electronics

Article

Proof of Fairness: Dynamic and Secure Consensus Protocol for Blockchain

Abdulrahman Alamer [1,*] and Basem Assiri [2]

[1] Department of Information Technology, Jazan University, Jazan 45142, Saudi Arabia
[2] Department of Computer Science, Jazan University, Jazan 45142, Saudi Arabia; babumussmar@jazanu.edu.sa or bas0911@hotmail.com
[*] Correspondence: amalameer@jazanu.edu.sa

Abstract: Blockchain technology is a decentralized and secure paradigm for data processing, sharing, and storing. It relies on consensus protocol for all decisions, which focuses on computational and resource capability. For example, proof of work (PoW) and proof of stake (PoS) are the most famous consensus protocols that are currently used. However, these current consensus protocols are required to recruit a node with a high computational or a large amount of cryptocurrency to act as a miner node and to generate a new block. Unfortunately, these PoW and PoS protocols could be impractical for adoption in today's technological fields, such as the Internet of Things and healthcare. In addition, these protocols are susceptible to flexibility, security, and fairness issues, as they are discussed in detail in this work. Therefore, this paper introduces a proof of fairness (PoF) as a dynamic and secure consensus protocol for enhancing the mining selection process. The selection of the miner node is influenced by numerous factors, including the time required to generate a block based on the transaction's sensitivity. Firstly, a reverse auction mechanism is designed as an incentive mechanism to encourage all nodes to participate in the miner selection process. In a reverse auction, each node will draw its strategy based on its computational capability and claimed cost. Secondly, an expressive language is developed to categorize transaction types based on their sensitivity to processing time, ensuring compatibility with our miner selection process. Thirdly, a homomorphic concept is designed as a security and privacy scheme to protect the bidder's data confidentiality. Finally, an extensive evaluation involving numerical analysis was carried out to assess the efficiency of the suggested PoF protocol, which confirms that the proposed PoF is dynamic and more efficient than current PoW and PoS consensus protocols.

Keywords: blockchain; auction-based; proof of fairness (PoF); miner; consensus protocol; security

check for updates

Citation: Alamer, A.; Assiri, B. Proof of Fairness: Dynamic and Secure Consensus Protocol for Blockchain. *Electronics* **2024**, *13*, 1056. https://doi.org/10.3390/electronics13061056

Academic Editors: Wenquan Jin, Christos J. Bouras, Shabir Ahmad and Meilan Jiang

Received: 19 January 2024
Revised: 8 March 2024
Accepted: 8 March 2024
Published: 12 March 2024

1. Introduction

Blockchain technology emerged as a secure and trusted paradigm for the sharing and storage of data. It is built on a decentralized methodology, which securely stores committed transactions in a chain of blocks that will be shared on distributed ledgers. Blockchain technology was first proposed to ensure the secure performance of Bitcoin cryptocurrency [1]. However, it has since been used for many cloud-based applications such as health cloud, mobile cloud, Internet of Things, etc. [2–5].

The blockchain operates according to a sequence of processes, starting with transaction validation, block creation, and distributed consensus algorithms, and ending with the block verification method [6]. It uses a network of computers, called nodes. In the beginning, transactions are initiated, and then a node (j) that acts as a miner node verifies the transactions (where j is an integer number greater than 0). Next, the miner node (j) will generate a new block (b) for validated transactions. After that, it sends b via the network to other connected nodes for verification. According to the verification, nodes vote on the correctness of the block. If the block's correctness is not confirmed, it is aborted and ignored. However, if the block's correctness is confirmed, it is chained to the block list (ledger) using hashing.

Undoubtedly, the process of selecting a node to serve as a proposed miner node is considered an essential process of blockchain methodology [7]. Consequently, many approaches have been introduced for smoothly selecting proposed nodes, such as proof-of-work (PoW), proof-of-stake (PoS), proof-of-space, and the practical Byzantine fault Tolerance (PBFT), which are deemed the most popular approaches for blockchain methodology [2,8].

These approaches are built on a methodology that involves selecting a node with powerful computational abilities and resources. For example, in PoW, the node that succeeds in solving a certain complex puzzle will be referred to as a powerful computational node and will then be selected as a miner node. In contrast, within PoS, a node that has a large amount of cryptocurrency to stake can win. Usually, nodes need to work as miners many times to build credit, indirectly requiring higher powerful computational resources among all connected nodes to be considered as proposed miners. The same thing applies to the proof-of-space and practical Byzantine fault tolerance (PBFT). Although these approaches have functionally succeeded in selecting a miner node, they are not practical for adoption in the most sensitive systems due to the following reasons:

- Reason 1. The PoW and PoS approaches build their selection methodologies on the following factors: powerful computational resources and credit. Thus, these methods result in frequently selecting the same node as a miner in each mining session selection, resulting in the following issues:
    - Disappointed nodes: Nodes with limited computational resources will become disappointed and they may stop participating in other selection processes. Thus, this issue will lead to what is termed the 'lazy' issue.
    - Lazy node: Since only nodes with powerful computational capabilities will always succeed in being selected as miners, other nodes with limited computational capabilities will become disinclined to participate in the mining selection process.

  Therefore, PoW, PoS, and similar approaches seem to fail to guarantee fairness and motivation for all nodes during the miner selection process. From another angle, a hacker can use the selection methodology as a vulnerability point to launch his attack and violate the whole system. In light of this reality, the hacker can launch the following attacks:
    - Domination attack. Hackers equipped with supercomputing capabilities can join a blockchain system as legitimate nodes. Here, hackers will be assigned as miner nodes in many selection sessions, due to the powerful computational capabilities. Therefore, this grants them full control of most generated blocks, which can then perform various malicious behaviors.
    - Block Injection. Since the hackers have been assigned as miner nodes for many miner selection sessions, due to their supercomputing capabilities, they could successfully guess the puzzle algorithm. Here, a hacker can generate various attacks, such as generating a fake session for selecting a miner node and then selecting themselves to inject a fake block into a blockchain system.
- Reason 2. These approaches also provide fixed monetary rewards as compensation for the participating miner node. However, providing a merely fixed reward is not adequate to ensure the continuity of the growing blockchain system, as the blockchain will be measured by the size of its budget. For example, the total monetary rewards of the recruited miner nodes should not exceed the system budget. When the system budget is less than the allocated reward amount, the system will fail to recruit a miner for any new block. Thus, the blockchain will not be practical for adoption for most cloud-based systems.

Therefore, designing a dynamic mining protocol for ensuring fair and safe miner selection still requires serious effort. This encouraged us to conduct more investigations into the mining process. The dynamic mining protocol considers factors that PoW and PoS do not take into consideration. Having high computational power and resources is required to generate a new block of transactions. Studying the transaction specifications is non-trivial because, to the best of our knowledge, there is typically no consensus protocol model that handles them efficiently. In this work, transaction specifications have been deeply studied to find a firm expressive approach to categorize them in an obvious way. Transactions can be classified into three main categories when considering the processing time: low-sensitivity, soft-sensitivity, and hard-sensitivity levels. Indeed, the various sensitivity levels of transactions will assist us in determining the period for generating their blocks. For example, for a transaction classified as low-sensitive (to time), the period for generating a block could be longer than for a transaction with high-sensitive data (e.g., in real-time systems). Since computational capability plays a major role in affecting timing, the diverse computational capabilities of nodes could be compatible with the transactions' various sensitivities. Thus, the period for generating a new block of transactions with low sensitivity could be suitable for a node with modest computational capability [9,10].

Therefore, varying the critical times for generating blocks based on their sensitivity opens up a chance for all nodes to be selected as miners. However, there are still some challenges to successfully designing a dynamic mining protocol, which are threefold: (1) Theoretical challenge: the theoretical framework for illustrating the interaction between nodes and selecting miner methodology needs more investigation. (2) Interaction challenge: encouraging nodes to participate in the mining process is ongoing. (3) Dynamic reward challenge: a dynamic reward for various block generation needs more consideration.

To address the above challenges, an auction-based theory can be exploited for modeling a selection mechanism to produce advantageous proprieties such as fairness and profitability [11,12]. Auction-based theories have been proposed for most spectrum network applications. Nevertheless, these models are suitable for the blockchain model, especially for the miner selection node process, which depends on the node capability and transaction sensitivity type as well as the time period for the generated block.

Guided by such a challenge, this paper aims to design a dynamic mining protocol, called proof of fairness (PoF), to solve the problems of the mining process in the blockchain. To the best of our knowledge, this is the first work that builds a secure incentive mechanism based on selecting a miner framework. The PoF protocol is resilient with respect to transaction categorization and the block generation period. It guarantees fairness in node selection, reducing the burden of reward costs on the system. In addition, it preserves the node's private data during the selection process. The main contributions of this work are as follows:

- A reverse auction (RA) mechanism is proposed as an incentive mechanism to encourage all nodes to participate in the miner selection node process. In RA, each node draws its strategy based on its computational capability and claimed cost. It then bids with its best strategy in each selection process.
- An expressive language (EL) is designed to categorize transaction types based on their sensitivity to processing times to ensure compatibility with our miner selection process.
- A homomorphic signcryption (HSC) scheme is designed as a security and privacy scheme to protect the bidder's data confidentiality from being disclosed.
- Inclusive validation is conducted in order to illustrate the efficiency of the proposed PoF protocol through performance evaluation and numerical analysis, demonstrating that the proposed mechanism effectively satisfies the fairness in selecting a miner node, in comparison to current PoW and PoS consensus protocols.

The rest of this article is organized as follows: Section 2 discusses the related work. Section 3 explains the preliminary results while Section 4 shows the algorithm of PoF. Section 5 shows the security mechanism for the PoF. The security analysis is presented in Section 6. The performance evaluation results, advantages, and challenges of our proposed protocol are discussed in Section 7. Finally, Section 8 concludes the paper and highlights the future directions.

2. Related Work

Blockchain is a promising technology that needs more research to enhance its application in many fields. Blockchain runs on a peer-to-peer network, where each node in the network has its copy of the ledger. Any proposal or update to the ledger takes place according to consensus protocol. This shape of a distributed system avoids having a single point of failure [13]. The consensus protocol principle controls the transaction validation process, new block proposing, and the block verification process [6,14–16]. This work focuses on improving the efficiency of the blockchain model and consensus protocol.

Blockchain applications use different consensus protocols such as [15,17,18], PoS [19,20], DPOS [21], proof of space [22], PBFT [23], and ripple [24]. As mentioned previously, each type of consensus protocol has some disadvantages. On top of that, all existing consensus protocols, except PoQ, do not facilitate the participation of the newly joined nodes or nodes with low computational power, low coins, and low memory space. PoQ offers a fair chance to all nodes at the expense of work efficiency and accuracy. Therefore, this work introduces a dynamic consensus protocol, namely PoF, to balance between the task requirement and node capability.

