IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 29 November 2022, accepted 1 January 2023, date of publication 13 January 2023, date of current version 8 September 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3236891*

RESEARCH ARTICLE

# Incentive Mechanism Design for Mitigating Frontrunning and Transaction Reordering in Decentralized Exchanges

**DANIEL MAWUNYO DOE[1], JING LI[1], NIYATO DUSIT[2], (Fellow, IEEE), LI WANG[3], AND ZHU HAN[1]**

[1]Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77004, USA
[2]Computer Science and Engineering, Nanyang Technological University, Singapore 639798
[3]School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding author: Daniel Mawunyo Doe (dmdoe@uh.edu)

**ABSTRACT** Decentralized exchanges (DEXes) provide effective price discovery and fair trading while dealing with the drawbacks of centralized exchanges, e.g., lack of transaction transparency and exclusive control of user assets and transaction fees. However, many DEXes suffer from frontrunning and transaction reordering, which fundamentally flaw their design. In this paper, we present a novel incentive mechanism design for mitigating frontrunning and transaction reordering even if frontrunners pay high transaction fees in DEXes. We utilize a weighted counting sort algorithm to order transactions based on the users' multi-dimensional private information (e.g., transaction delay and confidentiality). To elicit users' private information, we consider a multi-dimensional contract-theoretic design based on the users' willingness to share their private information. We show that the miner can always maximize its utility under the complete and incomplete information scenarios. We implement solutions to our multi-dimensional contract and sorting algorithm on a decentralized oracle network to create a decentralized system and design a web application to extensively evaluate the performance of our proposed incentive mechanism. We further show that ordering transactions based on users' private information increases the miner's utility by 78.42% − 84.57% and reduces the users' cost by 64.47% compared with the state-of-the-art fair sequencing services, automated arbitrage market maker, and miner extractable value auctions.

**INDEX TERMS** Blockchain, decentralized exchanges, incentive mechanism, multi-dimensional contract, transaction ordering.

## I. INTRODUCTION

### A. BACKGROUND AND MOTIVATIONS

Blockchains have gained significant traction in diverse areas over the few years. Many users have rushed to build innovative solutions on blockchains, such as bitcoin and ethereum. In particular, blockchains feature the key benefits of transaction decentralization, transparency, tokenization, and anonymity for a more reliable exchange of value among participants [1]. Blockchains also provide the essential platform for decentralized exchanges (DEXes), which enable users to buy and sell cryptocurrencies without the need for brokers [2]. While these are the marks of blockchain's tremendous success, certain behaviors of miners, such as reordering transactions to maximize their block utilization and profits, lead to higher transaction costs and longer mining delay for users. Also, transactions become more vulnerable to frontrunning as they languish in the mempool.

Furthermore, Daian et al. neologized the term miner extractable value (MEV) to express the various possibilities of using adversarial ordering optimization (AOO) to extract money from a blockchain smart contract system [3]. Recent investigations have determined that at least 28.8M USD was taken from MEV over the past decade [4]. Usually, the term MEV is closely associated with miners because they hold unlimited power to order transactions in a block. At the

The associate editor coordinating the review of this manuscript and approving it for publication was Abderrezak Rachedi.

application layer, MEVs like sandwich trading worsen the user experience[1] [5]. Sandwiched users have more slippage and poorer trade execution, e.g., increased latency [3]. Likewise, generalized frontrunners and the gas-price auctions that typically engage in (when two or more frontrunners raise their gas prices to compete for the next block) cause network congestion and high gas costs at the network layer [6]. MEV can also have deleterious effects between blocks. If a block's MEV exceeds the regular block reward, miners may be encouraged to remine and acquire the MEV, which triggers network reorganization and consensus instability [3].

There have been some efforts to tackle the above MEV extraction challenges. We classify these efforts into transaction-ordering-based and fees-optimization-based approaches. Firstly, in the transaction-ordering-based approach, the authors in [7] introduced fair sequencing service (FSS), a decentralized oracle network (DON)[2] for fair user transaction sequencing based on the time of arrival, which improves the users' transaction latency. However, this transaction-ordering-based technique typically leads to poor income for miners and redundant MEV on the network (e.g., [7]). Secondly, the works in [8] and [9] presented an auction theoretic approach to mitigate MEV via auctioning the right to order transactions on the blockchain network. Nonetheless, these approaches result in ''managed centralization,'' where a single sophisticated party always wins the auction and captures all of the MEV (e.g., [8], [10], [11]). Fees-optimization-based techniques, such as [12] and [13], regulated transaction fees to lower user expenses. In [12] and [14], the authors presented an automated market maker (AMM) to reduce users' costs by leveraging smart contracts (SCs) to alter gas fees upon incoming transaction requests autonomously. The authors of [13] introduced an automated arbitrage market maker (A2MM) that selects the best-effort two-point arbitrage among various AMMs based on transaction costs. Generally, in fees-optimization-based systems, fees cannot be forecasted with 100% accuracy before submitting an order since requests may be executed on bridge reserves with higher fees without the user's knowledge [14].

However, with all the promising efforts to mitigate MEV, certain challenges still need to be addressed. Most existing studies usually make an optimistic assumption that users are often willing to pay an equivalent amount for mining, corresponding to their profit margins. This assumption may not be realistic due to the network's various user types, e.g., users with diverse preferences for confidentiality and delay tolerance. One way is to ensure that users pay for transactions honestly to promote ethical miner behavior. However, this may not be easy to achieve if the miner does not know users' multi-dimensional information preferences, such as transaction delay and confidentiality. Users' transaction delay preferences are characterized by how long they can wait in a queue until their transactions are mined, which is often unknown to the miner, especially when there are several heterogeneous users. The confidentiality of users' transactions is also users' private information and will not be easily accessed by the miner due to privacy concerns. Although the miner may not know each user's private information, it may obtain certain statistics of such information from market research and transaction histories [15]. For instance, the miner may know the user type distributions (which is referred to as an incomplete information in this paper). Different levels of information asymmetry require the miner to design different optimal strategies to achieve the highest possible utility. Therefore, motivated by the preceding discussion, we ask and attempt to answer the following questions in this paper:

1) *How could transactions be ordered on the blockchain network to avoid MEV centralization and extraction?*
2) *How should users with multi-dimensional private information be incentivized to encourage honest behaviors?*
3) *How could transaction ordering and multi-dimensional contracts be implemented on the blockchain platform?*

### B. CONTRIBUTIONS

Contract theory is considered to be an effective approach for designing incentive mechanisms under asymmetric information scenarios [16]. Therefore, we leverage contract theoretic techniques to design an incentive mechanism with the users' multi-dimensional private information to encourage ethical behaviors in the blockchain network. Firstly, we characterize the user types based on their multi-dimensional private information (transaction delay and confidentiality), which can help the miner make effective decisions to maximize its utility. Secondly, we provide a contract design under the complete information scenario and incomplete information scenario to users based on the willingness to share their private information with the miner.[3] This contract design extracts the users' private information necessary for an efficient transaction ordering stage. Next, we deploy a weighted counting sort algorithm to order transactions on the blockchain network based on the users' private information. Expressly, our multi-dimensional contracting and transaction sorting algorithm are jointly achieved via a DON to prevent MEV centralization and extraction. The DON forwards transactions to a mempool after ingesting them and reaching a consensus on the transaction order based on users' private information. We term this integrated system as weighted sequencing service (WSS) in the subsequent literature for simplicity.

