A Preference-Based Committee Member Auction Consensus Model in Blockchain

Akshaya Mathur, Masoud Barati, Rajiv Ranjan
School of Computing
Newcastle University, UK
{a.mathur2, masoud.barati, raj.ranjan}@newcastle.ac.uk

***Abstract*—Blockchain has gained much popularity since it removes the necessity of a trusted central authority while maintaining stability, security and non-modifiability. The consensus algorithm plays a crucial role in maintaining the safety and efficiency of the blockchain. Although different consensus algorithms are designed on miners' features (e.g., processing power, stake etc.), the algorithms still face several security and scalability challenges. In order to address this, we propose a model that integrates users' preferences into the committee member auction consensus algorithms which is a lightweight, scalable and attack-proof algorithm. The model can help increase the system's performance by reducing the delays in the blocks validations. It elects committee members based on the preferences required by the users for every transaction and makes use of a bidirectional-linked blockchain to mitigate long-range attacks. Our proposed model is analyzed in terms of performance and security and is compared with existing committee member auctions algorithms. Different preference-based scenarios are simulated to support the analysis.**

***Index Terms*—Bidirectional-linked blockchain, Consensus algorithm, Preference models, Mining process**

I. Introduction

Blockchain is a shared, immutable distributed ledger of transactions duplicated and distributed across the entire network of computer systems. Even though blockchain allows secure and anonymous transactions, it faces several security and scalability issues. Many attacks have been launched on blockchain-based systems. For instance, the Reorg Tracker observed 18 double-spend attacks on four cryptocurrencies [1]. Blockchain networks using the Proof of Work (PoW) consensus algorithm are vulnerable to double spend attack, while those using the Proof of Stake (PoS) consensus algorithm are vulnerable to both long-range and eclipse attacks. Another major challenge faced by blockchain is scalability and throughput. When there is an increase in the transaction history, it could topple the overall system. As there is an increase in the number of transactions, the block's validation time increases due to the consensus process. For example, building a new block in the Bitcoin network takes almost 10 minutes.

Research has been conducted to eliminate the vulnerabilities of blockchain. A double-spending prevention mechanism for bitcoin zero-confirmation transactions is proposed in [2]. However, it works only with UTXO or Bitcoin models. Checkpoints are adopted to define the correct chain periodically in [3] to defend against long-range attacks. However, when generating the checkpoints, it is at risk for DDoS attacks. In terms of eclipse attacks, an eclipse-attack detection model for Ethereum is proposed [4]. Nevertheless, the model is capable of detecting the Ethereum network attack traffic based on two criteria: information entropy (information in the assault packets) and statistical statistics. Many different solutions are proposed for improving the scalability of the blockchain network. For example, layer two solutions uses an external, parallel network to facilitate transactions away from the main blockchain. However, these mechanisms add a lot of complexity to the existing system. A lightweight and scalable committee member auction consensus algorithm addresses the scalability and security vulnerabilities but does not consider the miner's validation time, failure rate and malicious activities.

In order to solve the aforementioned challenges, this paper proposes a model that improves the blockchain's performance and security vulnerabilities. The preference model can provide flexibility to users to select their priorities for the mining procedures. Users set the preference to elect committee members during the consensus process based on parameters like stake, processing power, cost, disk space, etc., for every transaction in the blockchain. For example, if the user wants higher security, they could set the preference to stake for the transaction as it shows the miner's intention. Similarly, users could set the preference to processing power if they desire a faster transaction validation. Many preference models are proposed that can be integrated into the consensus process. The proposed model uses a bidirectional-linked blockchain as it can resist long-range attacks by detecting if the block has been altered using the previous and next pointers.

The main contributions of this paper are as follows:

1) A system model with a bidirectional-linked blockchain with a custom consensus algorithm that integrates the preference model to the committee member auction consensus algorithms;
2) Performance and frequency results and analysis of the proposed system and existing committee member auction algorithm using different simulations;
3) Theoretical analysis of double spend, long-range and eclipse attack based on the results of the simulations.

The remainder of the paper is structured as: in Section II, we present the background and context about bidirectionally-

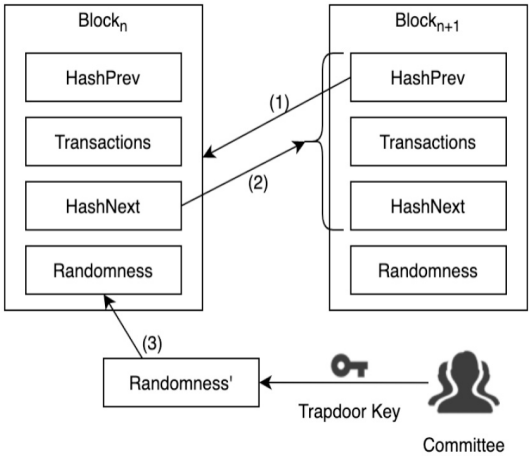

Fig. 1. A bidirectional-linked blockchain model

linked blockchain, committee member auction algorithms and cryptography tool. Section III proposes our preference-based committee member auction consensus algorithm. Section IV evaluates the performance and frequency of our proposed algorithm. Section V analyses the security features of our approach. Section VI, we present the related work and finally, we will present the conclusion and future work in section VII.