Researchers have investigated and developed blockchain technology through many comprehensive survey articles, highlighting the development of all blockchain aspects. Some researchers have worked on blockchain applications in other domains such as education, healthcare, the internet, and smart cities [2,25–29]. Such comprehensive studies enable us to understand the differences in the specifications of data and transactions from one system to another. They also show the differences in cost management, security requirements, and sensitivity levels.

On the other hand, there are many research articles focusing on blockchain technical aspects, such as blockchain protocols, algorithms, and architecture. While some works investigate blockchain's security and privacy aspects [30–36]. The outcomes of these works guide the process of the PoF design.

3. Preliminaries

3.1. Blockchain Mechanism

Blockchain technology refers to data storage with ledgers of blocks that are linked to each other.

3.2. Consensus Mechanism

In the blockchain mechanism, assets, transactions, miners, and verifiers are the most important elements to be defined.

- Asset: This is the main database that includes data that are prepared for processing using reading and writing operations. It can be financial data, such as bank account information, or any kind of data.
- Transaction: This is a sequence of read-and-write operations that are approved or aborted together. A transaction is the fundamental unit of a blockchain. One example is to transfer a value from one address to another [10,37].
- Miner: This is a node that is responsible for verifying the transaction's validity and generating a new proposed block in the blockchain. However, a node will only be assigned as a miner according to the consensus protocol, such as PoW or PoS.
- Verifiers: This is a group of nodes that are in charge of verifying the validity of the newly generated block [38].

3.3. Description of Transactions Classification

In a blockchain system, data are processed as transaction payloads. Then a miner node validates transactions and stores them in a newly generated block in the system. The transaction consists of multiple read-and-write operations, where some transactions have more operations than others.

Additionally, some operations within different transactions depend on the results of others. Such dependency forces some concurrent transactions to wait until another one commits or aborts. For example, there are two transactions, $T1$ and $T2$. Transaction $T1$ transfers money, USD 100 from $account1$ to $account2$, where the original balance of $account1$ is USD 150, while $T2$ transfers money, USD 70 from $account1$ to $account3$. Clearly, if $T1$ commits (is approved) and the balance of $account1$ becomes USD 50, then $T2$ should abort, and vice versa. Therefore, the transaction processing period significantly affects the correctness of the execution.

Consequently, one of the most important aspects in generating a valid new block of transactions is to consider the sensitivity to the processing time. Thus, this work aims to design an algorithm for miner node selection that considers the node's capabilities in response to the transaction's time sensitivity level, using a dynamic and fair consensus protocol. Indeed, in this work, we classify the transaction payload into three main categories, as follows:

- Low sensitivity: Transactions with low-sensitive payloads could be assigned as low importance to the execution time. For example, the time period for generating a block of transactions with low sensitivity could be measured by days but less than a week.
- Soft sensitivity: Transactions with soft-sensitive payloads are more important, and a new block must be generated within a few hours to a maximum of a day. Thus, the node with the capability of generating a block in a day or less has a chance of being selected as a miner node.
- Hard sensitivity: Transactions with hard-sensitive payloads can be in real-time and are measured from seconds to a maximum of a few hours. The node with the capability of generating a block in a view of seconds to a few hours will have a high chance of being assigned as a miner node.

As discussed above, the transaction classification is considered the main factor in generating a new protocol related to selecting a miner node. The proposed PoF is mainly based on the required computational capabilities, according to the transaction classification.

3.4. Architecture Model

As shown in Figure 1, the architecture of the proposed model consists of the following entities:

- Auction mechanism: In the proposed model, one trusted node acts as the auctioneer (or multiple nodes vote on decisions), starting an auction system by announcing the task along with its time period specifications. This is done to select one winner from the connected participating nodes to act as the miner node. Indeed, for decentralization, the auctioneer role can consist of multiple nodes that vote on a decision.
- Bidding mechanism: Each node selects its best strategy according to its time capacity to generate a block. Each node sends its bidding value to the auctioneer, aiming to offer a lower cost to win the auction session.
- Winner selection mechanism: According to the nodes' bidding values, the auctioneer determines the winner that suggests the lower cost, and is capable of generating a block within the required period.
- Payment mechanism: The winning node will be rewarded financially after performing the required task.

The system model requires a trusted server that is in charge of the system parameters; then it will be suggested that the server goes offline for security purposes.

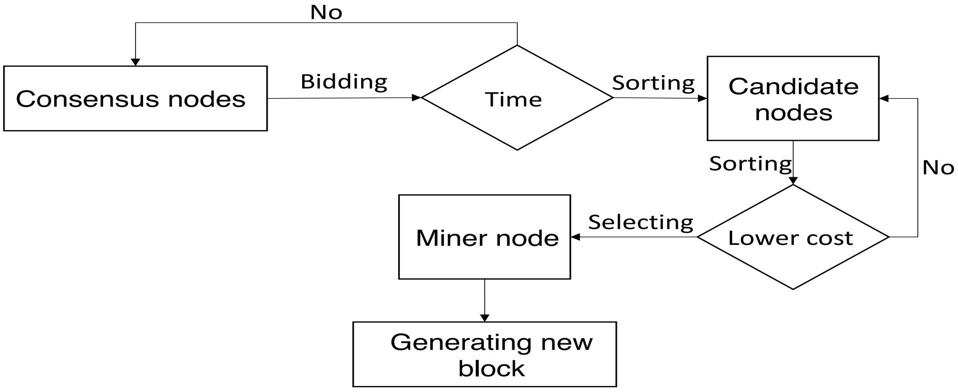


Figure 1. The proposed reverse auction system model for the miner selection process.

3.5. Auction Model

A reverse auction system is a type of auction model, but in a reverse way, in which a bidder offers a lower cost to win the game. Therefore, we designed a single-source reverse auction (ssR auction). It is formally defined as follows:

Definition 1. *(ssR auction). In the ssR auction, a buyer can buy a required good or service at a lower price from a single vendor.*

In the consensus system, we recognize that nodes are lazy and disappointed, so in an effort to optimize the reward system and to make nodes more interested, nodes can provide their capability resources in response to the announcement with more chances to be assigned as miners. Using the ssR auction model, we propose an *assignment system* as an *incentive mechanism*. It is formally defined as follows:

Definition 2. *(Assignment system). In the assignment system, the auctioneer, an_i, can assign the task, τ_i, to a trusted node, nd_i, with the ability to verify and generate a new block, b_i, within the required time, offering a lower reward, $r_i \in \mathcal{R}$. Note that i is a positive integer number, which is used as a local index for each group. Indeed, each node, nd_i, has n sets of bid strategies, such as $\mathcal{A}_i = \{\alpha_1, ..., \alpha_n\}$, where each $\alpha_i \in \mathcal{A}_i$ has a strategy of cost, c_i. Therefore, the cost function, F_C, for each strategy $\alpha_i \in \mathcal{A}_i$ guarantees that the cost should not exceed the reward, which is defined as follows:*

$$F_C(c_i) = \begin{cases} 1, & if c_i \leq r_i \\ 0, & otherwise \end{cases}$$

Definition 3. *(Reverse Monotonic Assignment [RMA]). In the proposed* assignment system, *the assignment rule, $\triangle$, is designed as a reverse monotone to ensure that a player (i)'s capability is monotonically decreasing when offering a lower cost, c_i, such that*

$$\triangle\ [\alpha_i(c_i), \sum_{i \notin j}^{n} \alpha_j(c_j)] = 1$$

where $\sum_{i \notin j}^{n} \alpha_j(c_j)$ denotes the bid strategies of all n players, except for player i. Consequently, the proposed *assignment system* is designed as a compatible incentive mechanism, which is formally defined as follows:

Definition 4. *(Incentive mechanism). The* assignment system *is a compatible incentive mechanism if each node, nd_i, plays with its best lower cost, c_i, value as its* dominant strategy, *α_i, to win the game.*

Definition 5. *(Dominant strategy). The strategy, α_i, with a cost, c_i, is called a dominant strategy if the utility, $\mathcal{U}_i$, of node nd_i consistently meets the following specifications:*

$$\mathcal{U}_i(\tau_i)[\alpha_i(c_i), \textstyle\sum_{i \notin j}^{n} \alpha_j(c_j)] \geq \mathcal{U}_i(\tau_i)[\overline{\alpha}_i(\overline{c}_i), \textstyle\sum_{i \notin j}^{n} \alpha_j(c_j)].$$

where $\overline{\alpha}_i(\overline{c}_i)$ is not the player (i)'s dominant strategy.

Definition 6. *(Rationality). Each player, i, is considered individually rational only if its utility, $\mathcal{U}_i(\tau_i)$, of playing its dominant strategy, $\alpha_i(c_i)$, is non-negative, such that*

$$\mathcal{U}_i(\tau_i)[\alpha_i(c_i), \sum_{i \notin j}^{n} \alpha_j(c_j)] \geq r_i.$$

Thus, in the *rationality* approach, the cost, c_i, of playing the dominant strategy, α_i, should be covered by the reward, r_i. This is the main prerequisite for each player, i, before being involved in the game.

Definition 7. *(Player's utility). The utility, $\mathcal{U}_i$, of each player, i, is formulated as follows:*

$$\mathcal{U}_i(\tau_i) \rightarrow (r_i - c_i) > 0$$

Definition 8. *(Budget). Given the reward, $\mathcal{R} = \{r_1, r_2, ..., r_n\}$, and the winning set of each auction session, $\mathcal{W}$, the total budget is*

$$\mathcal{B} = (\sum_{i \in \mathcal{W}} r_i) \geq 0.$$

3.6. Design Objective

Each node is considered selfish and always aims to win the session of the game, even if by acting in a malicious behavior, such as eavesdropping on other nodes' cost values to claim a lower false cost, c_i^f, and win the game session. This approach of malicious behavior will lead to misleading the auctioneer in distinguishing the node with a lower true cost, c_i^t, value. Subsequently, the objective of this work is to design a secure and fair auction (SFA) model that encourages nodes to participate in a rational and fair manner, within a secure auction process. However, to achieve our work's objective, the SFA model should solve the following issue:

Problem 1. Truthfulness cost [TC]: The purpose of proposing TC is to design a payment system, such that each participating node's utility cannot be guaranteed through reporting a false cost, c_i^f.

$$\mathcal{U}_i(c_i, c_{-i}) \leq \mathcal{U}_i(c_i^f, c_{-i}).$$

By solving the *TC* problem, we can ensure that the only way for each node to maximize its utility is by reporting its true cost, c_i^t, regardless of the other participants' node cost strategies, c_{-i}.

Problem 2. Secure Selecting Winner [SSW]: Given several nodes, $n = \{nd_1, ..., nd_n\}$, and a required period, $t_i = (t_s, t_e)$, the auctioneer aims to select a winner, $\mathcal{W} \subseteq n$, with a lower trust cost (c_i^t) that can generate a required block within the possible time periods, t^*.

Problem 3. Security and Privacy [SP]: The following information is considered as the node's private data.