To the best of our knowledge, this work is the first attempt to investigate the relationships between miners and users with multi-dimensional private information and related transaction ordering problems resulting from MEV. Compared to

[1] Sandwiching involves placing orders before and after a transaction. The attacker front-runs and back-runs concurrently, sandwiching the original pending transaction.

[2] A DON is a collection of autonomous blockchain oracles that supply data to a blockchain and eliminates single points of failure in smart contracts.

[3] In contract theory, incompleteness occurs from the fact that the information is expensive and sometimes inaccessible to (a) the parties at the time of contracting or (b) the parties or the enforcing court at the time of enforcement [17]. We use the complete information scenario as our preliminary benchmark, where the user information is readily available.

previous recent works, our contributions can be summarized as follows:

- *Transaction ordering mechanism design:* We apply a weighted counting sort algorithm with very low computational complexity to order transactions based on user preferences from our multi-dimensional contract.
- *Multi-dimensional contract design with users' private information:* We design a multi-dimensional contract that elicits users' private information and overcomes the information asymmetry between the miner and users.
- *Implementation of multi-dimensional contract and transaction ordering algorithm:* We deploy the solutions to our optimal contract design and transaction ordering algorithm on a DON to demonstrate the feasibility of the real deployment of our proposed scheme.
- *Experiment and results discussion:* We develop a web application platform to experiment and evaluate the performance of our proposed mechanism gains compared with existing approaches.

The rest of the paper is organized as follows. Section III presents the system model and the preliminaries for our incentive mechanism design. We present our multi-dimensional contract formulations and closed-form solutions in Section IV. Section V introduces the transaction sorting algorithm for our proposed mechanism and overall system architecture. In Section VI, we illustrate the experiment results and analysis of our proposed mechanism, and finally, Section VII concludes our discussion.

## II. RELATED WORKS

Blockchains have gained significant traction in diverse areas over the past few years. As such, several kinds of research have been centered on incentive mechanism designs in the blockchain aspect of this field of study. In [18], the authors proposed a data-sharing incentive model based on an evolutionary game theory using blockchain with a smart contract. The smart contract mechanism can control the excitation parameters in real time and encourage users to share data. The authors in [19] addressed a need for more understanding of the strategic behavior of rational processors within committees in shard-based consensus protocols by analyzing the behavior of processors using a game-theoretic model. In [20], the authors presented a reputation system called RTChain, which is integrated into the e-commerce blockchain to achieve a distributed consensus and transaction incentives.

Furthermore, in close relation to our work, the authors in [7] introduced fair sequencing service (FSS), a decentralized oracle network for fair user transaction sequencing based on the time of arrival, which improves the users' transaction latency. However, this transaction-ordering-based technique typically leads to poor income for miners and redundant MEV on the network (e.g., [7]). Secondly, the works in [8] and [9] presented an auction theoretic approach to mitigate MEV via auctioning the right to order transactions on the blockchain network. Nonetheless, these approaches result in "managed centralization," where a single sophisticated party always wins the auction and captures all of the MEV (e.g., [8], [10], [11]). In [12] and [13], regulated transaction fees to lower user expenses. In [12] and [14], the authors presented an automated market maker (AMM) to reduce users' costs by leveraging smart contracts (SCs) to alter gas fees upon incoming transaction requests autonomously. The authors of [13] introduced an automated arbitrage market maker (A2MM) that selects the best-effort two-point arbitrage among various AMMs based on transaction costs. Generally, in fees-optimization-based systems, fees cannot be forecasted with 100% accuracy before submitting an order since requests may be executed on bridge reserves with higher fees without the user's knowledge [14].

Our work differs from [7], [8], [9], [12], [13], and [14] in that we consider a multi-dimensional contract-theoretic design based on the users' willingness to share their private information, which is used for sorting transactions to optimize the users' and miners' utilities. In principle, our proposed mechanism is a more practical approach to transaction reordering and frontrunning on blockchain networks. The authors in [21] proposed a consensus scheme called PoRF, which is based on a reputation-based consensus that allows for fair and random transaction selection. It aims to address the problem of dishonest nodes colluding with other nodes to prioritize their transactions over others, reducing their latency and improving their QoS. However, the authors fail to consider the existential users' and miners' information asymmetry levels.

## III. SYSTEM MODEL

Consider a blockchain network with DON, users, and miners interacting via DEX contracts. We propose a contract-based incentive system to characterize the miners' and users' honest conduct under various information asymmetries. Users can choose from a range of contract items offered by the DON on behalf of the miners. Section III-A presents a modeling of the interaction between the DON, miners, and blockchain users. As our network model, we present the user types and contract preliminaries in Section III-B. In Section III-C, we provide the utility model, which specifies the users' payoffs and the miners' utility.

### A. SYSTEM ARCHITECTURE

Consider an ethereum proof-of-work (PoW) network, which is composed of immutable ledgers protected by a decentralized network of computers, known as "miners" [22]. We can also apply the concepts from this work to other blockchain networks, such as proof of stake, as long as they support DEXes. These miners are responsible for cumulating pending transactions into blocks, which are subsequently validated by the entire network and added to a global ledger. New blocks of transactions are continuously generated, while the blockchain network ensures that all transactions are

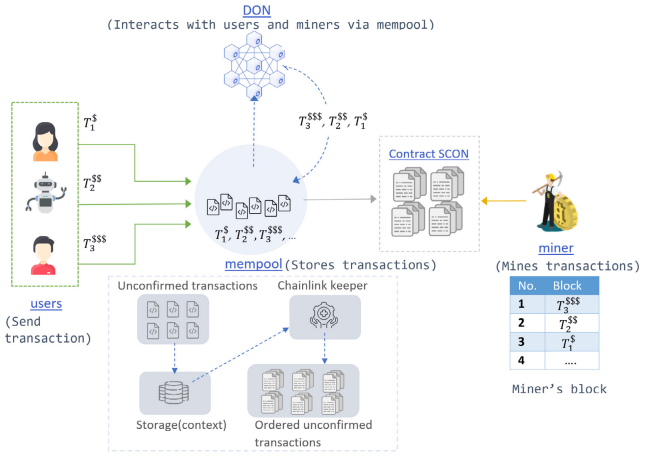


**FIGURE 1.** An overview of DON, miners, and users' interactions in our system architecture.

valid, e.g., no double-spending attacks.[4] Consequently, the blockchain network has a limited network downtime, which makes it challenging to guarantee that transactions will be ordered exactly as submitted on the blockchain [23]. Each block in the blockchain can only hold a certain number of transactions. Conventionally, miners have complete control over which pending transactions from the mempool they include in their blocks (an off-chain location for unconfirmed transactions). However, this autonomy brings with it the inherent issues associated with MEV.