II. Background

This section reviews the cryptography tools, existing committee member auction consensus algorithm and other contributions on the defence against blockchain attacks.

A. Verifiable Random Functions (VRFs)

The verifiable random functions are public-key pseudorandom functions that provide verifiable proof that its outputs were calculated correctly [8]. Algorand intoduced VRFs to privately check if the miner is selected to participate in the consensus phase [9]. Algorand is a highly scalable blockchain framework that uses Byzantine Agreement (BA) protocol to reach consensus. However, the miners in Algorand are weighted based on the balance of tokens in wallets, which means a miners with more tokens is more vulnerable to DDoS attacks and causes the performance of the blockchain to be downgraded.

B. Bidirectional-Linked Blockchain

The block structure in bidirectional-linked blockchain is slightly different from the existing blockchains. There are two pointers in a block: the forward pointer and reverse pointer along with transactions and randomness. The forward pointer stores the previous block hash value while the reverse pointer stores the hash value of the next block. It is represented in Fig. 1 as $HashPrev$ and $HashNext$, respectively. The consensus reached by the distributed participants is represented by the randomness which a replacement of the nonce. The bidirectional-linked blockchain will allow only appending operations, similar to other blockchain models.

C. Committee Member Auction consensus algorithm

The Committee Members Auction (CMA) consensus algorithm is a lightweight, scalable consensus algorithm that is attack resistance. It uses VRF for electing committee members across distributed blockchain nodes.

The CMA consensus algorithm starts by electing the committee members. The miners acquire their $vhash$ and π by using the seed (a random value generated for each term) and their respective private keys. If $vhash$ falls into a specific range γ, the miner is treated as a committee member. The elected committee members propose new blocks based on the transactions they receive through the gossip protocol. To avoid conflict over generating blocks, there are also priorities among committee members. The miner with the lowest $vhash$ has the highest block generating priority. When a miner receives a block from a higher priority miner, it will accept the block. Otherwise, the block is broadcasted using gossip protocol.

D. Preference Model

The preference model is a formal method of ranking the objects (in our case, miners) based on the user requests and preferences. A preference term can be either an atomic or composite. Atomic preference has a single object which can either be a qualitative or quantitative preference, while composite preference can have multiple qualitative or quantitative preference objects.

A preference can be expressed as "x is preferred over y", where x and y are instances of miners. A preference model is formally defined as follows.

Definition 1: Let C be a non-empty set of parameters of miners and $dom(C)$ the set of all possible instances of those parameters. We define preference as $P = (C, <^p)$ where $<^p \subseteq dom(C) \times dom(C)$ is s strict partial order (irreflexive, transitive and asymmetric), and if x,y $\in dom(C)$, then x $<^p$ y is interpreted as "y is preferred rather than x".

The definitions of numerical, prioritized and balanced preferences are provided below.

1) Numerical Preference: The numerical preference is a combination of a number of score preferences. A score preference is defined as a scoring value that takes a property value as its argument and returns a real value. The higher the value returned by the function, the more preferred the property value is. We define score preference as:

Definition 2: Let f: $dom(C) \rightarrow \Re$ be a scoring function and $<$ the usual less-than order in $\Re$. $P_f = (C, <^{P_f})$ is a score preference if for x, $y \in dom(C)$:

$$x <^{P_f} y \iff f(x) < f(y) \tag{1}$$

Numerical Preference takes the values returned by each score preference as its argument and returns another real number that gives information about the global preferences after considering all the properties referred by concrete score preferences. We define numerical preference as:

Definition 3: Let f, g and h be scoring functions that define score preferences $P_f = (C, <^{P_f})$, $P_g = (C, <^{P_g})$, $P_h = (C, <^{P_h})$, respectively and F: $\Re \times \Re \longrightarrow \Re$ be a

combination function. For $x = (x_1, x_2), y = (y_1, y_2), z = (z_1, z_2) \in dom(C_1) \times dom(C_2) \times dom(C_3)$, $P = (C_1 \cup C_2 \cup C_3 <^{rank_F(P_f,P_g,P_h)})$ is a numerical preference if:

$$\begin{aligned} x &<^{rank_F(P_f,P_g,P_h)} y \\ &<^{rank_F(P_f,P_g,P_h)} z \\ &\iff F(f(x_1), g(x_2)) \\ &< F(f(y_1), g(y_2)) \\ &< F(f(z_1), g(z_2)) \end{aligned} \tag{2}$$

2) Prioritized Preference: A prioritized preference P is composed of two preference terms P_1 and P_2, where P_1 is considered more important than P_2. Thus, P_2 is evaluated only if P_1 does not return enough information to rank the parameters or in case of conflict. The prioritized preference is defined as:

Definition 4: Let $P_1 = (C_1, <^{P_1})$ and $P_2 = (C_2, <^{P_2})$ be two different preference defined after C_1 and C_2 properties and $x = (x_1, x_2), y = (y_1, y_2) \in dom(C_1) \times dom(C_2)$ be two value tuples for each property. $P = (C_1 \cup C_2, <^{P_1 \& P_2})$ is a prioritized Preference if:

$$x <^{P_1 \& P_2} y \iff x_1 <^{P_1} y_1 \vee (x_1 = y_1 \wedge x_2 <^{P_2} y_2) \tag{3}$$

3) Balanced Preference (Pareto-optimality Principle): A balanced preference P is a combination of two preference terms P_1 and P_2. It uses the Pareto-optimality principle, a situation where no preference criterion can be made better without making at least one preference criterion worse off. Therefore, P_1 and P_2 are considered equally important. Intuitively, this preference balances the fulfilment of each preference component so that the composite preference is the average degree of preference, taking both components into account.