- The time, t_i^*, is considered private data because if the time to generate the block is disclosed, it will lead to violating the node's capability.

- The cost, c_i^t, value is considered private data because if it is disclosed, it will lead to detecting the node's best strategy, which may help other nodes set up their best cost strategies to win the session.

Therefore, the node's (t_i^* and c_i^t) values must be protected among the nodes. Solving the *SP* problem will guarantee the confidentiality of the node's data.

4. Details of the PoF Protocol

This section describes the proposed model of PoF, as shown in Figure 2.

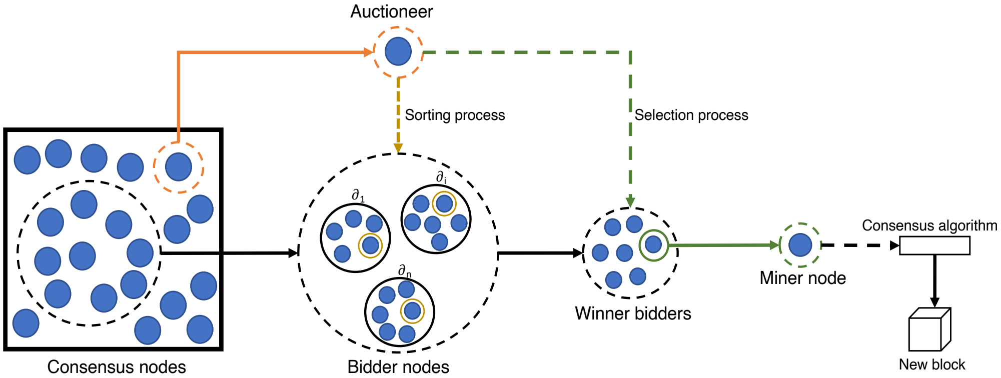


Figure 2. The miner selection system model.

4.1. Incentive Mechanism

The proposed PoF model is based on a proposed ssR auction to guarantee fairness among all participating nodes when selecting a node as a miner. In particular, for each new transaction, Γ_i, an auctioneer will start announcing it to all connected nodes. Thus, the interested node, nd_i, will start generating its bid values, $\mathcal{A}_i$, based on its dominant strategy in order to win the auction section. Once the auctioneer receives several $(\mathcal{A}_i)_{i=1}^n$ from n of interested nodes, $(nd_i)_{i=1}^n$, it will sort them in ascending order, according to their claimed costs. Therefore, the node with a lower cost will be selected as a miner node. The proposed PoF model is described in detail in the following phases.

4.1.1. Task Announcement

The auctioneer begins with the AS by announcing a block verification task, $\tau_i = \{\Gamma_i, t_s, t_e\}$, which includes the transaction, Γ_i, and time period, $[t_s, t_e]$, for verifying and generating the new block, b_i. Therefore, the computational capability requirements for each participating node should fit with the required time period, $[t_s, t_e]$, to verify and generate the new block, b_i.

4.1.2. Bidding

Each node, nd_i, with the capacity to participate in the τ_i is permitted to submit an auction bid as $\mathcal{A}_i = \{\hat{t}_i, c_i\}$. c_i represents the cost value for performing the τ_i and $\hat{t}_i = (t_i^s, t_i^e)$ denotes the period where node nd_i can perform the task τ_i within.

Each node, nd_i, has a different computational capability to perform the τ_i. Therefore, each node, nd_i, computes its period $\hat{t}_i$ according to its capability, as follows:

$$\hat{t}_i = [(t_s + \hat{t}_s^i) + (t_e - \hat{t}_e^i)]^\lambda$$

where $(0 \geq \hat{t}_s^i \leq t_s)$ is the expected beginning time of node nd_i to start performing the τ_i, which is computed as the following:

$$\hat{t}_s^i = \frac{t_s^i - t_s}{(t_e - t_s)}$$

In addition, $(0 \geq \hat{t}_e^i \leq t_e)$ is the expected time of node nd_i to finish performing the τ_i, which is computed as follows:

$$\hat{t}_e^i = \frac{t_e - t_e^i}{(t_e - t_s)}$$

The λ is an influence time factor, which is $\lambda > 0$. Note that nodes might have overlapping time periods.

4.1.3. Selecting Winner Bidder (SWB)

The objective of the auctioneer is to recruit a suitable node, nd_i, to be in charge of generating a new block, b_i, during the range of period $[\hat{t}_i \subset (t_s, t_e)]$ and with the lowest possible reward, r_i.

However, the node times $(\hat{t}_i)_{i=1}^n$ might overlap. To easily illustrate time conflicts, the following notations are presented:

- n denotes the number of bidder nodes.
- $g(f, u)$ denotes the conflict graph, where f represents the collection of number (n) bidders $\mathcal{A} = \{\mathcal{A}_1, ..., \mathcal{A}_n\}$ and u denotes a k set of edges $u = \{u_1, ..., u_k\}$, where each u_i represents m sets of $(\mathcal{A})_{i=1}^m$ that have the same period.

However, this makes (Problem 2) an NP-complete problem, which hinders the auctioneer from selecting the node as a winner. Therefore, from each $u_i \in u$, we select only one node, nd_i, with a lower cost, c_i, as the winner of its edge, u_i, and then sort all winners from all these edges, $(u_i)_{i=1}^k$, in a group, ∂. Finally, the auctioneer only needs to select one, $nd_i \in$, with a lower cost, c_i, as the final winner. The suggested greedy process is performed through the following steps:

- Step 1: For each u_i, sort nodes based on their neighbors' costs, $(c_i)_{i=1}^m$, as follows:

$$u_i(c_1)^\alpha < ... < u_i(c_i)^\alpha < ... < u_i(c_m)^\alpha.$$

  A node with a lower cost, c_i, will be selected as a winner in its edge, u_i.
- Step 2: Sort all winning nodes from each $(u_i)_{i=1}^k$ according to their cost, c_i, such that

$$u_1(c_i)^\alpha < u_2(c_i)^\alpha < ... < u_i(c_i)^\alpha < ... < u_k(c_i)^\alpha.$$

  where α represents the influence factor of variance importance for generating the new block. A small α causes the generation of the new block to be more important.
- Step 3: The lowest cost in Step 2 will be selected as a final winner, fulfilling the following requirement:

$$\triangle \cdot \hat{t}_i \subset (t_e - t_s), \triangle \cdot c_i \geq r_i$$

  Accordingly, Algorithm 1 represents the SWB process.

Lemma 1. *The proposed SWB guarantees a fair selection methodology among all participating nodes, $(nd_i)_{i=1}^n$.*

Proof. All nodes are first sorted according to their overlapping time to a set of groups, $u = \{u_1, ..., u_k\}$. Thus, we sort all winning nodes from each $(u_i)_{i=1}^k$ according to their cost, c_i, demonstrating the proposed reverse monotonic assignment [RMA] of the assignment system. □

4.1.4. Fair Reward Mechanism

A fair reward method is designed to guarantee individual rationality, to ensure that every node, nd_i, bids by its truthful cost, c_i^*, as its domain strategy. The fair reward method is formally defined as follows:

Algorithm 1 SWB Algorithm

```
Input: Task $\tau$ and bidder set $\mathcal{A} = \{\alpha_1, ..., \alpha_n\}$.
Output: Assignment $\triangle$.
$\tau \leftarrow \mathcal{A}$
for $i = 1 : m$ do
    $\triangle = 0$;
end for
Sort bidders according to $c_i$,
$u_i(c_1)^\alpha < ... < u_i(c_i)^\alpha < ... < u_i(c_m)^\alpha$,
$\tau \leftarrow \tau \backslash (u_i(c_i)^\alpha \bigcup i)$
Sort bidders according to $c_i$,
$u_1(c_i)^\alpha < u_2(c_i)^\alpha < ... < u_i(c_i)^\alpha < ... < u_k(c_i)^\alpha$,
$i \leftarrow argmin(c_i)$
return $\triangle = 1$
```

Definition 9. *(Fair reward rule). The fair reward scheme, R, can be defined as follows:* $r_i = c^*_{i+1}$ *for a winning node,* nd_i*, and* $r_i = 0$ *otherwise.*

Lemma 2. *If the assignment system satisfies monotonicity, then there is a winning node,* nd_i*, in each* u_i *group, and from all these winners, there must be a winner,* nd_i*, with the critical neighbor node,* nd_{i+1}*, such that* $(c^*_i < c^*_{i+1})$*.*

Proof. In each u_i group, we sort nodes $(nd_i)^m_{i=1}$ incrementally based on their cost values, $(c_i)^m_{i=1}$. It is easy to locate the node, nd_i, with a lower cost value, c_i. In addition, by sorting all these winning nodes according to their costs $(c^*_i)^n_{i=1}$, it is also easy to demonstrate the node, nd_i, with a lower c^*_i and its critical neighbor node, nd_{i+1}, with $(c^*_i < c^*_{i+1})$. □

Lemma 3. *The ssR auction mechanism can be designated as fair if a winning node obtains a fair reward, such that* $r_i > c^*_i$*.*

Proof. As the monotonous assignment rule provides a fair selection rule, it is easy to generate a fair reward, as illustrated in Algorithm 2. □

Algorithm 2 Fair reward algorithm

```
Input: Group bidders $(u_i)^k_{i=1}$, conflict graph, $g$, and assignment $\triangle$.
Output: Reward $R$
for $i = 1 : i \in u_i$ do
    if $\triangle = 0$ then
        $r_i = 0$
    else
        $\triangle \leftarrow i \setminus \{i\}$
        while $\triangle \neq \emptyset$ do
            $i \leftarrow lower(c^*_i)$
            if $i < nd^*_{i+1}$ then
                $r_i = c_(i + 1)^*$
            end if
            $\triangle \leftarrow \triangle \backslash (nd_i \bigcup u_i)$
        end while
    end if
end for
return $R = \{r_1, r_2, ..., r_k\}$
```

4.2. Computational Complexity

The computational complexity of the greedy assignment algorithm is investigated in terms of the cost of running time. Let $g = (f, u)$ be a conflict graph with n bidder nodes and their periods, t^*. The computational complexity comes from the following processes:

- Sorting process: In this process, the auctioneer will sort all n bidder nodes according to their similarity times. For example, it first sorts m bidder nodes that have similar time periods in their bids in a group, u_i. This process will take $m|u|$ time to sort each group, u_i, according to their similarity time, and it will take $O(k \, log \; u)$ time to sort all n bidder nodes, such that $u = \{u_1, ..., u_k\}$.
- Selection process: In this process, the auctioneer selects the bidder node with the lowest cost from each $|u_i|$, which will take $O(k|u|)$ time.

Therefore, the overall complexity is $O(k \, log \; k + |u|)$.