In this work, we deploy the DON, which comprises a committee of blockchain responsible for handling any interaction between the miners and users. By severing the ability to order transactions from the ability to produce blocks with our DON, we can avoid malicious value extraction by miners or transaction frontrunning by users through predefined ordering policies. The DON provides real-world data to a blockchain through middleware and is also capable of simple computation on such data. To achieve our DON implementation, we adopt a chainlink platform that gives blockchain developers an easy-to-use framework for writing hybrid SCs that connect to external resources by combining on-chain and off-chain computation [24]. We refer to the implementation from [25] and [26] to construct our DON. For simplicity, we show a list of major symbols and their definitions in Table 1.

Fig. 1 illustrates the typical DON, miner and users interaction on a blockchain network.[5] Users submit transactions to the mempool, and miners decide from the mempool on which transactions to include in their block of transactions and the transaction order. In Fig. 1, the variable $T_1^{\$}$ represents a user transaction that arrives first with a gas fee \$, where

[4]Other methods can be employed in blockchains to prevent double-spending attacks. However, this problem is not the focus of this paper.

[5]For simplicity, we consider multiple users and a miner in Fig. 1. However, all deductions extend to multiple miners. See Section VI-A for more on this extension.

**TABLE 1.** List of major symbols and their definitions.

| Symbols | Definition |
|---|---|
| $N$ | Total number of users |
| $\mathcal{I}$ | Set of user types |
| $\mathcal{Q}$ | Set of transactions |
| $\mathcal{Q}^*$ | Set of solution for optimal transaction order |
| $\mu_i$ | User $i$ with multi-dimension data $(\theta_i, t_i)$ |
| $\theta_i$ | User $i$'s transaction confidentiality |
| $t_i$ | User $i$'s delay tolerance |
| $t_{max}$ | Maximum acceptable delay tolerance |
| $s_i$ | User $i$'s transaction block size |
| $\omega_i = \omega(s_i)$ | User $i$'s transaction workload |
| $r_i = r(s_i)$ | Reward for user $i$'s transaction |
| $\boldsymbol{\varphi} = \{\varphi_i\}_{i\in\mathcal{I}}$ | Contract items comprises $(\omega_i, r_i)$ |
| $\gamma$ | Miner's hash power |
| $\beta$ | Fixed bonus for mining new block |
| $\lambda$ | Predefined transaction fee rate |
| $\mathbb{P}(...)$ | Probability of (...) |
| $V(\theta_i, t_i)$ | Evaluation of user $i$'s transaction |
| $U_{\varphi_i}$ | Miner's utility for contract items $\varphi_i$ |
| $W_{\varphi_i}$ | User's utility for contract items $\varphi_i$ |
| $c_i$ | Miner's evaluation of reward cost coefficient |
| $\mathcal{P}_{...}$ | Priority level of transactions |
| $\sigma_i$ | Transaction weights |

$ < $$ < $$$. This concept also holds for the subsequent transactions $T_2^{\$\$}$, and $T_3^{\$\$\$}$. Generally, users send these transactions to the mempool for mining. With chainlink keeper functions, the DON orders unconfirmed transactions from mempool storage to the contract scheme contracted-out number (SCON) for the miner [27]. The DON chainlink keeper functions provide a mechanism for executing basic blockchain network tasks [26]. Typically, a miner will select transactions solely based on the highest gas fees (the transaction fee), e.g., $T_3^{\$\$\$}$, $T_2^{\$\$}$, and $T_1^{\$}$. As a result, a miner can extract extra profits from users by capitalizing on its ability to arbitrarily reorder transactions, creating what is generally referred to as MEV [4], [28]. The subsequent subsections provide the significant preliminaries required for our mechanism design.

### B. NETWORK MODEL

#### 1) USER'S TYPES

We consider a group of $N$ users on the blockchain network, which are characterized by two-dimensional private information: the transaction confidentiality $\theta$ and the delay tolerance $t$. Different applications, e.g., stock trading, financial transactions, NFT, etc., require different confidentiality and delay tolerance levels. In this work, $\theta$ represents private information such as the transaction revenue or arbitrage opportunity observed by the user. For the convenience of presentation, we denote a user with multi-dimensional data $\mu_i$ as $\mu_i \triangleq (\theta_i, t_i)$ as a type-$i$ user for all users belonging to a set $\mathcal{I} = \{1, \cdots, I\}$ of $I$ types,[6] where $\theta_i$ and $t_i$ represent the type-$i$ user's confidentiality and delay tolerance, respectively. Each user type $i \in \mathcal{I}$ comprises $n_i$ users, such that $\sum_{i\in\mathcal{I}} n_i = N$. Practically, users evolve with time and, therefore, will have

[6]We refer to multi-dimensional contracts by P. Bolton for more on user types descriptions [16].

different individual preferences. However, we assume the user's type does not change for simplicity.

Due to the existence of diverse user private information, it is challenging for a miner to predict user behaviors without the information about each user's type.[7] To address this information asymmetry challenge, we propose a contract-theoretic approach that elicits private information from users.

#### 2) CONTRACT FORMULATION

Contract theory is a promising and extensively used theoretic tool for addressing problems with private information. Hence, we introduce our contract formulation in this subsection.

#### 3) MINER'S CONTRACT

The miner will offer a contract that stipulates the relationship among users' delay tolerance, transaction workload and cost. Concretely, contract $\mathcal{C} = (t_{max}, \boldsymbol{\varphi})$ includes a maximum delay $t_{max}$ (for all user types) and $I$ contract items $\boldsymbol{\varphi}$ for each user type, such that $\boldsymbol{\varphi} = \{\varphi_i\}_{i\in\mathcal{I}}$. Notation $t_{max}$ represents the maximum delay acceptable by users on the blockchain network, which can be achieved by setting verifiable delay functions (VDFs) on transactions [29], i.e., miners with $t_i \leq t_{max}$ can finish mining and propagating transactions in time. We specify the relationship between each type-$i$ user's transaction workload and cost in each contract item $\varphi_i \triangleq (\omega_i, r_i)$, where $\omega_i$ and $r_i$ represent the transaction workload and the reward (e.g., money) for mining each type-$i$ user's transaction, respectively, if the miner completes a transaction workload within the required time. We mention that the values of $\omega_i$ and $r_i$ are functions of the mining resource demands, e.g., block size of user $i$'s transaction denoted as $s_i$, such that $\omega_i = \omega(s_i)$ and $r_i = r(s_i)$ [30]. The miner with $t_i > t_{max}$ proposes a zero contract item for any type-$i$.

#### 4) USERS' CHOICES

Before establishing the contract, each user decides on a delay tolerance level and an amount that it is willing to pay for its transaction, based on the transaction's confidentiality, e.g., how much revenue it can obtain from that transaction. Next, the user proceeds to choose a contract item that best describes its type or maximizes its payoff. If a user chooses the contract item $\varphi_i$, the miner needs to ensure that $\omega_(s_i)$ is mined and propagated within time $t_{max}$.[8] In return, the miner receives a reward $r(s_i)$ that determines the user's transaction cost. Usually, the miner and users will not participate if their respective payoffs are negative (defined in Section III-C1 and Section 4d, respectively). Consequently, we specify the miner's and the users' payoffs in the following.

[7]The miner can know each user's type from their multi-dimensional private information provided in the software layer when sending transactions to the blockchain for mining in Section V.

[8]Any reasonable interval can be $t_{max}$. However, we average network execution time for different blocks.