Definition 5: Let $P_1 = (C_1, <^{P_1})$ and $P_2 = (C_2, <^{P_2})$ be two different preference defined after C_1 and C_2 properties and $x = (x_1, x_2), y = (y_1, y_2) \in dom(C_1) \times dom(C_2)$ be two value tuples for each property. $P = (C_1 \cup C_2, <^{P_1 \otimes P_2})$ is a balanced preference if:

$$\begin{aligned} x <^{P_1 \otimes P_2} y \iff &(x_1 <^{P_1} y_1 \wedge (x_2 <^{P_2} y_2)) \vee \\ &(x_2 <^{P_2} y_2 \wedge (x_1 <^{P_1} y_1))) \end{aligned} \tag{4}$$

III. Proposed System Model

Our proposed model integrates users preferences into the committee member auction consensus algorithm in order to select more matchable miners for validating the transactions. The model also makes use of the bidirectional-linked blockchain to resist some attacks. The preference models integrated into the consensus algorithm are numerical, prioritized and balanced preferences. The numerical preference exploits scoring functions which takes input as miner's parameters and a scoring values and returns a score. These scores are aggregated to get the total score P. The scoring function $f(s)$ for the numerical preference model is defined as:

$$f(s) = x \times v, \tag{5}$$

where x is the numeric value of the miner's parameter and v is the scoring value. The priority preference model compares the most preferred miner's parameter to calculate the score. If the score is equal, then other miner's parameters are considered. The balance preference model gives equal priority to all preferred miner's parameters. The algorithms used for the aforementioned preference models is shown in Algorithm 1, 2 and 3, respectively.

Algorithm 1 Numeric Preference Algorithm

Input: Array Miner's parameter with scoring value
Input: Miner's parameter with miner's value
Output: $tScore$ be the total score calculated by the miner
function GETNUMERICPREFERENCESCORE
  $tScore \leftarrow 0$ ▷ Total Score
  for $\forall\ pref\ \in\ prefs$ **do**
    if $pref$ is "$stake$" **then** ▷ Stake
      $tScore \leftarrow tScore + (minerParams[\text{“}stake\text{”}] \times pref[\text{“}stake\text{”}])$
    end if
    if $pref$ is "$power$" **then** ▷ Processing Power
      $tScore \leftarrow tScore + (minerParams[\text{“}power\text{”}] \times pref[\text{“}power\text{”}])$
    end if
    if $pref$ is "$disk$" **then** ▷ Disk Space
      $tScore \leftarrow tScore + (minerParams[\text{“}disk\text{”}] \times pref[\text{“}disk\text{”}])$
    end if
    if $pref$ is "$cost$" **then** ▷ Cost
      $tScore \leftarrow tScore + (minerParams[\text{“}cost\text{”}] \times pref[\text{“}cost\text{”}])$
    end if
  end for
  return $tScore$
end function

Algorithm 2 Prioritized Preference Algorithm

Input: Array of Miner's parameter
Input: Miner's parameter with miner's value
Output: $tScore$ be the total score calculated by the miner
function GETPRIORITYPREFERENCESCORE
  $tScore \leftarrow 0$
  if $pref$ is "$stake$" **then**
    $tScore \leftarrow tScore + pref[\text{“}stake\text{”}]$
  end if
  if $pref$ is "$processingPower$" **then**
    $tScore \leftarrow tScore + pref[\text{“}processingPower\text{”}]$
  end if
  if $pref$ is "$diskSpace$" **then**
    $tScore \leftarrow tScore + pref[\text{“}diskSpace\text{”}]$
  end if
  if $pref$ is "$cost$" **then**
    $tScore \leftarrow tScore + pref[\text{“}cost\text{”}]$
  end if
  return $tScore$
end function

Algorithm 3 Balanced Preference Algorithm

Input: Array of Miner's parameter
Input: Miner's parameter with miner's value
Output: $tScore$ be the total score calculated by the miner
function GETBALANCEDPREFERENCESCORE
  $tScore \leftarrow 0$
  for $\forall\ pref\ \in\ prefs$ **do**
    if $pref$ is "$stake$" **then**
      $tScore \leftarrow tScore + pref[\text{"}stake\text{"}]$
    end if
    if $pref$ is "$processingPower$" **then**
      $tScore \leftarrow tScore + pref[\text{"}processingPower\text{"}]$
    end if
    if $pref$ is "$diskSpace$" **then**
      $tScore \leftarrow tScore + pref[\text{"}diskSpace\text{"}]$
    end if
    if $pref$ is "$cost$" **then**
      $tScore \leftarrow tScore + pref[\text{"}cost\text{"}]$
    end if
  end for
  return $tScore$
end function

The committee member auction consensus algorithm is based on periodic elections of miners and requires distributed participants (miners) to have a synchronized clock. Each election period is called a term. The algorithm can be divided into three steps: election of committee members, proposing a new block and reaching consensus and generating the new block.