Theorem 1. *The proposed PoF successfully achieves a selected miner node with high satisfaction and lower payment. Thus, (**Problem 1**) and (**Problem 2**) are solved.*

Proof. The fairness of selecting the node and the reward rules are determined according to Lemmas 1–3. A winning node will be rewarded with a non-negative reward and the other node will receive no reward. □

5. Proposed Security Mechanism for PoF

5.1. Security and Privacy Scheme

To protect the node bid values from being disclosed during the auction process, this work designed a homomorphic signcryption (HSC) scheme, which is compatible with the proposed PoF protocol. The proposed HSC scheme enables each bidder to signcrypt its strategy value and submit it to the auctioneer as a ciphertext. Since the HSC scheme satisfies a homomorphic concept, the auctioneer can sort bidders that overlap with similar time expectations without revealing their private costs. The proposed HSC scheme is based on the following:

5.1.1. Bilinear Group

The proposed HSC scheme is executed on a bilinear mapping methodology [39], which is defined as follows:

Definition 10. *(Bilinear mapping). Bilinear mapping is an admissible bilinear pairing, such that* $\hat{e} : \Im_1 \times \Im_1 \rightarrow \Im_2$, *which involves mapping over elliptic curves* [40]*. The properties of* $\hat{e}$ *are illustrated as follows:*

- *Bilinearity property: Let* $g_1, g_2, g_3 \in \Im_1$ *and* $x, z \in Z_q^*$, *such that:*
  - $\hat{e}(g_1, g_2 + g_3) = \hat{e}(g_1, g_2)\hat{e}(g_1, g_2) \rightarrow \Im_2$.
  - $\hat{e}(xg_1, zg_2) = \hat{e}(g_1, g_2)^{xz} = \hat{e}(zg_1, xg_2) \rightarrow \Im_2$.
- *Non-degeneracy property: Let* $g_1 \neq 0, g_2 \neq 0 \in \Im_1$, *such that* $\hat{e}(g_1, g_2) \rightarrow \mathbb{G}_2 \neq 1$.
- *Computational ability property:* $\hat{e}$ *is efficiently computable.*

Note that $\Im_1$ is an additive group and $\Im_2$ is a multiplicative group of order q for some large prime q values.

5.1.2. Complexity Assumptions

We describe the discrete logarithm problem related to the proposed HSC as follows:

Definition 11. *Computational Diffie–Hellman* (CDH) *Problem. Given* $g, xg, zg \in \Im_1$ *and* $x, z \in Z_q^*$, *where* g *is a generator of* $\Im_1$, *the* (CDH) *problem is used to compute* $xzg \in \Im_1$ *within polynomial time.*

Definition 12. *Decisional Bilinear Diffie–Hellman* $(DBDH)$ *Problem. Given* $g, xg, zg, yg \in \Im_1$, *and* $x, z, y \in Z_q^*$, *where* g *is a generator of* $\Im_1$, *the* $(DBDH)$ *problem is used to decide whether* $h = \hat{e}(g, g)^{xzy}$ *within polynomial time, where* $h \in \Im_2$.

5.2. Details of the PoF Security and Privacy Scheme

The PoF security and privacy scheme is described in detail, according to the following phases:

Phase 1: Initialization system: The trust server generates the system parameters $(\Im_1, \Im_2, \hat{e}, q)$ based on the security parameter, p, where g is a generator point of $\Im_1$. Then the trust server selects the master private key, $\varphi \in Z_q^*$, and computes the corresponding public key, $K_{pub} = \varphi g$. In addition, it chooses a secure hash function, $H_1 : \{0, 1\}^{256} \rightarrow \Im_1$. The trust server finally publishes the public parameters as $PM = (\Im_1, \Im_2, q, g, \hat{e}, K_{pub}, H_1)$.

Based on the system's public parameters, PM, each node, i, selects a random number, $x_i Z_q^*$, as its private key and then computes its corresponding public key, $PK_i = x_i g$. In addition, the node, i, selects a random number, $q_i \in Z_q^*$, as a salt key to generate its pseudo-identity as $S_i = q_i H_1(ID_i)$.

Phase 2: Transaction generating: When node nd_i generates a new transaction, Γ_i, to be added to the blockchain system, it sends Γ_i to the auctioneer, nd_{a_i}, with its importance rate, μ, to generate a Γ_i block. The node, nd_i, will select $y_i \in Z_q^*$ and compute both $A_{i1} = y_i PK_{a_i}$ and $A_{i2} = E_{AES}[y_i g, (\Gamma_i, \mu)]$. It then sends (A_{i1}, A_{i2}) to the auctioneer, nd_{a_i},.

Phase 3: Auction: Once receiving (A_{i1}, A_{i2}), the auctioneer, nd_{a_i}, uses its private key to obtain (Γ_i, μ), as $(\Gamma_i, \mu) = D_{AES}[\frac{1}{x_{a_i}} A_{i1}, A_{i2}]$. Based on μ, the auctioneer, nd_{a_i}, determines the range of time (t_s, t_e) to generate a new block, $\mathbb{B}_i$ for Γ_i and π_i, as a task's identity number. It then generates a task, $\tau_i = \{\Gamma_i, t_s, t_e\, \pi_i\}$, and identifies itself to all connected nods by generating a digital signature on τ_i as,

$$\alpha_i(\tau_i) \leftarrow Sign(\tau_i, x_{a_i}),$$

Finally, it starts an opening reverse auction session by announcing $(\tau_i, \alpha_i(\tau_i))$.

Phase 4: Bidding: Any node, nd_i, that is interested in participating in the auction system will first need to verify the τ_i validity, as follows:

$$\alpha_i(\tau_i)' \leftarrow \mathbf{Verify}(PK_{nd_a}, \alpha_i(\tau_i)).$$

The interested node, nd_i, then generates its bid $\mathcal{A}_i = \{\hat{t}_i, c_i\}$ values based on its computational capability for participating in τ_i. Thus, node nd_i will signcrypt its $\mathcal{A}_i$, as shown below. It randomly selects $r_i \in Z_q^*$ and computes the following:

- $E_{i1} = r_i PK_{nd_a}$
- $E_{i2} = \hat{t}_i \pi_i g$
- $E_{i3} = \frac{1}{r_i}(\pi_i K_{pub} + E_{i2})$
- $E_{i4} = r_i(PK_{nd_a} + E_{i3})$
- $E_{i5} = r_i \pi_i (c_i PK_{nd_a} + K_{pub}) + r_i g$
- $E_{i6} = (x_i + r_i) E_{i5}$
- $E_i = (E_{i1}, E_{i4}, E_{i5}, E_{i6})$

Each node, nd_i, then sends its (E_i, S_i) to the auctioneer, nd_{a_i}.

Phase 5: Verification: Once receiving $(E_{i6})_{i=1}^n$ from n nodes, the auctioneer, nd_{a_i}, aggregates all ciphertexts $(\sum_{i=1}^n E_{i6})$, of the received public keys, $(PK_{nd_i})_{i=1}^n$, and verifies them simultaneously, such that the auctioneer, nd_{a_i}, only accepts $(\sum_{i=1}^n E_{i6})$ if the following equation holds:

$$V_i = \frac{\prod_{i=1}^n \hat{e}(E_{i6}, g)}{\prod_{i=1}^n \hat{e}(E_{i5}, (\frac{1}{x_{nd_a}} E_{i1} + PK_{nd_i}))} = 1 \qquad (1)$$

If the V_i holds, the auctioneer, nd_{a_i}, accepts $(E_i)_{i=1}^n$ as a valid ciphertext from n nodes and will perform the next phase.

Lemma 4. *The verification phase is complete.*

Proof. $V_i = 1$, since:

$$\begin{aligned} V_i &= \frac{\prod_{i=1}^{n} \hat{e}(E_{i6}, g)}{\prod_{i=1}^{n} \hat{e}(E_{i5}, (\frac{1}{x_{nd_a}} E_{i1} + PK_{nd_i}))} \\ &= \frac{\prod_{i=1}^{n} \hat{e}((x_i + r_i)E_{i5}, g)}{\prod_{i=1}^{n} \hat{e}(E_{i5}, (\frac{1}{x_{nd_a}} E_{i1} + PK_{nd_i}))} \\ &= \frac{\prod_{i=1}^{n} \hat{e}((x_i E_{i5}, g)\hat{e}(r_i E_{i5}, g)}{\prod_{i=1}^{n} \hat{e}(E_{i5}, (\frac{1}{x_{nd_a}} (r_i PK_{nd_a}) + PK_{nd_i}))} \\ &= \frac{\prod_{i=1}^{n} \hat{e}((E_{i5}, x_i g)\hat{e}(E_{i5}, r_i g)}{\prod_{i=1}^{n} \hat{e}(E_{i5}, (\frac{1}{x_{nd_a}} (r_i x_{nd_a} g) + PK_{nd_i}))} \\ &= \frac{\prod_{i=1}^{n} \hat{e}(E_{i5}, PK_{nd_i})\hat{e}(E_{i5}, r_i g)}{\prod_{i=1}^{n} \hat{e}(E_{i5}, (r_i g + PK_{nd_i}))} \\ &= \frac{\prod_{i=1}^{n} \hat{e}(E_{i5}, PK_{nd_i})\hat{e}(E_{i5}, r_i g)}{\prod_{i=1}^{n} \hat{e}(E_{i5}, r_i g)\hat{e}(E_{i5}, PK_{nd_i})} \\ &= \frac{\prod_{i=1}^{n} \hat{e}(E_{i5}, PK_{nd_i} + r_i g)}{\prod_{i=1}^{n} \hat{e}(E_{i5}, PK_{nd_i} + r_i g)} = 1 \end{aligned}$$

□

Phase 6: Sorting process: The auctioneer, nd_{a_i}, will first sort the nodes that have submitted the same timestamp, as follows:

- For each node, nd_i, the auctioneer, nd_{a_i}, computes

$$\omega_i = \frac{\hat{e}(K_{pub}, E_{i4})}{\hat{e}(K_{pub}, E_{i1} + \pi_i K_{pub})}$$

- For each node, nd_i, the auctioneer, nd_{a_i}, computes the following:
  - $T_{i1} = \hat{e}(K_{pub}, \pi_i(t_s)g)$
  - $T_{i1} = \hat{e}(K_{pub}, \pi_i(t_e)g)$

  It then accepts ω_i as a valid timestamp if the following equation is held:

$$T_i \leftarrow T_{i2} \leq \omega_i \leq T_{i2}$$

- Sort m sets of bidder nodes $(nd_i)_{i=1}^{m}$ that have valid $(T_i)_{i=1}^{m}$. Group the same $(\omega_i)_{i=1}^{m}$ in a set, ∂_i, such that

$$\partial_i = \{\omega_i, ..., \omega_m\}.$$

Theorem 2. *The auctioneer, nd_{a_i}, can sort nodes according to their time availability $(\omega_i)_{i=1}^{n}$, but it is not able to disclose the nodes' time availability. This is because the auctioneer generates a second ciphertext of $(\omega_i)_{i=1}^{n}$ from the first ciphertexts $(E_{i4}, E_{i2})_{i=1}^{n}$. Therefore, the participating nodes' private data are protected.*