### C. UTILITY MODEL

#### 1) MINER'S UTILITY

The miner's utility is defined as the difference between its reward obtained from the blockchain transaction and transaction mining cost. Consider a miner with an available resource $x$, a transaction workload $\omega$, and the miner's hash power $\gamma$ can be expressed as $\gamma(\omega(s_i), x)$ [31]. We denote the miner's block size by $s$, which comprises the total transactional and metadata size. The token reward for miners constitutes a fixed bonus $\beta \geq 0$ for mining a new block and a variable transaction fee $\lambda s_i$ defined by the occupied transaction block size $s_i$ of a user $i$ and a predefined transaction fee rate $\lambda$ [32]. Therefore, the $i$-th miner's token reward $r(s_i)$ can be expressed as follows:

$$r(s_i) = (\beta + \lambda s_i)\mathbb{P}(\gamma(\omega(s_i), x), s_i), \quad [9] \tag{1}$$

where $\mathbb{P}(\gamma(\omega(s_i), x), s_i)$ represents the probability that the miner obtains the reward for contributing a block to the blockchain [31]. The miner's utility $U$ can be calculated as

$$U_{\varphi_i} = \mathbb{1}_{t_i \leq t_{max}} r(s_i) - \theta_i \omega(s_i), \tag{2}$$

where $\theta_i\omega(s_i)$ is the miner's cost for type-$i$ user's transaction and $\mathbb{1}_{t_i\leq t_{max}}$ can be expressed as

$$\mathbb{1}_{t_i \leq t_{max}} = \begin{cases} 1, & \text{if } t_i \leq t_{max}, \\ 0, & \text{if } t_i > t_{max}, \end{cases} \tag{3}$$

which means that only miners with $t_i \leq t_{max}$ are likely to take a transaction. A miner with $t_i > t_{max}$ gets a deduction to discourage the possibility of MEV extraction once contracting.

#### 2) USERS' PAYOFF

Each user's payoff in each trading is based on its utility and payment. Specifically, a type-$i$ user's utility can be expressed in terms of its confidentiality and valuation for transaction workload, which are usually its private information. The user payment generally comprises the cost for mining its transaction, depending on how much reward the miner obtains from the mining. If a type-$i$ user selects a contract item $\varphi_i$, the respective payoff $W$ can then be calculated as:

$$W_{\varphi_i} = V(\theta_i, t_i) - c_i r(s_i), \tag{4}$$

where $V(\theta_i, t_i)$[10] is the type-$i$ user's evaluation function regarding $\theta_i$ and $t_i$, which is a strictly increasing concave function of $V(\theta_i, t_i)$, where $V(\theta_i, t_i) = 0$, $V'(\theta_i, t_i) > 0$, and $V''(\theta_i, t_i) < 0$ for all $i \in \mathcal{I}$. The term $c_i r(s_i)$ represents the transaction cost from the miner, where $c_i$ is an additional fee imposed by the miner based on its valuation on the reward from providing the required service to users. However, for simplicity, we set $c_i = 1$ to fundamentally not change the nature of our solution.

[10]We consider this function to be a set of weights applied to both $\theta_i$ and $t_i$ of any transaction to determine its value, as discussed in Section V.

## IV. MULTI-DIMENSIONAL CONTRACT DESIGN FOR USERS

In this section, we analyze the miner's optimal incentive mechanism for users with different information asymmetry levels. To investigate the impact of these information asymmetry levels between the miner and users, we consider the following contract designs:

1) *Complete information scenario*: The miner is aware of each user's type, which provides an upper bound of its reward compared with the incomplete information scenario [16].
2) *Incomplete information scenario*: In this case, the miner does not know the user type but only knows the distribution of user types. This scenario leads to the second-best outcome with a non-linear price discrimination for user transactions [16].

For each design, we first present the contract feasibility condition and then solve for the optimal contract. The contract feasibility and optimality are defined as follows [16]:

*Definition 1 (Contract Feasibility): A contract is feasible if each user obtains the maximal payoff by choosing the contract item designed for its type.*

*Definition 2 (Contract Optimality): A contract is optimal if it maximizes the miner's payoff among all feasible contracts.*

The rest of this section includes the optimal contract design under the complete information and incomplete information scenario in Section IV-A and IV-B, respectively.

### A. COMPLETE INFORMATION SCENARIO

In this subsection, we investigate the miner's optimal contract under the complete information scenario where the miner knows each user type. This design makes it feasible for the miner to observe and ensure that each user type accepts the contract items designed for that type and will not accept any contract item not designed for it. Also, the miner can obtain a first-best outcome, which leads to perfect price discrimination [16]. Nonetheless, the miner is still required to guarantee that each user obtains a non-negative payoff, which incentivizes users to accept their corresponding contract item. Expressly, a contract is feasible if and only if it satisfies Individual Rationality (IR) constraints:

*Definition 3 (Individual Rationality): A contract is individually rational if it provides a non-negative payoff to each type-i that accepts the contract item $\varphi_i$ designed for its type, i.e.,*

$$W_{\varphi_i} \geq 0, \quad \forall i \in \mathcal{I}. \tag{5}$$

Therefore, the optimal contract $\mathcal{C}^*_{complete} = (t^*_{max}, \varphi^*)$ under the complete information scenario is the solution to the optimization problem:

*Problem 1 (Contracting Under Complete Information Scenario Problem):*

$$\max_{t_{max}, \varphi} \; U_{\varphi_i} \tag{6a}$$

$$\text{s.t. } W_{\varphi_i} \geq 0, \quad \forall i \in \mathcal{I}, \tag{6b}$$

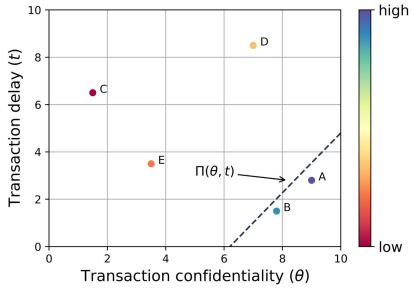


**FIGURE 2.** Miner's preference order for different user types based on $\Pi(\theta, t)$.

which aims to maximize the miner's utility as shown in (6a) under the IR constraint presented in (6b). To solve this problem, we first calculate the miner's reward $r(s_i)$ for any given $s_i$ from (1). Next, we substitute $r(s_i)$ into the miner's objective function to obtain the optimal transaction workload $\omega^*(s_i)$ as well as the optimal transaction delay time $t^*_{max}$ (see Theorem 1).

*Lemma 1: For any given transaction workload $\omega(s_i)$ even if $\omega(s_i) \neq \omega^*(s_i)$, it is optimal for the miner to choose its reward as $r(s_i) = \theta_i \omega(s_i), \forall i \in \mathcal{I}$.*

We provide the proof of Lemma 1 in Appendix, which establishes that the miner will offer zero payoffs to all users contracting under the complete information scenario. Subsequently, we can determine the optimal transaction workload for each user type that minimizes the miner's cost based on Lemma 1. Next, we present another observation based on Lemma 2, showing the impact of choosing each user type-$i$ on the miner's cost.

