocks, however, we gave up these ideas to maintain the decentralized nature of Bitcoin and avoid opening new attack vectors such as the time jacking attack. Moreover, optimal tie-breaking is just imaginary.

Fig. 13 demonstrates this by comparing the attacker's optimal revenue under the uniform tie-breaking protocol with the optimal revenue under the original protocol.

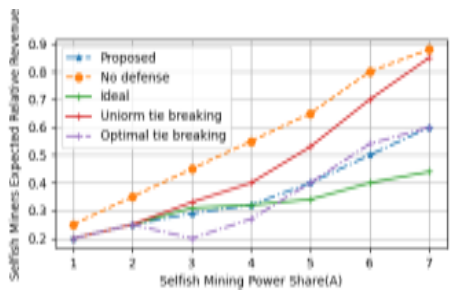

Fig. 12. Comparison with other defense models.

Table 4. Comparison exist between different methodologies in terms of attackers optimal revenue.

Fraction of hash rate	No defense	Ideal	Uniform tie-breaking	Optimal tie-breaking	Proposed
0.2	0.2	0.2	0.197714	0.2	0.2
0.25	0.25	0.257517	0.250675	0.230675	0.26
0.3	0.3	0.337911	0.31963	0.29963	0.35
0.35	0.35	0.452576	0.429719	0.409719	0.48
0.4	0.4	0.59696	0.571822	0.551822	0.62
0.45	0.45	0.775621	0.761926	0.741926	0.78
0.5	0.5	0.88781	0.956533	0.936533	0.97

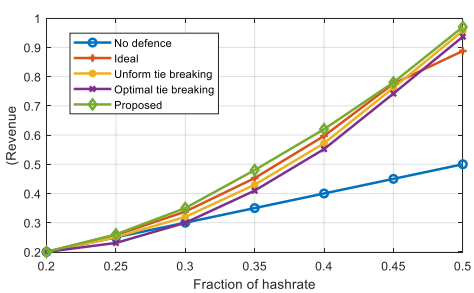

Fig. 13. Comparison exist between different methodologies in terms of attackers optimal revenue.

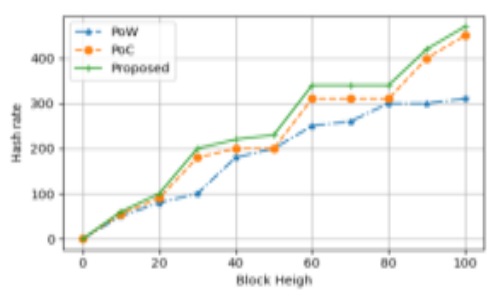

Fig. 14. Comparison of Block height with proposed techniques.

Fig. 14 depicts the block height of data is compared with the hash rate of the various previously proposed techniques. From the graph, it is clear that the block height of the proposed output achieves 91% which is 8% higher than the existing output when compared with baseline, PoW and PoC.

Fig. 15 depicts the selfish pools Annual Recurring Revenue (ARR) of data is compared with the hash power of the various previously proposed techniques. From the graph, it is clear that the selfish pools ARR of the proposed output achieves 0.79 which is higher than the existing output when compared with baseline, Honest mining, Honest and selfish, and Frngs model.

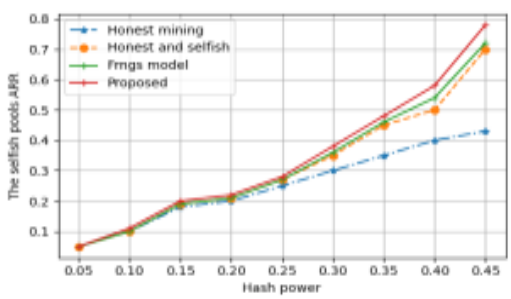

Fig. 15. Comparison of the selfish pools ARR.

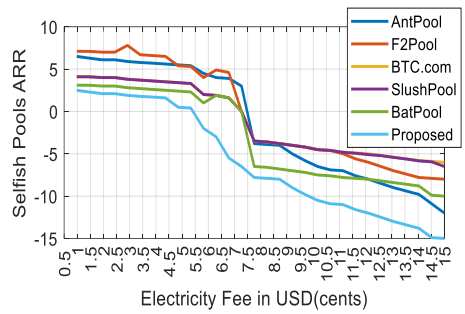

Fig. 16. Comparison of the Electricity fee.

The above graph 16 depicts the selfish pools Annual Recurring Revenue (ARR) of data is compared with the Electricity fee of the various previously proposed techniques. The electricity fee is reduced as compared to the existing techniques. From the graph, it is clear that the selfish pools ARR of the proposed output achieves 14.5 which is higher than the existing output when compared with baseline, AntPool, F2 pool, BTC.com, Slush pool, and Bat pool.

Thus, with our proposed framework, the scalability issues are successfully overwhelmed and at last, the scalability of the Bitcoin network gets increased with less propagation time, handles fork with high throughput and reduced latency.