1) Election of Committee Members: During the election of the committee members, the threshold value n along with the preferred terms and preference model are acquired by the miners from the transaction. For each term, the miner can perform the calculation based on the preference models and preference terms to acquire a value total score P. For example, if the transaction has preference model as numeric preference and preference terms as processing power and stake with their respective scoring values, every miner can acquire their score P by performing calculations based on the Algorithm 1. If the total score is grater than the threshold ($P > n$), the miner is considered a committee member for the term and can validate the block.

After the committee member's election, the trapdoor keys are divided into η parts (where η is the number of committee members elected) and distributed across the committee members to protect the tampering of the forward pointer.

2) Propose a New Block: After the committee members are elected for the term, a new block is proposed based on the transaction they received. The priority among committee members is based on the value of P. A higher value for P leads to a higher priority for validating and generating blocks. When a miner receives a block from a higher priority miner, it will automatically accept the block; otherwise, it broadcasts its block.

3) Reaching Consensus: After the consensus is reached by the committee members, every member sends their part of the trapdoor key along with the hash of the newly proposed block $Block_{n+1}$. The smart contract repairs the randomness of the block $Block_n$ when enough secrets are collected to reconstruct the trapdoor key. By repairing the randomness of $Block_n$, the forward pointer of the block remains unchanged and the reverse pointer of $Block_{n+1}$ points at $Block_n$ with a new $HashNext$.. Finally, $Block_{n+1}$ is appended to the chain with both the forward and the reverse pointer.

A. Consensus Process

Initially, a blockchain network collects the transactions and converts it into a block. After generating a block, the entire network must agree on the transaction's validity, i.e., reach a consensus before confirming the transaction to the blockchain. Each transaction in the block will contain the users preference of the preference model and a collection of miners parameters which is preferred. For numeric preference, scoring values for preferred miners parameters will be contained within the transaction.

As an example, in Fig. 2, we have a blockchain network with 8 miners A, B, C, D, E, F, G and H. During each election period, all the miners check if they are elected as committee members by calculating their total scores based on the preference information present in the transaction. It is supposed that after the first step, miners A, D, E and H are elected as committee members and miner A has the highest priority over the other miners during the term, miner A proposes a new block. The trapdoor keys are divided into four parts and distributed to the committee members. The miners D, E and H verify the newly proposed block by miner A. After the committee member verifies the transaction, the committee member sends out a message which includes the trapdoor key. When a considerable number of the trapdoor keys are collected, the smart contract will automatically get invoked to repair the randomness of $Block_n$. The miner A will broadcast the newly generated block to the network. If any of the miner D, E or H does not agree with the newly proposed block, they do not send out their trapdoor key. If majority of the committee members do not agree with the block, it is rejected.

IV. PERFORMANCE EVALUATION

This section evaluates the performance of the preference-based committee member auction and the existing committee member auction algorithms by executing different scenarios.

A. Environment Setting

The preference-based CMA and existing CMA simulation environment are coded in Java 8. The Java Open JDK is used to build the simulation. The hardware configuration includes an Apple M1 chip 8 core processor and 8 GB of RAM. The default parameters of the simulation are given in Table I.

For the simulations, every miner is configured with processing power, stake, disk space and cost with a random