*Lemma 2: The miner's payoff for only choosing type-i is expressed as*

$$\Pi(\theta_i, t_i) \triangleq \frac{\theta_i^3}{4\mathbb{P}(\gamma(\omega(s_i), x), s_i)} + \frac{\theta_i^2 \left(1 + \frac{\theta_i^2}{4\lambda}\right)}{4\lambda \mathbb{P}(\gamma(\omega(s_i), x), s_i)}. \tag{7}$$

We provide the of Lemma 2 in Appendix. Lemma 2 describes the miner's trade-off towards different users' private information, i.e., $\theta$ and $t$. As a result, we can transform users' two-dimensional private information into a one-dimensional model, which shows the miner's preference for different user types:

*Definition 4 (Miner's Preference List): The miner has a higher preference for a user type-i than type-j (expressed by $i \succ j$) if and only if $\Pi(\theta_i, t_i) > \Pi(\theta_j, t_j)$.*

Fig. 2 shows how the miner's preference changes for each user type over $(\theta, t)$ parameter spectrum. Each axis represents the users' sensitivity towards $(\theta, t)$. Expressly, the miner's preference on users' types is higher for users with high confidentiality and low transaction delay than low confidentiality and high delay users. We express the set of user types that have the same high miner preference as follows:

$$\mu_i^* \triangleq \underset{i \in \mathcal{I}}{\text{argmax}} \; \Pi(\theta_i, t_i). \tag{8}$$

For instance, suppose that there are five user types $A, B, C, D$, and $E$ characterized with $(\theta, t)$ as shown in Fig. 2. The miner's preference list is $A \succ B \succ D \succ E \succ C$ and $\mu_i^* = \{A\}$. In Theorem 1, we provide the optimal contract for the miner under the complete information scenario considering different cases of the set $\mu_i^*$:

*Theorem 1: Under complete information scenario, the miner's preference order for users' transactions can be derived as*:

1) *if* $\mu_i^* = \{i\}$, *the miner's optimal contract is* $t_{max}^* = t_i, \varphi_i^* = [\frac{\theta_i^2}{4\lambda\mathbb{P}(\gamma(\omega(s_i),x),s_i)}, \frac{\theta_i^3}{4\lambda\mathbb{P}(\gamma(\omega(s_i),x),s_i)}]$, *and* $\varphi_i^* = \mathbf{0}, \forall i \neq j$,
2) *if* $|\mu_i^*| > 1$, *the miner's optimal contract is to choose any one type user* $i \in \mu_i^*$ *with* $t_{max}^* = t_i, \varphi_i^* = [\frac{\theta_i^2}{4\lambda\mathbb{P}(\gamma(\omega(s_i),x),s_i)}, \frac{\theta_i^3}{4\lambda\mathbb{P}(\gamma(\omega(s_i),x),s_i)}]$, *and* $\varphi_i^* = \mathbf{0}, \forall i \neq j$,
3) *if* $|\mu_i^*| > 1$, *offering only positive contract to one type* $i \in \mu_i^*$ *results in the same miner's maximal utility.*

Proof of Theorem 1 is provided in Appendix. Theorem 1 illustrates that the miner only provides a positive contract item to the most desired user type and offers zero contract to all other users. Additionally, due to the non-uniqueness of the most preferred user type, the optimal contact may not be unique but will always exist. Conversely, it is not optimal to choose (offer a positive contract item) to several user types within or outside the set $\mu_i^*$, as this would reduce the miner's reward.

Logically, having less user information would result in different behavior of the miners. As a result, we present in the subsequent subsection that the miner's optimal contract under an incomplete information scenario is more complicated than the complete information scenario.

### B. INCOMPLETE INFORMATION SCENARIO

In this subsection, we present the miner's optimal contract under the incomplete information scenario, where it does not know which type each user belongs to but is aware of the probabilistic distribution of users' types, i.e., a probability $\mathbb{P}_i$ that a user belongs to a type-$i$. The user type-$i$ probability $\mathbb{P}_i$ for this case can be expressed as

$$\mathbb{P}_i = \frac{N!}{n_i!(N-n_i)!}\mathbb{P}^{n_i}(1-\mathbb{P})^{N-n_i}, \quad n_i \in N. \tag{9}$$

Moreover, this user type probability in our contract design applies to any distribution that best characterizes the user behavior.

In this design, the miner needs to account for any expected reduction in its payoff for incorrectly proposing contract items not precise for a specific user. Hence, the miner's utility for user type-$i$ in this case is

$$\mathbb{P}_i(U_{\varphi_i}) = \mathbb{1}_{t_i \leq t_{max}}\mathbb{P}_i\{r(s_i) - \theta_i\omega(s_i)\}. \tag{10}$$

Suppose the miner adopts the previously derived optimal contract design from the complete information scenario for the incomplete information scenario. In that case, the miner will have a higher payoff for offering a higher contract item to user type-$i$ instead of user type-$j$, where $V(\theta_j, t_j) > V(\theta_i, t_i)$, and vice versa. The miner will likely receive a low utility when user type-$j$ and $N$ are not sufficiently large. Usually, this type of contract design leads to a second-best outcome, where the miner needs to adopt a non-linear pricing model [16]. Motivated by the structure of the complete information scenario, where miners only select the most preferred user type, we construct a simplified contract where the miner only offers two types of contract items; one is positive payoff for a group $\chi \subseteq \mathcal{I}$ of user types, and the other is zero payoff for the rest user types in $\mathcal{I}\backslash\chi$. This contract structure provides a tractable approach to characterize the incomplete information scenario efficiently.

Also, the miner cannot force a user to accept a specific contract item, but it can design the contract to ensure each user accepts their corresponding contract item. Hence, the miner needs to further guarantee the Incentive Compatibility (IC) constraints:

*Definition 5 (Incentive Compatibility): A contract is incentive compatible if it provides the maximal payoff for each type-i user when he chooses the contract item* $\varphi_i$ *designed for its type, i.e.,*

$$W_{\varphi_i} \geq W_{\varphi_j}, \quad \forall i, j \in \mathcal{I}. \tag{11}$$

We can find the optimal contract $\mathcal{C}_{incomplete}^* = (t_{max}^*, \varphi^*)$, which is a solution to the optimization problem:

*Problem 2 (Contracting Under Incomplete Information Scenario Problem):*

$$\max_{t_{max},\varphi} \ \mathbb{E}\{U_{\varphi_i}\} = \sum_{i=1}^{N}\mathbb{P}_i(U_{\varphi_i}) \tag{12a}$$

$$\text{s.t. } W_{\varphi_i} \geq 0, \quad \forall i \in \mathcal{I}, \tag{12b}$$

$$W_{\varphi_i} \geq W_{\varphi_j},$$

$$\varphi_i > \mathbf{0}, \varphi_j > \mathbf{0}, \quad \forall i, j \in \chi;$$

$$\varphi_k = \mathbf{0}, \quad \forall k \in \mathcal{I}\backslash\chi. \tag{12c}$$

In this problem, we also maximize the miner's utility as presented in (12a) under the IR and IC constraints shown in (12b) and (12c), respectively.