Proof. The ω_i ciphertext is completely encrypted, which is generated under CDH and $DBDH$ assumptions. The correctness of ω_i is approved as follows:

$$\begin{aligned}\omega_i &= \frac{\hat{e}(K_{pub}, E_{i4})}{\hat{e}(K_{pub}, E_{i1} + \pi_i K_{pub})} \\ &= \frac{\hat{e}(K_{pub}, r_i(PK_{nd_a} + E_{i3})}{\hat{e}(K_{pub}, E_{i1} + \pi_i K_{pub})} \\ &= \frac{\hat{e}(K_{pub}, (r_i PK_{nd_a} + r_i E_{i3})}{\hat{e}(K_{pub}, E_{i1} + \pi_i K_{pub})}\end{aligned}$$

$$\begin{aligned}&= \frac{\hat{e}(K_{pub}, r_i PK_{nd_a})\hat{e}(K_{pub}, r_i E_{i3})}{\hat{e}(K_{pub}, E_{i1} + \pi_i K_{pub})} \\ &= \frac{\hat{e}(K_{pub}, r_i PK_{nd_a})\hat{e}(K_{pub}, r_i[\frac{1}{r_i}(\pi_i K_{pub} + E_{i2})])}{\hat{e}(K_{pub}, E_{i1} + \pi_i K_{pub})} \\ &= \frac{\hat{e}(K_{pub}, r_i PK_{nd_a})\hat{e}(K_{pub}, (\pi_i K_{pub} + E_{i2}))}{\hat{e}(K_{pub}, E_{i1} + \pi_i K_{pub})} \\ &= \frac{\hat{e}(K_{pub}, E_{i1})\hat{e}(K_{pub}, \pi_i K_{pub})\hat{e}(K_{pub}, E_{i2})}{\hat{e}(K_{pub}, E_{i1} + \pi_i K_{pub})} \\ &= \frac{\hat{e}(K_{pub}, E_{i1} + \pi_i K_{pub})\hat{e}(K_{pub}, \hat{t}_i \pi_i g)}{\hat{e}(K_{pub}, E_{i1} + \pi_i K_{pub})} \\ &= \hat{e}(K_{pub}, \hat{t}_i \pi_i g)\end{aligned}$$

□

Phase 7: Selecting the winning node: After sorting bidders to a set, k, of groups $(\partial_i)_{i=1}^{k}$, the auctioneer, nd_{a_i}, then selects the winning node from each group $(\partial_i)_{i=1}^{k}$. Accordingly, it will sort all winners incrementally, according to their costs, and then select the lower cost as the winner of winners, and determine the respective payments as follows:

- Monotonic assignment: The auctioneer, nd_{a_i}, uses its private key (x_{nd_a}) to obtain a cost, c_i, value from ciphertexts, E_{i5}, for each node, $nd_i \in \partial_i$, as follows:
  - $D_i = \dfrac{1}{x_{na_a}} E_{i1}$
  - $C_i = \dfrac{E_{i5}}{(\pi_i + 1)D_i}$

  Thus, C_i is an encrypted value of the real cost, c_i; hence, the auctioneer, nd_{a_i}, knows nothing about the node, nd_i,'s real cost, c_i, value. Thus, the node, nd_i,'s private cost, c_i, is still secure.
  It then starts sorting nodes $(nd_i)_{i=1}^{m} \in \partial_i$ according to their $(C_i)_{i=1}^{m}$ in ascending order, such that

  $$(C_1^{\alpha} < C_2^{\alpha} < ... < C_m^{\alpha}) \in \partial_i$$

  The auctioneer, nd_{a_i}, selects the node with a lower cost (C_i) and then deletes the other nodes. It replays this step for all groups $(\partial_i)_{i=1}^{k}$. Finally, it sorts all winning nodes from each $(\partial_i)_{i=1}^{k}$ according to their costs, C_i, such that

  $$\partial_1(C_i)^{\alpha} < \partial_2(C_i)^{\alpha} < ... < \partial_i(C_i)^{\alpha} < ... < \partial_k(C_i)^{\alpha}.$$

  Therefore, the auctioneer, nd_{a_i}, selects the lower cost value $(\partial_i(C_i)^{\alpha})$ as a winner and then assigns the winning node, nd_i, as a miner node for this session.

- Critical payment: For the winning node, $nd_i \in \partial_i$, the auctioneer, nd_{a_i}, determines its critical neighbor $nd(i+1)^* \in \partial^*_{i+1}$ by running *Algorithm 2*. It then sends the $\partial_{i+1}(C_{i+1})^*$ to the smart payment contract.

Theorem 3. *The auctioneer, nd_{a_i}, can sort nodes according to their costs $(c_i)^k_{i=1}$ in each group of nodes without disclosing the real cost information. The sorting process is performed on a second ciphertext of $(C_i)^k_{i=1}$, which is generated from the first ciphertexts $(E_{i1}, E_{i5})^k_{i=1}$. Thus, the participating nodes' private data are protected.*

Proof. The C_i ciphertext is completely encrypted, which is generated under the CDH assumption. The correctness of C_i is approved as follows:

$$\begin{aligned} C_i &= \frac{E_{i5}}{(\pi_i+1)D_i} \\ &= \frac{E_{i5}}{(\pi_i+1)\dfrac{1}{x_{na_a}}E_{i1}} \\ &= \frac{r_i\pi_i(c_iPK_{nd_a}+K_{pub})+r_ig}{(\pi_i+1)r_ig} \\ &= \frac{(c_ix_{nd_a}+\varphi)r_i\pi_ig+r_ig}{\pi_ir_ig+r_ig} \\ &= (c_ix_{nd_a}+\varphi) \end{aligned}$$

□

Theorem 4. *The proposed PoF protocol satisfies the fairness and competition among nodes within a secure environment during the process of selecting a miner node.*

Proof. The bidder, i, is determined according to Lemmas 1–3. In line with the miner node selection rule, every node submits its true strategy within a secure system, and the winning node, nd_i, is selected according to the lower cost among all nodes and will be paid with $c_i^{t*} > c_i^t$, without disclosing the c_i^t value. Notably, the time required for block generation helps all nodes to calculate their best strategy, allowing them to bid in a miner selection session with a high chance of being selected or obtaining non-negative utility if not selected. Therefore, the fairness and individual competitive manner within a secure environment can be proved. □

6. Security Analysis

This section illustrates the security analysis of the proposed homomorphic signcryption (HSC) scheme, which is designed for security and privacy for the proposed PoF protocol. Thus, in this section, we will prove the ability of the HSC scheme to solve (Problem 3) according to the following security and privacy properties.

- Privacy and integrity: In compliance with Definition 11, the bid value is signcrypted under the CDH assumption, which prevents a malicious node, nd_j, from disclosing or even modifying the content of any target bid value. Since each participating node, nd_i, signcrypts its bid value using a receiver's public key, as well as random secure, r_i, to calculate $(E_{i1}, E_{i3}, E_{i4}, E_{i5})$, the malicious node, nd_j, will not be able to reveal or forge any ciphertext of $(E_{i1}, E_{i3}, E_{i4}, E_{i5})$ within a polynomial time without knowing the receiver's private key and the random value, r_i. In addition, the malicious node, nd_j, will not be able to generate a fake signature on any targeted bid value without reaching the sender's private key that is utilized to generate $E_{i6} = (x_i + r_i)E_{i5}$.

Lemma 5. *The proposed PoF protocol achieves privacy-preservation and integrity simultaneously.*

Proof. As the (E_i) ciphertext is signcrypted with a secret random key, r_i, under two public keys (PK_{nd_a}, K_{pub}), it is difficult for an adversary to reveal the real r_i and both secure keys, which are the auctioneer's private key, x_{nd_a}, and the φ of the public key, K_{pub}. In addition, (E_i) is computed under the CDH assumption, which is difficult to be computationally solved in polynomial time. Therefore, the adversary or malicious node will not be able to modify or disclose the bidder's real cost or time period for generating a block. □

- Authentication: The authentication property is guaranteed since the proposed security and privacy scheme is based on the signcryption technique. Each participating node, nd_i, authenticated itself by generating a signature on its encrypted bid, using its private key as $E_{i6} = (x_i + r_i)E_{i5}$. However, the auctioneer, nd_{a_i}, only accepts the E_i ciphertext if it is valid according to the verification step, as illustrated in the following equation:

$$V_i = \frac{\hat{e}(E_{i6}, g)}{\hat{e}(E_{i5}, \frac{1}{x_{nd_a}}E_{i1})\hat{e}(E_{i5}, PK_{nd_i})} = 1.$$

In this work, we exploit an aggregation methodology to reduce the computational cost as used in Equation (1).

Lemma 6. *The proposed PoF protocol achieves authentication properties.*

Proof. Each registered bidder node, nd_i, generates a signcrypted ciphertext under the CDH. At the same time, only a registered auctioneer, nd_{a_i}, can authenticate the bidder node, nd_i, legitimacy with a valid E_i ciphertext by verifying the E_i validation, using its private key under the $DBDH$. Therefore, the authentication goal of verifying the nodes' validation is achieved. □

- Secure winner selection: The proposed security and privacy scheme is based on a signcryption technique with a concept of a homomorphic technique in order to propose a security mechanism for PoF. According to Definitions 11, and 12, the auctioneer, nd_{a_i}, first sorts all bidders according to their time period, sorting bidders that provide the same period in a group and selecting a bidder with a lower cost as a winner from each group. Finally, it sorts all winners in descending order, according to their costs, and then selects a lower cost. The auctioneer, nd_{a_i}, performs all selecting winner processes without disclosing the bidders' actual time periods and their real costs under the $DBDH$.

Lemma 7. *The proposed PoF protocol ensures secure winner selection.*

Proof. Each registered bidder node, nd_i, signcrypts its value with a secret random key, r_i, and (PK_{nd_a}, K_{pub}). In contrast, according to Definition 12, only the auctioneer, nd_{a_i}, can sort all bidders based on their periods, as it sorts bidders that provide the same time period in a group by computing $(\omega_i)_{i=1}^{n}$. The time value of each bidder is encrypted under r_i and the public key K_{pub} to generate the ciphertext of time value as follows:

$$E_{i4} = r_i[PK_{nd_a} + \frac{1}{r_i}(\pi_i K_{pub} + \hat{t}_i \pi_i g)],$$

The auctioneer, nd_{a_i}, will perform a sorting process among all bidders without revealing their actual time periods as time $(\hat{t}_i)$ is preserved under r_i and K_{pub}. In addition, the auctioneer, nd_{a_i}, selects a winner from all winner bidders by decrypting the ciphertext using its private key to compute $D_i = \frac{1}{x_{nd_a}}E_{i1}$. Even with computing D_i, the bidder's private cost is still encrypted under the public key (PK_{nd_a}), where the proposed scheme is designed in a way to enable the auctioneer, nd_{a_i}, to select a winning node without disclosing its real cost value. Therefore, the goal of a secure winner selection is achieved. □

Theorem 5. *The proposed HSC scheme successfully achieved all security and privacy properties for the proposed PoF protocol from solving (Problem 3).*

Proof. According to Lemmas 5 and 6, the proposed HSC scheme is efficiently capable of protecting the t_i^* and c_i^t of each participating node during the auction session from being disclosed or modified. In contrast, the auctioneer node can select an authenticated node as a winning node without revealing their private data. Therefore, we can claim that (Problem 3) has efficiently been solved with the proposed HSC scheme. □

7. Performance Evaluation

This section first shows a numerical analysis of the fairness of PoF, which is investigated through the satisfaction of the nodes and the auctioneer. Secondly, this section presents the experimental simulation results and analysis, starting by illustrating the parameters and the platform used in this evaluation. Then, a comparison among the PoF with PoW and PoS protocols will be displayed in terms of the efficient mining processes. Finally, cryptographic cost terms and communication overhead will be evaluated.