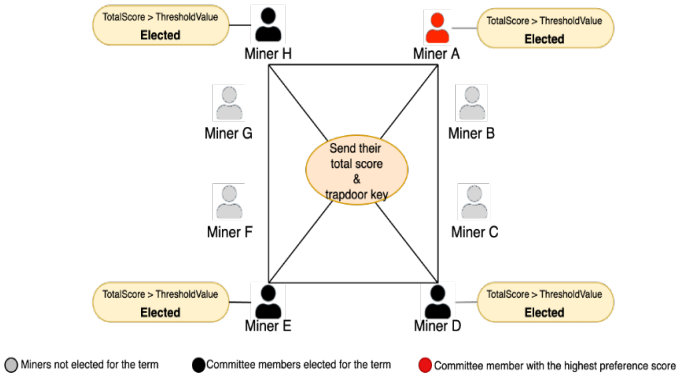

Fig. 2. Preference Based CMA Working

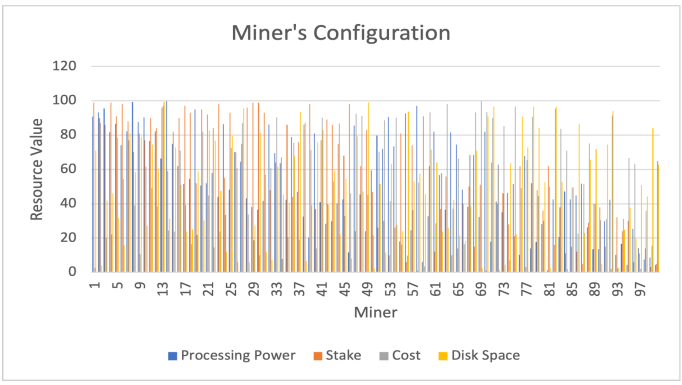

Fig. 3. Miner Configuration

value. For all the following scenarios, the miner configuration is the same. The configuration is shown in Fig. 3. The difficulty of mining the block is based on the computing power of the hardware of the miner. In other words, higher the processing power, higher the computation power, lower the processing time, and lower the difficulty of validating the block. The network latency is considered to be negligible. It is assumed that the user sends the preference model and the preferred miners parameters with the transaction. For each experiment 1000 transactions are generated. Every transaction goes through the consensus algorithm to be validated.

TABLE I
SIMULATION PARAMETER SETTINGS

Paramter	Value
The number of participant members (η)	100
The incoming transaction speed (tx/s)	1
Total incoming transaction	1000
Number of transaction in a block	1

B. Performance Comparison

The performance of each node in the blockchain network varies. The first experiment assumes that the collections of preference in all 1000 transactions contains only processing power and sake. For numeric preference, a higher scoring value is given to the processing power, and a lower value is given to the sake. In terms of implementation, 90 is given to processing power, and 10 is given to the stake. Therefore, a miners with a higher score than the threshold value are selected. In the case of priority preference, a higher priority is given to processing power than to the sake. Therefore, a miners with higher processing power are selected. Similarly, a miner with both high stake and high processing power is selected for Balanced preference. In the case of the existing committee member auction algorithms, the miners are selected randomly based on the value of γ which is obtained from the VRF. The result of the experiment is shown in Fig. 4.

The graph shows that the validation block's time for the priority preference is shorter than the numeric and balanced preferences. This is because in the case of priority preference, always the miner with the highest processing power is selected to mine the block. The time taken to do so would be less as the difficulty to mine would be less. The simulation found that the miner 78, having processing power configured as 99.90, was always elected to mine as it has the highest processing power among all the other miners. For numeric preference, a miner with the highest overall score, i.e., 90% of processing power and 10% of the stake, is considered compared to only processing power in priority. The simulations found that the miner 60, configured with the processing power as 99.1 and stake as 70, was always elected for mining the block. It has slightly less processing power but a higher stake than the miner 78, which has a stake of 58. In the case of balanced preference, equal preference is given to the processing power and stake, i.e., 50% value of processing power and 50% value of the stake are considered. Miner 8, configured with the processing power of 90.8 and the stake of 99 was always selected as it had an equally high stake and processing power, but the processing power is significantly lower than that of the miner 78 and the miner 60. In the case of the existing committee member auction, the performance could not be guaranteed since the miner is selected randomly. Some transactions take very less time as the elected miner has high processing power to propose the block and vice versa. Every time the simulation ran, a different set of miners was elected because of the randomness of the algorithm.

For the second experiment, the preferences are set randomly for all the 1000 transactions, i.e., it is not guaranteed that processing power is given priority. For the numeric, priority and balanced preferences, the miners parameters are selected randomly. In the case of numeric preference, the scoring values are also generated randomly. The performance result is shown in Fig. 5. From the graph, it is evident that there is no guarantee of performance when processing power is not considered. The preference-based CMA and existing CMA algorithms have a similar trend, and the performance depends on the elected miner's processing power.

V. Security Analysis

Blockchain faces major scalability and security challenges like double spend, long range and eclipse attacks. Double