Let $\chi_i$ represent any subset of user types in $\mathcal{I}$, and $\chi^*$ represent the set of user types resulting in the maximal miner utility in the optimal incomplete information scenario. In the following subsection, we provide how to solve the optimal contract, establish the optimal contract for any arbitrary type set $\chi_i$, and give the guideline for finding the optimal type set $\chi^*$.

From (12), the complexity of our problem increases since the number of IR and IC constraints become $I^2$. As presented in Lemma 3, we first reconstruct IR and IC into an equivalently small set of equations to solve this problem. Second, we present the optimal solution considering a set of users $\chi_i$, i.e., $\mathcal{C}_{incomplete}^*$, and evaluate the The probability of having $n_{\chi_i}$

users belonging to the types in $\chi_i$ as:

$$\mathbb{P}(n_{\chi_i}) = \binom{N}{n_{\chi_i}} \mathbb{P}_{\chi_i}^{n_{\chi_i}} (1-\mathbb{P})^{N-n_{\chi_i}}, \tag{13}$$

where $\mathbb{P}_{\chi_i} = \sum_{i\in\chi_i} \mathbb{P}_i$ as the probability of having $i$ users in set $\chi_i$. Let $T_{max}$ and $\theta_{max}$ represent the maximum delay tolerance and confidentiality of user types in $\chi$, such that $T_{max} = \max\{t_i\}_{i\in\chi_i}$ and $\theta_{max} = \max\{\theta_i\}_{i\in\chi_i}$, respectively. Then $\mathcal{C}^*_{incomplete}(\chi_i)$ can be characterized by Lemma 3.

*Lemma 3: For any arbitrary set of types under incomplete information scenario, the optimal contract is given as follows*:

1) $t^*_{max} = T_{max}$,
2) *for all user types in* $\chi_i$*:*

$$\varphi^* = \frac{\theta^2_{max}}{\mathbb{P}(\gamma(\omega(s_i), x), s_{\chi_i})} \left( \frac{\sum_{n_{\chi_i=1}}^{N} \mathbb{P}(n_{\chi_i}) \frac{1}{\sqrt{1+\lambda}}}{\sum_{n_{\chi_i=1}}^{N} \mathbb{P}(n_{\chi_i}) n_{\chi_i} \theta_{max}} \right), \quad \frac{\theta^3_{max}}{\mathbb{P}(\gamma(\omega(s_i), x), s_{\chi_i})} \left( \frac{\sum_{n_{\chi_i=1}}^{N} \mathbb{P}(n_{\chi_i}) \frac{1}{\sqrt{1+\lambda}}}{\sum_{n_{\chi_i=1}}^{N} \mathbb{P}(n_{\chi_i}) n_{\chi_i} \theta_{max}} \right). \tag{14}$$

3) *for any user type* $i \notin \chi_i$, $\phi^* = \mathbf{0}$.

Next, we obtain the optimal user type set $\chi^*$. We show the insight concerning the user types composed in the optimal type set $\chi^*$ for any value of $N$. Generally, we expect the miner to choose user types with higher preference. Nonetheless, the following counter-intuitive results demonstrate that choosing certain user types with lower preferences (excluding others with higher preferences) may increase the miner's utility.

*Proposition 1: For the optimal contract under incomplete information scenario* $\mathcal{C}_{incomplete}$*, there exist user types* $i$ *and* $j$ *such that* $i \in \chi^*, j \notin \chi^*$*, and* $\Pi(\theta_i, t_i) > \Pi(\theta_j, t_j)$.

Proof of Proposition 1 is provided in Appendix. The insights behind Proposition 1 are: 1) choosing a user type with higher preference may not be optimal when the probability that this user type exists is small, 2) the miner's cost is decided by the user types' maximum computational time $t_i$ and the maximum workload $\omega(s_i)$ in the type set. Consequently, the combination of multiple high-preference types may not have an excellent overall performance.

## V. MULTI-DIMENSIONAL CONTRACT AND TRANSACTION ORDERING MECHANISM

After contracting in Section IV, the miner must sequence transactions strategically to obtain maximal payoff. However, this step is challenging to achieve without MEV extraction or the miner's utility trade-offs for user satisfaction. In this section, we present our transaction ordering algorithm with our multi-dimensional contract in Section V-A and the overall system architecture design in Section V-B. Concretely, this approach can help resolve the ordering problems in MEV extractions and decentralized finance (DeFi). It simultaneously aims to address the problems of high gas costs for diverse user transactions types.

### A. TRANSACTION ORDERING MECHANISM DESIGN

As mentioned in the Section IV, blockchain users reach a service-level agreement (SLA) with a miner to include its transaction to the current block [33]. In this case, the SLA document comprises the multi-dimensional contract designed for each user type. This contract design specifies the acceptable delay and confidentiality constraints of user transactions. The miner has to consider its block size and user requirements before ordering transactions. Let the set $\mathcal{Q} = \{Q_1, Q_2, \cdots, Q_z, \cdots, Q_Z\}$ represent $Z$ number of transactions in the mempool and the set $\mathcal{A} = \{1, 2, \cdots, m, \cdots, M\}$ denote $M$ number of miner's blocks. The miner's ordering solution $\mathcal{Q}^*$ can be presented as $\mathcal{Q}^* = \{Q_{1,1}, Q_{2,2}, \cdots, Q_{z,m,}, \cdots, Q_{Z,M}\}$, where $Q_{z,m}$ denotes a transaction $z$ located at the block $m$. Expressly, we formulate the WSS transaction ordering and contracting problem as an optimization problem shown below:

*Problem 3 (WSS Transaction Ordering and Contracting Problem Formulation):*

$$\max_{t_{max},\varphi} \ \mathbb{E}\left\{U_{\varphi_i}\right\} \tag{15a}$$

$$\text{s.t. } W_{\varphi_i} \geq 0, \quad \forall i \in \mathcal{I}, \tag{15b}$$

$$W_{\varphi_i} \geq W_{\varphi_j}, \quad \forall i, j \in \mathcal{I}, \tag{15c}$$

$$\varphi_i \geq \mathbf{0}, \varphi_j \geq \mathbf{0}, \quad \forall i, j \in \chi,$$

$$\varphi_k = \mathbf{0}, \quad \forall k \in \mathcal{I} \backslash \chi$$

$$\sum_{z=1}^{Z} Q_z = \mathcal{Q}, \quad \sum_{z=1}^{Z} \sum_{m=1}^{M} Q_{z,m} = \mathcal{Q}^*. \tag{15d}$$

In this problem formulation, (15a) presents the objective function for maximizing the miner's utility. This objective function is formulated similar to the multi-dimensional contract design from Problem 2, which captures Problems 1 and 2 if the probabilities are known, e.g., $\mathbb{P}_i = 1$. The constraints in (15b) and (15c) represent the IR and IC constraints, respectively. The IC constraint is dropped if the user type falls under the complete information scenario. The constraint in (15d) ensures that the sum of each block size and transaction match the total available block size and transactions.

We divide this problem into two parts, with the first part comprising the solutions provided in Section IV. Essentially, the solutions from Section IV, which comprises (15a)-(15c), reveal each user types' private information needed for our transaction ordering stage. Under the complete information scenario, we offer the contract items from the solution in Theorem 1 under the incomplete information scenario, and under the incomplete information scenario, we offer the contract items from Lemma 3. The second part provides a weighted transaction ordering algorithm to handle subsequent constraints. This weighted transaction sorting algorithm ensures the optimal ordering of transactions to meet the miner's block size requirements and total available transactions on the network.