7.1. Numerical Analysis

In this part, we briefly illustrate a numerical analysis for the proposed PoF protocol in terms of fairness in selecting a miner node. Thus, measuring satisfaction among participating nodes is a practical methodology for numerically analyzing the proposed PoF's efficiency.

Therefore, in this analysis, we set N number of nodes as $ND = \{nd_1, ..., nd_N\}$ and a task, $\tau = \{\Gamma, t_s, t_e\}$, (as mentioned earlier, t_s and t_e mean the task's starting and ending times).

7.1.1. participating Node's Satisfaction

Node satisfaction is achieved when most nodes have a higher chance to participate in the auction session. Therefore, the overall node satisfaction (ND^{Stf}) is measured based on the following equations:

$$ND^{Stf} = [\frac{z}{N}] \tag{2}$$

where z is the total number of nodes that are capable of processing τ within a specific period of time, $t_s - t_e$. Thus, ND^{Stf} shows the ratio of the nodes that participate in the auction of τ among all nodes, N. Indeed, we find $Time_c$, which is a set of nodes that are capable of processing τ within the time limits, where the size of $Time_c$ is z, and it is computed based on the following equations:

$$Time_c.z \to \left[\sum_{i=1}^{k}(nd_i.\tau^{\varepsilon})\right] || [t_s - t_e] \tag{3}$$

where "$||$" refers to time duration, and ε is a threshold for measuring task τ sensitivity, as follows:

- $\varepsilon = 1$ denotes τ with hard sensitivity, where there is a short period of execution.
- $\varepsilon = 2$ denotes τ with soft sensitivity, where there is a medium period of execution.
- $\varepsilon = 3$ denotes τ with low sensitivity, where there is a long period of execution.

In addition, k is an integer number that represents the number of nodes that have processing capabilities to process the task, τ (without considering the time limits). Therefore, we find the set of nodes that have processing capabilities, denoted as P_c, and the size of P_c is k. Indeed, for all nodes, N, if the node's capabilities ($nd_i.cp$) are more than the task's cost ($\tau^{\varepsilon}.c$), it will be a member of P_c, as shown in the following equations:

$$P_c.k \to \left[\sum_{i=1}^{N}\left(\frac{nd_i.cp}{\tau^{\varepsilon}.c}\right)\right] > 0 \tag{4}$$

In short, among N values, we first find the number of nodes that have processing capabilities to process the task, τ; let us say k nodes. Secondly, among k nodes, we find the number of nodes that can process the τ within time limits; let us say z nodes. Therefore, it is clear that $(z \subseteq k \subseteq N)$.

For example, for a highly sensitive task, τ^1, only z nodes can participate; those nodes can process this task on time. Therefore, the ratio of node participation will be (z/N). However, for low-sensitivity tasks τ^3, almost all nodes can participate, which makes $z \approx N$, and increases node satisfaction to almost 1.

In contrast, other consensus protocols do not reach such node satisfaction. For example, within PoS, if one node stakes a high amount, all nodes that cannot stake a higher amount will drop off. If this scenario keeps happening, those nodes will be disappointed instead of satisfied. The same example somehow applies to the PoW and other consensus protocols.

7.1.2. Auctioneer's Satisfaction

The auctioneer's satisfaction, AN^{Stf}, is achieved when n of the announced tasks $\tau = \{\tau_1, ..., \tau_n\}$ are processed within the required time periods and with a lower payment of rewards $(\sum_{i=1}^{n} r_i)$. In fact, the total paid rewards should not exceed the budget, $\mathcal{B}$. In addition, having more participating nodes increases node competition, which enables the auctioneer to select the node with minimum bids and, consequently, minimum rewards. Therefore, the auctioneer's satisfaction can be measured as the following equations:

$$AN^{Stf} \rightarrow \left[\left(\frac{z}{\sum_{i=1}^{n} (\tau_i^{\varepsilon})} \right), \left[\left(\sum_{i=1}^{n} \tau_i^{\varepsilon} * \sum_{i=1}^{n} r_i \right) \right] \leq \mathcal{B} \right] \tag{5}$$

Therefore, AN^{Stf} linearly increases when more nodes participate, and it is significantly affected by the n sizes of processed tasks, $\sum_{i=1}^{n} (\tau_i^{\varepsilon})$, and rewards $(\sum_{i=1}^{n} r_i)$.

On the other hand, some other consensuses protocols assign a fixed reward, regardless of the task cost, which challenges fairness and the auctioneer's satisfaction. Some consensuses protocols require solving a complex computation (e.g., mathematical puzzle) in addition to task processing, which is another overhead that requires corresponding rewards.

7.2. Parameter Setting and Platform

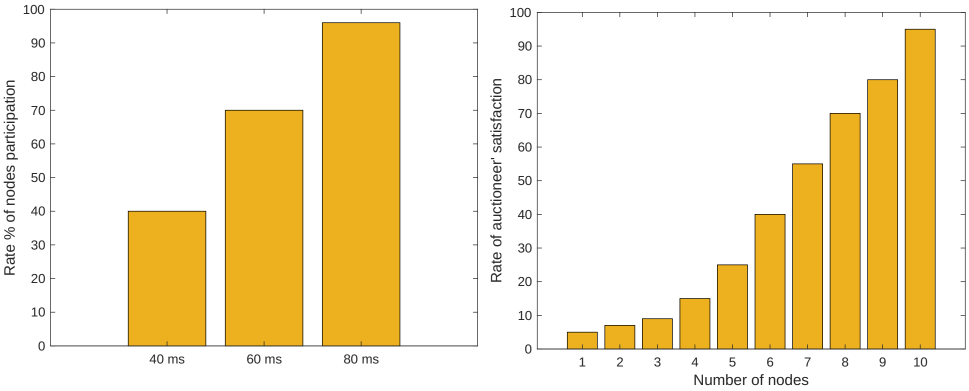
- Cryptographic system parameter. The proposed security and privacy cryptographic scheme is executed on the type A of the JPBC (http://gas.dia.unisa.it/projects/jpbc (accessed on 18 October 2023)) library based on a security parameter, $\Theta = 128$, over the elliptic curve, $y^2 = x^3 + x$, and field, $\mathbb{F}_q$, with the embedding degree, $\kappa = 2$. The operation is executed using Java programming language.
- Blockchain system parameter. A simulation of the proposed scheme is used to determine its actual fairness in the mining process. In the simulation setting, we instantiate 100 virtual node servers. We assume that the 100 nodes randomly send their bidding values in each mining session, including their costs and time periods to the auctioneer. For a real simulation, we use a private blockchain network using the Hyperledger Besu (https://www.hyperledger.org (accessed on 21 October 2023)). This is an open-service platform that provides web client management and monitoring tools based on Java and JavaScript. The parameter settings are built on three servers for the test chain. The first server node is installed to manage the HTTP execution based on next.js (https://nextjs.org/ (accessed on 21 October 2023)). The second node is a truffle server installed to manage the Ethereum. The third node is the nginx server installed to act as an auctioneer server.

7.3. Performance of PoF Protocol

In the simulation, we appreciate that all nodes are arbitrarily connected and managed by the auctioneer. In this experiment, the auctioneer server is in charge of selecting a miner node as a winner by considering the level sensitivity of the block, the time of generation, and its social cost. We set up a time interval $[t_s, t_e]$, determined to be 40 ms, 60 ms, and 80 ms as the required times to generate a new block. Thus, the node that has the computational capability of generating a block within the required time interval can participate in the auction process by sending its bidding value.

As shown in Figure 3a, it can be inferred that the number of bidder nodes is influenced by the time periods. For example, if the time period is 60 ms, the number of nodes increases because nodes with various computational capabilities have a higher chance of being selected as winners.

This will positively affect the auctioneer satisfaction as shown in Figure 3b. This is because the cost of a range of 100 nodes initially decreases as more nodes participate in the auction system. Therefore, the auctioneer will easily find a participant node with a lower social cost that has compatible computational power for achieving the required mining process at the required time. The auctioneer's satisfaction is measured by selecting a node that can perform the task at a lower cost. Moreover, node satisfaction is measured by increasing the chances for all nodes to win the auction session fairly.



(**a**) Influence of the time period.

(**b**) Auctioneer's satisfaction.

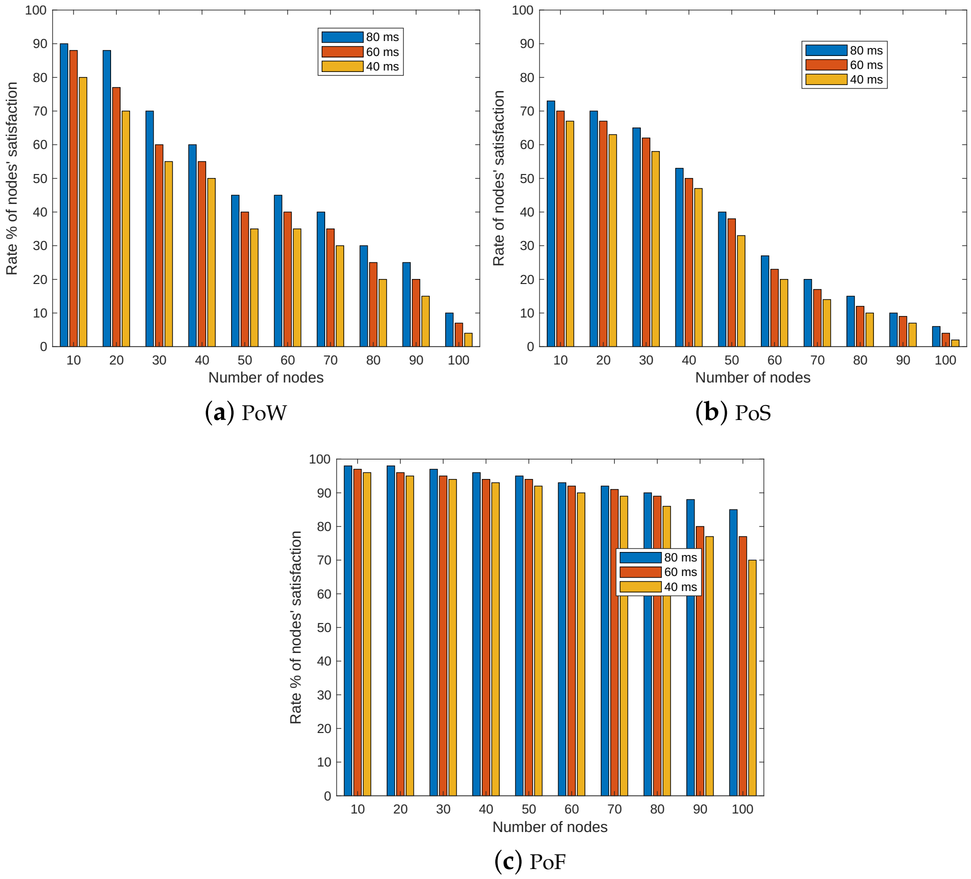
Figure 3. Effectiveness analysis of the proposed PoF protocol.