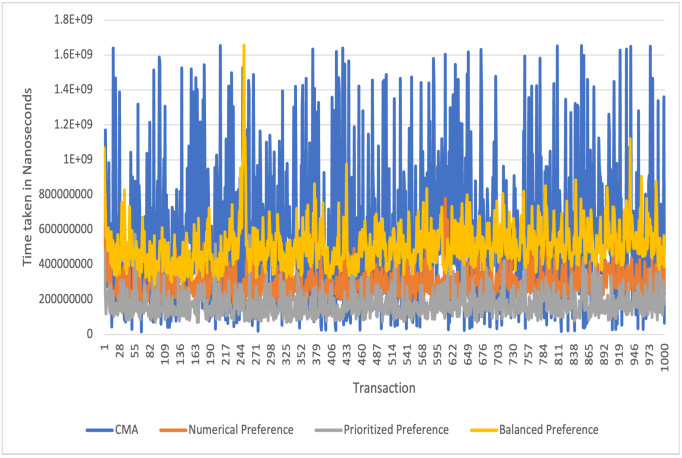

Fig. 4. Performance Evaluation for Processing Power and Stake as Preference

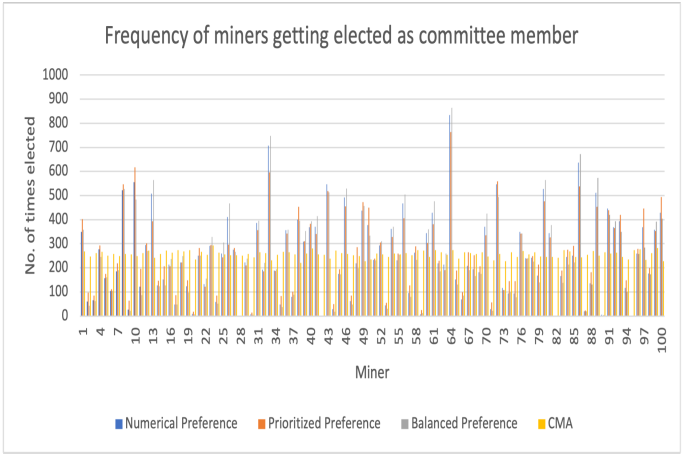

Fig. 6. Miner getting elected during election of committee members

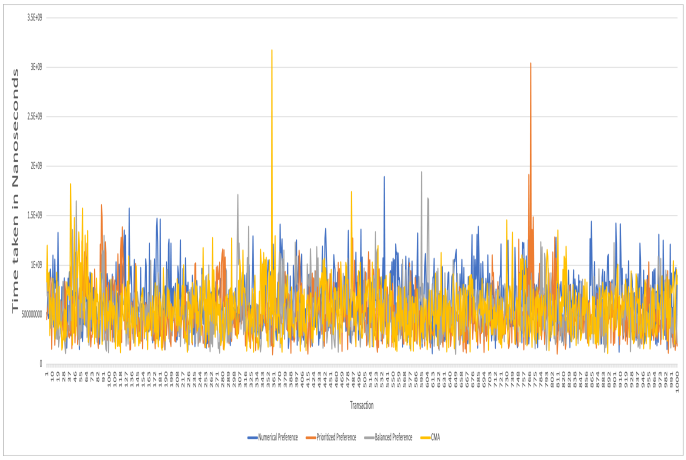

Fig. 5. Performance Evaluation for Random Preference

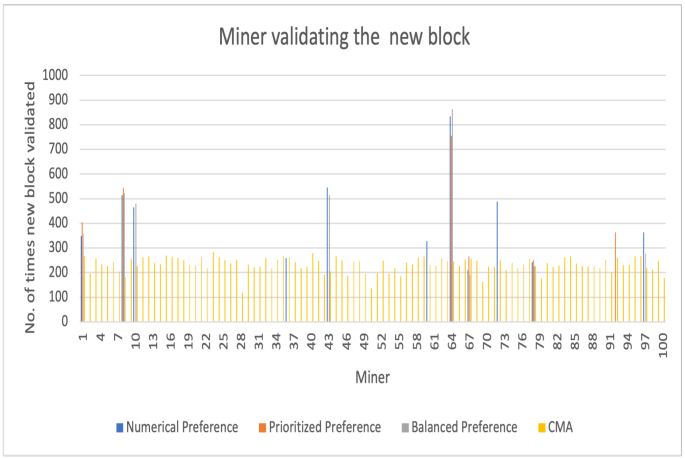

Fig. 7. Miner generating the new block

spend and long-range attacks are caused by uncertainty of the added blocks and the subsequent blocks. However, the subsequent direction of any block, starting from the genesis block, may be known using the reverse pointer design, making the entire chain undisputed.

For the third experiment, a comparison is made on the number of times a miner is elected as a committee member. The preference is set randomly for all the 1000 transactions, similar to the second experiment. The result is shown in Fig. 6. It is evident that some of the miners are getting elected more than others due to their configuration in preference-based CMA. The randomness of the system is low as the only factor that brings randomness is the threshold value which is randomly generated at the beginning of each term. Each colour shows the number of times a miner is elected for the particular preference model. In the case of the existing CMA, the probability of getting selected is (τ/η), where τ is the number of committee members and η is the number of participant members in a blockchain network. Therefore, every miner has an equal chance of getting elected.

A similar trend can be observed when comparing the number of times a miner proposes the block. In the case of the preference-based model, we find that only a handful of miners are generating the new block. The experiment showed that in most cases, only 11% of the miners were selected to propose a new block. This is according to the Pareto principle, which states that for most outcomes, roughly 80% of the consequences come from 20% of the cause. In the case of the exiting CMA, every miner has an equal chance of proposing a new block as it depends on the value of γ. Since the same committee members are getting elected most of the time, the model is vulnerable to security and scalability challenges as compared to the existing CMA. If the same miner has to solve several blocks, the processing time would be high, which could lead to higher delays.

By adding preference, we gave flexibility to the users to select the type of miners they want to mine their block. Therefore, theoretically speaking, if a miner wants to always get elected and propose the block, they should not dominate only a single parameter, for example, processing power, as there is no guarantee on what the users can set as a preference. Preference added certain randomness to the system and increased the

entropy of the system.

Further security analysis of the attacks is presented below.

A. Double Spend Attack and Long-Range Attack Resistance

A double spend attack occurs when the same cryptocurrency is being spent twice and when the transaction information is altered and entered into the blockchain. A long-range attack is caused when a miner tries to create an alternative chain from an existing blockchain. Both these attacks occur when an uncertain new block is added to the blockchain.