To understand the modeling preliminaries of our transaction ordering algorithm, we introduce the classification

groups of transactions. Without loss of generality, transactions are grouped into four main categories based on the users' desired confidentiality and delay. These categories include very high priority (VHP), high priority (HP), mid priority (MP), and low priority (LP) transactions [34]. A VHP transaction requires high confidentiality and a short delay compared with an LP transaction with low confidentiality and long delay. HP transactions demand high confidentiality but long delay, and MP transactions require low confidentiality and a short delay. We classify transactions into these categories to determine the maximum delay for each transaction, which can be represented as $\frac{1}{4}t_{max}, \frac{2}{4}t_{max}, \frac{3}{4}t_{max}$, and $t_{max}$ for VHP, HP, MP, and LP transactions, respectively. The transaction categories can be calculated as follows

$$\mathcal{P}_1 = (t_i < 0.5 \times \tfrac{1}{4}t_{max}) \cap (\theta_i > \tfrac{1}{N}\textstyle\sum_{i=1}^{N}\theta_i), \quad (16)$$

$$\mathcal{P}_2 = (t_i > 0.5 \times \tfrac{2}{4}t_{max}) \cap (\theta_i > \tfrac{1}{N}\textstyle\sum_{i=1}^{N}\theta_i), \quad (17)$$

$$\mathcal{P}_3 = (t_i < 0.5 \times \tfrac{3}{4}t_{max}) \cap (\theta_i < \tfrac{1}{N}\textstyle\sum_{i=1}^{N}\theta_i), \quad (18)$$

$$\mathcal{P}_4 = (t_i > 0.5 \times t_{max}) \cap (\theta_i < \tfrac{1}{N}\textstyle\sum_{i=1}^{N}\theta_i), \quad (19)$$

where $\mathcal{P}_1, \mathcal{P}_2.\mathcal{P}_3$ and $\mathcal{P}_4$ represent VHP, HP, MP, and LP transactions, respectively.

In the WSS design, we apply a weight $\sigma_i$ to each transaction expressed as

$$\sigma_i = \frac{t_i \times \theta_i \times \omega(s_i)}{N}, \quad (20)$$

where $\sigma_i$ is the weight of $Q^i$ computed as a product of the delay tolerance specified by the user, the transaction confidentiality, and the transaction workload, this weight determines the position and cost of a particular transaction. In this case, the total number of transactions $Z$ is used instead of the number of particular transaction types to make this approach agnostic and generic for satisfying all three designs in Section IV. We employ the counting sort algorithm to order the weights, which determines the position of each transaction in the mempool based on the weights of each transaction. We use counting sort because it is fast, efficient, and suitable for the nature of our problem [35]. Counting sort also excels at sorting out values that have repeating occurrences in a given set.

Additionally, the time complexity of counting sort algorithm is $O(N_w + K_w)$, where $N_w$ is the number of elements in the input weights and $K_w$ is the range of weights [36]. The time complexity for our multi-dimensional contract can be computed as $O(N)$ [37]. Therefore, the overall system's time complexity can be evaluated as $O(N) + O(N_w + K_w)$. WSS applies a second phase check by timestamp for repeated similar transactions to ensure fairness in terms of the time of arrival. We present algorithm 1 and Section V-B to explain the WSS mechanism design.

**Algorithm 1** The Proposed Weighted Sequencing Service (WSS) Mechanism Algorithm

**Data:** $\theta$ = confidentiality, $t$ = delay, $W$ = user payoffs
$\sigma$ = transaction weights, $N$ = total transaction,
$\omega$ = transaction workload
**input :** $t, \theta, \omega, N$
**output:** $\mathcal{Q}^* = getSolution(arr)$

```
/* sorts transaction weights */
void countSort(arr):
    max = *maxElem(arr.begin(), arr.end());
    min = *minElem(arr.begin(), arr.end());
    range = max - min + 1;
    count(range), output(arr.size());
    for (i = 0; i < arr.size(); i++) do
        count[arr[i] - min]++;
    end
    for (i = 1; i < count.size(); i++) do
        count[i] + = count[i - 1];
    end
    for (i = arr.size() −1; i >= 0; i–) do
        output[count[arr[i] - min] - 1] = arr[i];
        count[arr[i] - min]–;
    end
    for (i = 0; i < arr.size(); i++) do
        arr[i] = output[i];
    end
return
/* contract & sorting execution */
Function main():
    Initialize N;
    /* execute miner's contract */
    [t_i, θ_i] ← executeContract();
    Size(σ_N) ← N;
    /* Weights calculation */
    for (i = 1; i < N; i++) do
        σ_i = (t_i × θ_i × ω(s_i))/N
        arr[i] = σ_i × W_i
    end
    /* sort transaction weights */
    countSort(arr);
return
```

### B. OVERALL MODEL ARCHITECTURE

In our model, users can submit their transactions directly to the DON. To ensure a transparent ordering is available to all miners on the blockchain, the users must simultaneously submit transactions to these multiple nodes. However, we explore an alternative in which the DON monitors the mempool of a target blockchain and selects transactions from it on behalf of a relying SCON [27]. Typically, the DON utilizes web services as its data source and considers the mempool as a data source to produce reports corresponding to the user transactions. We show the flow of our proposed WSS implementation in Fig. 3 and the detailed description in Appendix.

In Fig. 3, users submit their transactions to the mempool in step 1. The DON in step 2 executes the miner's

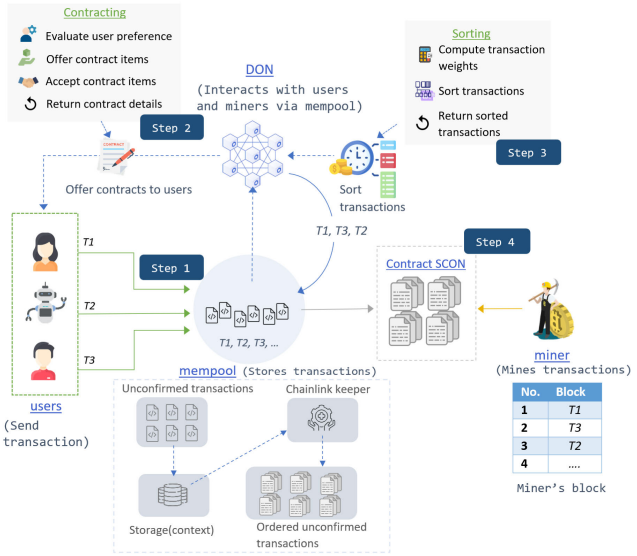


**FIGURE 3.** Users send three transactions, T1, T2, and T3, to the mempool. The DON observes the three transactions, provides a contract to extract $(\theta, t)$, orders them in the mempool, and forwards them to SCON.

contract to reveal each user's type and obtain the various parameters (e.g., confidentiality and delay) needed for the transaction ordering stage.[11] These optimal contract solutions from Theorem 1 and Lemma 3 are coded directly into the chainlink SCs to enhance the latency performance of our mechanism. Next, the DON sequences[12] user transactions sent to the mempool using Alg. 1 in in step 3 of Fig. 3 The DON ingests transactions and then reaches a consensus on their ordering, rather than letting a single node dictate. Both WSS multi-dimensional contracting and transaction ordering algorithm are available through an application programming interface (API) calls to ensure that all nodes obtain the same results [38]. The DON comprises chainlink keepers that employ WSS to sequence these transactions and then forwards the transactions to the mempool for the miners. The miners interact with the newly-ordered user transactions through the blockchain SCON, as shown in step 4 of Fig. 3. To validate the mining sequence, we check the transaction order against the miner's block id (e.g., $Q^i_{\alpha_i} = U^i_{\alpha_i}$, where $U^i_{\alpha_i}$ denotes the miner's transaction order).