A Fairness of Selecting Miner Node

The fairness in selecting a miner node is mainly measured by evaluating the participating nodes' satisfaction. Since the PoW and PoS require powerful computational features and resources, most nodes that have limited computational powers feel as if they are not capable of winning the session. Thus, this will lead to node satisfaction issues, which results in stopping nodes from participating in any further mining selection process. Even when assigning various time durations (40 ms, 60 ms, and 80 ms) as the time for generating a block to encourage node participation in the miner selection process, they still feel unsatisfied, as shown in Figure 4a,b.

Compared with the proposed PoF protocol, Figure 4c illustrates that nodes are satisfied even when increasing the number of nodes. This is because the proposed PoF protocol selects a miner node based on two factors: lower costs and low time durations to generate blocks. Therefore, the node that is capable of generating a block within the required time duration, and at a lower cost, has a high chance to win the selection session.

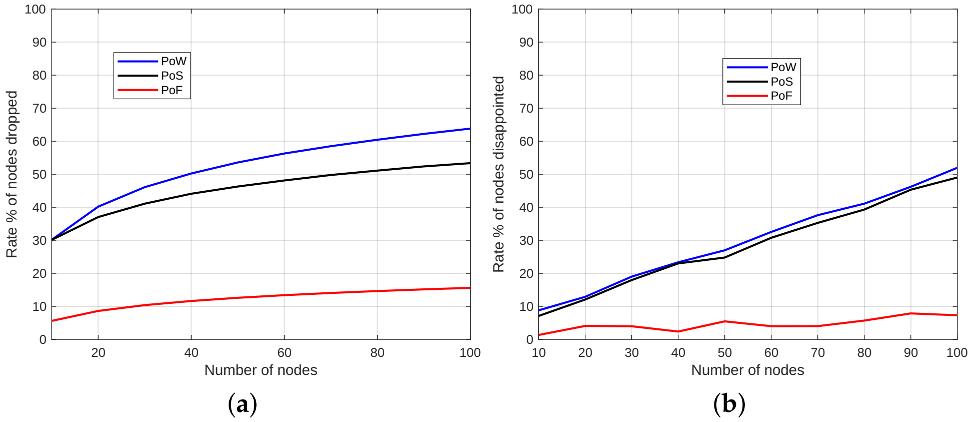
Moreover, in PoW and PoS protocols, the puzzle complexity and amount of money linearly increase with the number of participating nodes, which will increase the transmission time and cost. Then the auctioneer could be incapable of handling all received messages within ε. Thus, as shown in Figure 5a, most of the received messages of bidder nodes in PoW and PoS could be dropped, which allows excluding several nodes from being selected as a miner. Consequently, the fairness and competition among nodes in PoW and PoS protocols are not guaranteed.



(**a**) PoW (**b**) PoS

(**c**) PoF

Figure 4. Rate of participating nodes' satisfaction.



(**a**) (**b**)

Figure 5. Rate of participating nodes dropping out and disappointing. (**a**) Rate of participating nodes dropping out. (**b**) Rate of disappointing participating nodes.

In contrast, the proposed PoF protocol gives all nodes, even those with limited computational capabilities, a sense of having a high chance of being selected as a miner. Our PoF protocol works effectively and dynamically to select a miner process with guaranteed fairness among participating nodes, in comparison with other PoW and PoS protocols. Thus, nodes in the proposed PoF protocol are more eager to participate in each miner selection session, while nodes in PoW and PoS are lazy or disappointed, as shown in Figure 5b.

7.4. Comparisons of Effectiveness

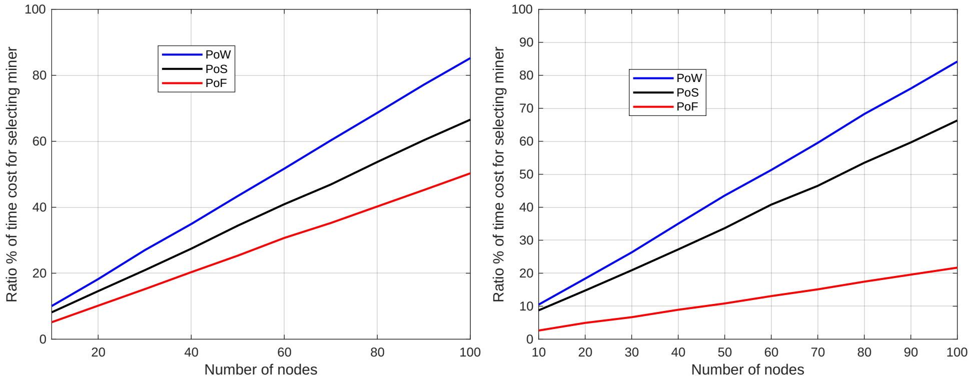
The performance effectiveness of the proposed protocol has been compared with PoW and PoS in terms of the following considerations:

Time Cost of Miner Node Selection

The total time cost in selecting a miner node is majorly affected by two factors: the time of the computational cost and the time of the transmission cost. To produce an accurate evaluation, we set a threshold (ϵ = 1 s) as the maximum time of the session that the auctioneer, nd_{a_i}, can use to select a winning node, nd_i, as a miner from the n bidder nodes. Additionally, we set a threshold (ε = 10,240 bits) as a maximum capability of the auctioneer, nd_{a_i}, to deal with n received messages from n nodes during the ϵ of selecting the miner node.

- Time of computational cost: Comparing PoF to PoW and PoS protocols, we observe that, with increasing the number, n, of bidder nodes, the computational costs linearly increase in PoW and PoS protocols compared to our proposed PoF protocol, as illustrated in Figure 6a. This is because, with increasing n bidder nodes, the complexity of the mathematical puzzle in the PoW protocol will also be increased, which increases the computational cost of solving the puzzle. Additionally, within the PoS protocol, the computational cost for counting and collecting the amount of cryptocurrency will increase by increasing n bidder nodes.
- Time cost of transmission message: As shown in Figure 6b, the proposed PoF protocol has a lower transmission cost compared with PoW and PoS protocols. The transmission cost in the PoW protocol increases linearly with the cost of the mathematical puzzle. As discussed previously, with the increasing number of bidders, the complexity of the puzzle will also increase. Thus, this will also increase the size of the message length, and the transmission time cost will increase. In addition, in the PoS protocol, the transmission cost also increases linearly when increasing the amount of cryptocurrency. When increasing the number of bidders, each node will increase its amount of cryptocurrency to win the session.

Thus, the time cost of selecting miner nodes in PoW and PoS protocols is not practical as the number of connected nodes increases, due to the requirement for high computational power or high credit, along with high bandwidth for transmitting their messages. Therefore, the proposed PoF protocol is more suitable with limited computational and communication overhead, with stable bandwidth and computational costs when selecting a miner node.



(**a**) Time of Computational Cost (**b**) Time Cost of Transmission Message

Figure 6. Rate of time cost when selecting the miner node.

7.5. Performance of a Security and Privacy Scheme of PoF Protocol

Since the node's winning depends on providing its best strategy, the node's data strategy should be protected from being disclosed or forged. However, the designed cryptography must be compatible with the time period of the miner node selection process. This work mainly focuses on the computational and communication overhead of proposed cryptographic operations and omits the computations performed for selecting miners and generating blocks.

7.5.1. Computational Cost

To evaluate the proposed security and privacy scheme and illustrate its performance efficiency, the time consumption of performing the cryptographic computational operations is considered. The scalar multiplication in $\Im_1$ and pairing $\hat{e}$ are the main operations that are considered to evaluate the time consumption of the proposed scheme. Let mG_1 and $\hat{p}$ denote the scalar multiplication in $\Im_1$ and bilinear pairing $\hat{e}$, respectively. The computational cost of the proposed scheme will be evaluated using the following phases:

- Bidding phase (Bid): In this phase, each node, nd_i, encrypts its best strategy, which generates nine multiplication operations, $9(mG_1))$, in $\Im_1$.
- Verification phase (Ver): In this phase, the auctioneer will take three bilinear pairing for n nodes $n[3(\hat{p})]$ to verify their validation.
- Sorting phase $(Sort)$: In this phase, the auctioneer will also take three bilinear pairing $3(\hat{p})$ to sort nodes based on their time and social costs.
- Selecting the winner phase $(WinSel)$: In this phase, the auctioneer needs to compute three multiplication operations, $3(mG_1))$ in $\Im_1$.

Table 1 illustrates the computational costs of the proposed scheme with 50 connection nodes. It shows that the verification process is the most expensive phase, consuming large computational costs. This is because the auctioneer is required to verify the validity for n bidder nodes. However, as shown in Table 1, by evaluating the scheme with 100 connection nodes, the verification process does not largely consume costs, as the auctioneer performs an aggregation algorithm to aggregate all ciphertexts of the nodes, verifying them simultaneously. In contrast, the computational time costs of the other phases are not affected by n.

Table 1. Computational overhead.

	(Bid)	Ver	$Sort$	$WinSel$
Operations	$9\,mG_1$	$3\hat{p}$	$n(3\hat{p})$	$3\,mG_1$
		$n = 50$		
Max Time	16.20 ms	27.85 ms	461.20 ms	5.30 ms
Min Time	13.70 ms	20.80 ms	189.5 ms	3.50 ms
Average Time	14.61 ms	22.70 ms	276.30 ms	4.10 ms
		$n = 100$		
Max Time	16.91 ms	28.10 ms	635.50 ms	5.62 ms
Min Time	13.70 ms	21.66 ms	337 ms	3.50 ms
Average Time	14.75 ms	23.10 ms	454 ms	4.80 ms

7.5.2. Communication Cost

The ciphertext length size is used to measure the cost of the proposed communication propriety. The communication overhead of the proposed protocol is analyzed in terms of considering communication from bidders to the auctioneer. In submitting a bid value, a node, nd_i, is required to send the ciphertext of its bids (E_i, S_i) to the auctioneer, nd_{a_i}, which is $(5|\Im_1|)$. The binary length for every multiplication generated by a scalar point in $\Im_1$ is 160 bits. Therefore, 800 bits is the size length of the ciphertext (E_i, S_i) that is generated for communication from the bidders to the auctioneer.