The proposed model will completely resist the long-range attack because of the bidirectional-linked blockchain. The reverse pointer ($HashPrev$) will hold the previous block's hash value, making it impossible to create a new chain. The chain can be undisputed as the subsequent of any block can be determined starting from the genesis block using the pointer values.

Double spend could be treated as Gambler's ruin problem, a famous statistical scenario centred around conditional probabilities and experimental outcomes as analyzed by Nakamoto in [12]. The probability of an attacker catching up with the honest miner (M) can be calculated as:

$$M(q,b) = 1 - \sum_{k=0}^{b} \frac{\lambda^k e^{-\lambda}}{k!}(1 - (\frac{q}{p})^{b-k}) \tag{6}$$

where b represents the number of blocks merchant waits before handing over physical goods. p is the probability an honest node finds the next block. q is the probability the attacker finds the next block. λ is the blocks producing rate of the attacker during the interval that honest miners produce b blocks, which is calculated by:

$$\lambda = b\frac{q}{p} \tag{7}$$

Based on equation 7 to find out the probability that the attacker could overtake the honest miners (which means that the double-spend attack happens), b is replaced with $b+1$

$$M(q,b) = 1 - \sum_{k=0}^{b+1} \frac{\lambda^k e^{-\lambda}}{k!}(1 - (\frac{q}{p})^{b+1-k}) \tag{8}$$

For preference-based committee member auction consensus algorithm, q is proportion to the miner's resource values like computing power, stake owned, etc. Theoretically, the value of q would be propositional to the preference parameter set for the term. Therefore, the probability that the attacker controls all the committee members for each term is propositional to q. A double-spend attack can happen if the attacker controls all the committee members. But, if many committee members are not controlled by the attacker (honest committee members), they will not provide their part of the trapdoor keys if they dispute the block, making the model resistant to the double spend attack.

From the experiments, it was noted that the same committee members are elected most of the time. If the attacker controls these miners, the probability of the double spend attack increases.

B. Eclipse Attack Resistance

An eclipse attack is a network-based attack in which an attacker creates an artificial environment around a miner to manipulate it into wrongful action. This attack depends on the entropy (randomness) of the system. If the entropy is high, it is difficult to predict the next term's committee members, making the attack ineffective.

The entropy of the system is defined as:

$$H(X) = -\sum_{i=1}^{n} P(u_i) \log_b P(u_i) \tag{9}$$

where $P(u_i)$ is the probability of miner u_i participating in the consensus algorithm, and b is the base of the logarithm used. The likelihood of the users being elected is propositional to their resource values. From our experiment and according to the Pareto principle, we found that the entropy of the preference-based CMA system was low. Lower entropy means a lower level of security. If certain miners are targeted, it would be easier to launch an eclipse attack. The existing CMA has a higher entropy as the elected miner was random, making it difficult to predict the committee member of the following term and launch the attack.

Even if the attack was launched at the elected committee member with the highest priority to propose the block, it would not hinder the mining process. This is because the committee member with the next highest priority can propose their transaction request and continue to reach a consensus. In preference-based CMA, the attacker has to control all the committee members to hinder the consensus algorithm. Since most of the time, the same miners are elected, this would not be impossible or ineffective. For the original CMA, it would be impractical to launch as every term will have a different set of miners.

VI. Related Work

There are several blockchain-based solutions for mitigating double spend attacks [2], [5]. These solutions detect the attack using a listening period and observers. They discourage double spending attempts of zero-transactions in Bitcoin or UTXO models by creating a special type of outputs that enforce private key disclosure in case of double spend attempts. However, in a peer-to-peer network, the message delivery between nodes is often not so timely, and the order of messages is not guaranteed, which makes their observers unreliable. In [3], the authors proposed checkpoints which are immutable. It adopts a multi-variable (block, active user, stake parameters) based strategy to decide the next checkpoint. However, it is vulnerable to DDoS attacks, especially when creating checkpoints and relies on a centralized server to define the correct chain periodically. In [4], an eclipse-attack detection model for Ethereum has been proposed. The model is based on a random forest classification algorithm, in which features of attack connection flow are defined. Nevertheless, it is only responsible for detecting attack traffic based on two features, namely information entropy (information in the attack packets) and statistical features, for the Ethereum network.

However, these approaches focused on mitigating the attacks for blockchain networks using proof of work and proof of stake consensus algorithm. More flexible consensus algorithm is required which has a higher performance and is resistant to the attacks. We propose a preference-based committee member auction consensus algorithm that uses bidirectional-linked blockchain.

VII. Conclusion

This paper proposes a preferenced-based committee member auction consensus algorithm along with a bidirectional-linked blockchain. We investigate the pros and cons of introducing a preference model into the committee member auction consensus algorithm. The committee members (miners) are elected based on the preference set by the users for a transaction. Every miner calculates their score using the algorithms to check if they are elected as a committee member. The bidirectional-link between the blocks is constructed based on the Chameleon-hash function, whose trapdoor keys are split among every committee member. On investigating the performance and security of the preference-based model, it was found that the performance can be guaranteed only if the preference is set to high processing power. The preference model did add some entropy to the system and gave flexibility to the user to select the criteria of the miners responsible for validating the block. The existing committee member auction does not guarantee performance. It was also found that the process of electing the miner was influenced by the Pareto principle. Only 11% of the miner was responsible for proposing a new block. Since the entropy of the model is low, it is possible to carry out double spend and eclipse attacks and the model also faces scalability challenges.