5. CONCLUSION

In our proposed work "Towards the Rear Attuned Defense Scheme", the issues caused by selfish mining are successfully tackled with the aid of the Newfangled selfish mining revelation algorithm with a Fork Tenacity Strategy. The establishment of the secured fail parameter makes the scheme compete with the block race caused by the selfish miner. Finally, the scheme reliability is achieved with the help of a Slice-up Tender mint consensus protocol. Thus, our proposed scheme avoids the selfish mining attacks, thereby minimizing the process of earning unfair rewards by a selfish miner and achieves better time convergence with less electricity fee.

REFERENCES

1. V. Gramoli, "From blockchain consensus back to byzantine consensus," *Future Gen-*

eration Computer Systems, Vol. 107, 2020, pp. 760-769.
2. S. Rahmadika and K. H. Rhee, "Blockchain technology for providing an architecture model of decentralized personal health information," *International Journal of Engineering Business Management*, Vol. 10, 2018, pp. 1-12.
3. N. Z. Aitzhan and D. Svetinovic, "Security and privacy in decentralized energy trading through multi signatures, block chain and anonymous messaging streams," *IEEE Transactions on Dependable and Secure Computing*, 2016, p. 1.
4. S. Zhu, W. Li, H. Li, C. Hu, and Z. Cai, "A survey: Reward distribution mechanisms and withholding attacks in Bitcoin pool mining," *Mathematical Foundations of Computing*, Vol. 1, 2018, pp. 393-414.
5. A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in *Proceedings of the 20th International Conference on Financial Crypto*, Vol. 9603, 2016, pp. 515-532.
6. A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in *Proceedings of International Conference on Financial Cryptography and Data Security*, 2016, pp. 515-532.
7. Y. Sompolinsky and A. Zohar, "Secure high-rate transaction processing in bitcoin," in *Proceedings of International Conference on Financial Cryptography and Data Security*, 2015, pp. 507-527.
8. N. T. Courtois and L. Bahack, "On subversive miner strategies and block withholding attack in bitcoin digital currency," *arXiv Preprint*, 2014, arXiv:1402.1718.
9. M. Conti, E. S. Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of bitcoin," *IEEE Communications Surveys & Tutorials*, Vol. 20, 2018, pp. 3416-3452.
10. S. Lee and S. Kim, "Detective mining: Selfish mining becomes unrealistic under mining pool environment," *IACR Cryptology ePrint Archive*, 2019, No. 486.
11. T. Leelavimolsilp, L. Tran-Thanh, and S. Stein, "On the preliminary investigation of selfish mining strategy with multiple selfish miners", *arXiv Preprint*, 2018, arXiv: 1802.02218.
12. I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *Financial Cryptography and Data Security*, 2014, pp. 436-454.
13. E. Heilman, "One weird trick to stop selfish miners: Fresh bitcoins, A solution for the honest miner (poster abstract)," in *Financial Cryptography and Data Security Workshops*, 2014, pp. 161-162.
14. S. Solat and M. Potop-Butucaru, "Zero block: Preventing selfish mining in bitcoin," *CoRR*, 2016, abs/1605.02435.
15. L. Bahack, "Theoretical Bitcoin attacks with less than half of the computational power (draft)," *arXiv Preprint*, 2013, arXiv:1312.7013.
B. L. Shultz, "Certification of witness: Mitigating blockchain fork attacks," MS Thesis, Department of Mathematics, Columbia University, NY, 2015.
16. J. Göbel, H. P. Keeler, A. E. Krzesinski, and P. G. Taylor, "Bitcoin blockchain dynamics: The selfish-mine strategy in the presence of propagation delay," *Performance Evaluation*, Vol. 104, 2017, pp. 23-41.
17. C. Grunspan and R. Pérez-Marco, "On profitability of selfish mining," *arXiv Preprint*, 2018, arXiv:1805.08281.
18. S. Solat and M. Potop-Butucaru, "Zeroblock: Timestamp-free prevention of block withholding attack in bitcoin," *arXiv Preprint*, 2016, arXiv:1605.02435.

19. R. Zhang and B. Preneel, "Broadcasting intermediate blocks as a defense mechanism against selfish-mine in bitcoin," *IACR Cryptology ePrint Archive*, 2015, Report No. 2015/518.
20. R. Zhang and B. Preneel, "Publish or perish: A backward-compatible defense against selfish mining in bitcoin," in *Proceedings of Cryptographers' Track at the RSA Conference*, 2017, pp. 277-292.

Summiya A. Pathan presently works as Assistant Professor in Department of Computer Science and Engineering, SECAB Institute of Engineering and Technology, Vijayapura, Karnataka, India since 2009. Her area of interest involves wireless sensor networks, smart city, RFID in IOT and blockchain. Later she perceived her Ph.D. in Bitcoin as Blockchain.

Noorullah C. Shariff presently serving as Professor & Head, Department of Electronics and Communication Engineering, SECABITE, has an experience of 34 years teaching experience in the Department of E&CE and Computer Science. His research interests include information security and risk management, cloud computing, IOT, big data and data mining. He is a member of ISTE, IAENG, ISRD, ISOI.