8. Conclusions and Future Work

This paper presents a new dynamic mining protocol that takes into account the time-sensitivity of transactions and selects the appropriate miner accordingly, rather than solely considering computational and resource capabilities. The paper introduces a reverse auction mechanism as an incentive for all nodes to participate in the miner selection process. Additionally, an expressive language is devised to classify transaction types based on their sensitivity to processing times, making them compatible with our miner selection process. Furthermore, a homomorphic concept is developed as a privacy-preserving scheme to safeguard the confidentiality of bidders' data. Finally, the effectiveness of the PoF is extensively validated through numerical analysis and simulation. The results demonstrate that the proposed mechanism effectively ensures fair miner node selection while reducing the burden of reward costs on the system. In the future, the research will focus on a smooth, decentralized, and low-cost auctioneer selection process, as well as applying and measuring the performance of the PoF on real applications. This will include extensive testing of the performance and efficiency of the proposed security scheme against different kinds of attacks.

Author Contributions: Conceptualization, A.A. and B.A.; Methodology, B.A.; Formal analysis, A.A. and B.A.; Investigation, A.A. and B.A.; Writing—original draft, A.A.; Writing—review & editing, B.A. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the Deputyship of Research Innovation, Ministry of Education, in Saudi Arabia for funding this research work through project number ISP-2024.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Bus. Rev.* **2008**, 21260.
2. Hakak, S.; Khan, W.Z.; Gilkar, G.A.; Assiri, B.; Alazab, M.; Bhattacharya, S.; Reddy, G.T. Recent advances in blockchain technology: A survey on applications and challenges. *Int. J. Ad Hoc Ubiquitous Comput.* **2021**, *38*, 82–100. [CrossRef]
3. Tang, Y.; Xiong, J.; Becerril-Arreola, R.; Iyer, L. Ethics of blockchain: A framework of technology, applications, impacts, and research directions. *Inf. Technol. People* **2020**, *33*, 602–632. [CrossRef]
4. Buterin, V. A next-generation smart contract and decentralized application platform. *White Paper* **2014**, *3*, 1–2.
5. Miao, J.; Wang, Z.; Wu, Z.; Ning, X.; Tiwari, P. A blockchain-enabled privacy-preserving authentication management protocol for Internet of Medical Things. *Expert Syst. Appl.* **2024**, *237*, 121329. [CrossRef]
6. Assiri, B.; Khan, W.Z. Enhanced and lock-free tendermint blockchain protocol. In Proceedings of the 2019 IEEE International Conference on Smart Internet of Things (SmartIoT), Tianjin, China, 9–11 August 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 220–226.
7. Alamer, A.M.A. A secure and privacy blockchain-based data sharing scheme in mobile edge caching system. *Expert Syst. Appl.* **2024**, *237*, 121572. [CrossRef]
8. Zhang, S.; Lee, J.H. Analysis of the main consensus protocols of blockchain. *ICT Express* **2020**, *6*, 93–97. [CrossRef]
9. Assiri, B.; Busch, C. Approximately opaque multi-version permissive transactional memory. In Proceedings of the 2016 45th International Conference on Parallel Processing Workshops (ICPPW), Philadelphia, PA, USA, 16–19 August 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 393–402.
10. Assiri, B.; Busch, C. Approximate count and queue objects in transactional memory. In Proceedings of the 2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Lake Buena Vista, FL, USA, 29 May–2 June 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 894–903.
11. Li, Q.; Jia, X.; Huang, C. A truthful dynamic combinatorial double auction model for cloud resource allocation. *J. Cloud Comput.* **2023**, *12*, 106. [CrossRef]
12. Alamer, A.; Basudan, S. An efficient truthfulness privacy-preserving tendering framework for vehicular fog computing. *Eng. Appl. Artif. Intell.* **2020**, *91*, 103583. [CrossRef]
13. Pawar, V.; Sachdeva, S. ParallelChain: A scalable healthcare framework with low-energy consumption using blockchain. *Int. Trans. Oper. Res.* **2023**.. [CrossRef]
14. Chaudhary, R.; Jindal, A.; Aujla, G.S.; Aggarwal, S.; Kumar, N.; Choo, K.K.R. BEST: Blockchain-based secure energy trading in SDN-enabled intelligent transportation system. *Comput. Secur.* **2019**, *85*, 288–299. [CrossRef]
15. Mingxiao, D.; Xiaofeng, M.; Zhe, Z.; Xiangwei, W.; Qijun, C. A review on consensus algorithm of blockchain. In Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, Canada, 5–8 October 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 2567–2572.

16. Creydt, M.; Fischer, M. Blockchain and more-Algorithm driven food traceability. *Food Control* **2019**, *105*, 45–51. [CrossRef]
17. Gervais, A.; Karame, G.O.; Wüst, K.; Glykantzis, V.; Ritzdorf, H.; Capkun, S. On the security and performance of proof of work blockchains. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 3–16.
18. Gupta, D.; Saia, J.; Young, M. Proof of work without all the work. In Proceedings of the 19th International Conference on Distributed Computing and Networking, Varanasi, India, 4–7 January 2018; pp. 1–10.
19. Barhanpure, A.; Belandor, P.; Das, B. Proof of stack consensus for blockchain networks. In Proceedings of the Security in Computing and Communications: 6th International Symposium, SSCC 2018, Bangalore, India, 19–22 September 2018; Revised Selected Papers 6; Springer: Berlin/Heidelberg, Germany, 2019; pp. 104–116.
20. Gaži, P.; Kiayias, A.; Zindros, D. Proof-of-stake sidechains. In Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 139–156.
21. Saad, S.M.S.; Radzi, R.Z.R.M. Comparative review of the blockchain consensus algorithm between proof of stake (pos) and delegated proof of stake (dpos). *Int. J. Innov. Comput.* **2020**, *10*.
22. Ren, L.; Devadas, S. Proof of space from stacked expanders. In Proceedings of the Theory of Cryptography: 14th International Conference, TCC 2016-B, Beijing, China, 31 October–3 November 2016; Proceedings, Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 262–285.
23. De Angelis, S.; Aniello, L.; Baldoni, R.; Lombardi, F.; Margheri, A.; Sassone, V. PBFT vs. Proof-of-Authority: Applying the CAP Theorem to Permissioned Blockchain. In Proceedings of the 2nd Italian Conference on Cyber Security ITASEC-2018, Milan, Italy, 6–9 February 2018; CEUR-WS: Aachen, Germany, 2018.
24. Schwartz, D.; Youngs, N.; Britto, A. The ripple protocol consensus algorithm. *Ripple Labs Inc. White Pap.* **2014**, *5*, 151.
25. Assiri, B. Using leader election and blockchain in E-health. *Adv. Sci. Technol. Eng. Syst. J.* **2020**, *5*, 46–54. [CrossRef]
26. Reyna, A.; Martín, C.; Chen, J.; Soler, E.; Díaz, M. On blockchain and its integration with IoT. Challenges and opportunities. *Future Gener. Comput. Syst.* **2018**, *88*, 173–190. [CrossRef]
27. Bhaskar, P.; Tiwari, C.K.; Joshi, A. Blockchain in education management: Present and future applications. *Interact. Technol. Smart Educ.* **2021**, *18*, 1–17. [CrossRef]
28. Cho, S.; Lee, S. Survey on the Application of BlockChain to IoT. In Proceedings of the 2019 International Conference on Electronics, Information, and Communication (ICEIC), Auckland, New Zealand, 22–25 January 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–2.
29. Alam, S.; Shuaib, M.; Khan, W.Z.; Garg, S.; Kaddoum, G.; Hossain, M.S.; Zikria, Y.B. Blockchain-based initiatives: Current state and challenges. *Comput. Netw.* **2021**, *198*, 108395. [CrossRef]
30. Khubrani, M.M.; Alam, S. A detailed review of blockchain-based applications for protection against pandemic like COVID-19. *TELKOMNIKA (Telecommun. Comput. Electron. Control)* **2021**, *19*, 1185–1196. [CrossRef]
31. Syed, T.A.; Alzahrani, A.; Jan, S.; Siddiqui, M.S.; Nadeem, A.; Alghamdi, T. A comparative analysis of blockchain architecture and its applications: Problems and recommendations. *IEEE Access* **2019**, *7*, 176838–176869. [CrossRef]
32. Zheng, Z.; Xie, S.; Dai, H.; Chen, X.; Wang, H. An overview of blockchain technology: Architecture, consensus, and future trends. In Proceedings of the 2017 IEEE International Congress on Big Data (BigData Congress), Honolulu, HI, USA, 25–30 June 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 557–564.
33. Zhang, R.; Xue, R.; Liu, L. Security and privacy on blockchain. *ACM COmputing Surv. (CSUR)* **2019**, *52*, 1–34. [CrossRef]
34. Sheneamer, A.; Roy, S.; Kalita, J. An effective semantic code clone detection framework using pairwise feature fusion. *IEEE Access* **2021**, *9*, 84828–84844. [CrossRef]
35. Basudan, S. A Scalable Blockchain Framework for Secure Transactions in IoT-Based Dynamic Applications. *IEEE Open J. Commun. Soc.* **2023**, *4*, 1931–1945. [CrossRef]
36. Basudan, S. A puncturable attribute-based data sharing scheme for the Internet of Medical Robotic Things. *Libr. Hi Tech* **2022**, *40*, 1064–1080. [CrossRef]
37. Assiri, B.; Busch, C. Transactional Memory Scheduling Using Machine Learning Techniques. In Proceedings of the 2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP), Heraklion, Greece, 17–19 February 2016 IEEE: Piscataway, NJ, USA, 2016; pp. 718–725.
38. Alamer, A.; Basudan, S. A Security and Privacy-Preserving Accessing Data Protocol in Vehicular Crowdsensing Using Blockchain. In Proceedings of the Seventh International Congress on Information and Communication Technology: ICICT 2022, London, UK, 21–24 February 2022; Springer: Berlin/Heidelberg, Germany, 2022; Volume 2, pp. 315–327.
39. Alamer, A.M.A.; Basudan, S.A.M.; Hung, P.C. A privacy-preserving scheme to support the detection of multiple similar request-real-time services in IoT application systems. *Expert Syst. Appl.* **2023**, *214*, 119005. [CrossRef]
40. Boneh, D.; Franklin, M. Identity-based encryption from the Weil pairing. *SIAM J. Comput.* **2003**, *32*, 586–615. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.