Future work will focus on completely eliminating the double spend and eclipse attack by increasing the entropy of the model. Moreover, investigating the scalability of our solution in a real blockchain network will be another potential research direction.

References

[1] D.J. Moroz, D.J. Aronoff, N Narula and D.C. Parkes, "Double-spend counterattacks: Threat of retaliation in proof-of-work systems." arXiv preprint arXiv:2002.10736, 2020.

[2] C. Pérez-Solà, S. Delgado-Segura, G. Navarro-Arribas, and J. Herrera-Joancomartí, "Double-spending prevention for bitcoin zero-confirmation transactions," Int. J. Inf. Security, vol. 18, no. 4, pp.451–463, 2019.

[3] I.A.I AlMallohi, A.S.M Alotaibi, R. Alghafees, F. Azam and Z.S. Khan, "Multivariable based checkpoints to mitigate the long range attack in proof-of-stake based blockchains" in Proc. 3rd Int. Conf. High Perform. Compilation Comput. Commun., pp. 118–122, 2019.

[4] G. Xu, B. Guo, C. Su, X. Zheng, K. Liang, D.S. Wong and H. Wang, "Am I eclipsed? A smart detector of eclipse attacks for Ethereum". Computers & Security, 88, pp.101604, 2020.

[5] A. Gervais, G.O. Karame, K. Wust, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains" in Proc. ACM SIGSAC Conf. Comput. Commun. Security, pp.3–16, 2016.

[6] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on bitcoin's peer-to-peer network" in Proc. 24th USENIX Security Symp. (USENIX Security), pp.129–144, 2015.

[7] C. Xu, Y. Qu, T.H. Luan, P.W. Eklund, Y. Xiang and L. Gao, "A Lightweight and Attack-Proof Bidirectional Blockchain Paradigm for Internet of Things" in IEEE Internet of Things Journal, vol. 9, no. 6, pp.4371-4384, 2021.

[8] N. Bitansky, "Verifiable random functions from non-interactive witness-indistinguishable proofs" in Journal of Cryptology, 33(2), pp.459-493, 2020.

[9] Y. Gilad, R. Hemo, S. Micali, G. Vlachos and N. Zeldovich, "Algorand: Scaling byzantine agreements for cryptocurrencies" in Proceedings of the 26th symposium on operating systems principles, pp.51-68, 2017.

[10] L.M. Bach, B. Mihaljevic and M. Zagar, "Comparative analysis of Blockchain consensus algorithms" in 41st International Convention on Information and Communication Technology, Electronics and Micro-electronics (MIPRO), pp.1545-1550, 2018.

[11] R. García, J.M., D.C Ruiz and R.A. Cortés, "A model of user preferences for semantic services discovery and ranking", in Extended Semantic Web Conference, pp.1-14, 2010.

[12] S. Nakamoto and A. Bitcoin, "A peer-to-peer electronic cash system" in Decentralized Business Review, pp.21260, 2008.

[13] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei and C. Qijun "A review on consensus algorithm of blockchain" in IEEE International Conference on SMC, pp.2567-2572, 2017.

[14] W. Kießling, "Foundations of preferences in database systems" in VLDB'02: Proc. of the 28th International Conference on Very Large Databases, pp.311-322, 2002.

[15] J.M. García, D. Ruiz and A. Ruiz-Cortés, "A Model of User Preferences for Semantic Services Discovery and Ranking" in ESWC: 7th Extended Semantic Web Conference, pp.1-14, 2010.

[16] S.M.H. Bamakan, A. Motavali and A.B. Bondarti, "A survey of blockchain consensus algorithms performance evaluation criteria." in Expert Systems with Applications, 154, pp.113385, 2020.

[17] M. Barati and R. St-Denis, "Team formation through preference-based behavior composition" in German Conference on Multiagent System Technologies, pp. 54-71, 2017.

[18] G. Ramezan and C. Leung, "An analysis of proof-of-work based blockchains under an adaptive double-spend attack" in IEEE Trans. Ind. Informat., vol.16, no.11, pp.7035–7045, 2020.

[19] N. Anita and M. Vijayalakshmi, "Blockchain security attack: A brief survey" in Proc. IEEE 10th Int. Conf. Comput. Commun. Netw. Technol. (ICCCNT), pp. 1–6, 2019.

[20] R. Dunford, Su Quanrong and E. Tamang, "The pareto principle.", 2014.

[21] P. Otte, M. de Vos, and J. Pouwelse, "TrustChain: A sybil-resistant scalable blockchain" in Future Gener. Comput. Syst., vol. 107, pp. 770–780, 2020.

[22] J.M García, D. Ruiz and A. Ruiz-Cortés, "A Model of User Preferences for Semantic Services Discovery and Ranking." in ESWC: 7th Extended Semantic Web Conference, pp.1-14, 2010.