Peer-to-peer networks are complicated, and a miner can take advantage of these complications to launch MEV extractions. Also, an adversary with several peers and a fast network connection can frontrun others. However, we can significantly raise the bar for MEV extraction and front-running attempts with our type-revealing contract design and transaction ordering algorithm. Additionally, this incentive mechanism ensures the miner's utility is maximized, offers a good quality of service to each user type, and reduces unethical behaviors on the network.

## VI. EXPERIMENT RESULTS AND ANALYSIS

In this section, we first present the system configuration for our experiments in Section VI-A. Secondly, we introduce the performance metrics and experiment benchmarks for the basis of our experiment in Section VI-B. Finally, we perform extensive experiments to evaluate the performance gains of the proposed mechanism and validate our results in Section VI-C.

### A. SYSTEM CONFIGURATION

In this section, we present the configuration design for our proposed scheme's pervasive experiments and analysis. All experiments are conducted regarding blockchain system architecture and standards [1]. We execute the experiments utilizing Python 3.6 environment and solidity v0.8.0 on a Core i7 CPU computer capacitated with 3.8 GHz and 32GB RAM processor speed and memory. In our evaluation, we employ the blockchain states of Uni and Sushiswap [39], two of the largest on-chain DEXes obtaining 73.27% of the market volume. Consequently, our proposed scheme implementation performs as expected on these two platforms. We use Uniswap as a pricing oracle to fetch the 2, 000 USD/ETH prices for any arbitrary transaction [40]. We assume that the 2, 000 USD/ETH transaction price is zero when none is available, thus neglecting the corresponding transaction. We utilize a price of 2, 500 USD/ETH as of Dec 2021.

To deploy the multi-dimensional contract and sorting algorithm in this system, we execute a chainlink keeper from the chainlink platform with SCs on Ubuntu 18.04 LTS OS. We use solidity programming languages for scripting and writing our SCs. The solutions to our multi-dimensional contract design are directly written into SCs that execute on the selected batch of transactions to extract the user types for transaction ordering. We select a batch of 25 transactions per transaction execution round by default. We also built a web interface with ReactJS [41] to manage adding transactions, chainlink oracle node initialization, and experiment results (see appendix). In this experiment, we initialize 10 chainlink oracle nodes for decentralizing the execution of our contract and sorting algorithms. However, due to the cost involved in deploying transactions on-chain, we limit the majority of transaction executions to off-chain to reduce the cost. Additionally, our mechanism ensures legacy compatibility with the existing blockchain and lower gas costs when implemented using chainlink technologies. Other predefined parameters include: $t_{max} = 10\text{min}, \beta = 12.5, \lambda = 15, s = 10\text{kb}$, and $\mathcal{Q} = 1,000$.

We set the user confidentiality $\theta$ and delay preferences $t$ to [None, LP, MP, HP, VHP], which maps to $\theta$ and $t$ to [2, 4, 6, 8, 10] in the menu options to determine the weights for each transactions, as shown Fig. 5b in Appendix. In the

[11] We use the executeContract() function from Alg. 1, which computes the contract items for each user using solutions from Theorem 1 and Lemma 3.

[12] We execute the weight computation and countSort() function in Alg. 1 to sort the user transactions.

experiment, we consider 300 different user types and 10 different miners on the blockchain network.

### B. PERFORMANCE METRICS AND EXPERIMENT BENCHMARKS

In this section, we introduce the preliminaries, such as experiment criteria and benchmarks, for Section VI-C. The experiment criteria explain how the different results for this discussion were chosen, and the benchmarks explain the measurement metrics that were used to get these results.

#### 1) EXPERIMENT CRITERIA

To analyze our proposed incentive mechanism, we offer the following outcomes in this experiment: contract analysis, miner's utility analysis, users' payoff analysis, fraction of users served, user participation rate analysis, and security analysis. Based on contract evaluation, the contract analysis expands the miner's utility and the user's payoff, providing a comprehensive overview of contracts in our proposed incentive mechanism. To investigate the analysis of our contract design, we present the benchmarks in Section VI-B2. In addition, as stated in Section VI-C1, our contract analysis goes into greater depth regarding how different contract designs affect the miner's utility and the users' payoff.

Based on the benchmark in Section VI-B3, our system performance results indicate the miner's utility and user's payoff as transactions increase, the fraction of users served, the user participation rate, and security analysis. To begin, the miner's utility and the user's payoff as transactions on the blockchain increase indicate the performance of our proposed strategy as transactions on the blockchain increase. We observe and record the effects on the miner's utility and the user's reward to obtain these results, as discussed in Sections 4c and VI-C3, respectively. Second, based on the proportion of user transactions processed and the number of user types on the blockchain platform, the fraction of users served and the user participation rate reflect the performance of our proposed scheme. We actively compute the total number of mined user transactions and user types using the experiment setup from Section VI-A. Finally, the security analysis demonstrates our proposed incentive scheme's unique weaknesses and mitigating mechanisms, as discussed in Section VI-C6.

#### 2) BENCHMARK 1

This benchmark includes the miner and users' evaluation towards complete information, incomplete information, uniform, and no contract. The complete information contract comprises the solution from Section IV-A, and the incomplete information contract describes the results from Section IV-B. To expand the depth of contract analysis studies, we present Appendix's uniform contract, which has a single contract item applicable to all users. In addition, we propose a no contract scenario in which users are not offered any contracts, which essentially characterizes the existing system. This metric provides a framework for evaluating the miner's utility and the users' payoff preferences in relation to the various contract design situations.

#### 3) BENCHMARK 2

This benchmark comprises FSS, A2MM, and MEV auction. Firstly, the authors in [7] introduced an FSS, which orders transactions based on a first-come-first-served approach in the mempool but did not consider the miner's utility optimization and user preferences. Secondly, the authors in [13] presented an A2MM, which performs optimal transaction routing based on transaction fees but fails to consider the user preferences introduced in our work. Lastly, the work in [9] presented MEV auctions to mitigate MEV via auctioning the right to order transactions but failed to consider the users' payoff optimization and preferences. We utilize these metrics to evaluate our system's performance concerning the miner's utility and user's payoff as transactions increase, the fraction of users served, the user participation rate, and security analysis.

### C. DISCUSSION ON IMPLEMENTATION

#### 1) CONTRACT ANALYSIS

From Figs. 4a and 4b, the contract analysis such as miner's utility and users' payoff contrasted with contract evaluation will be discussed.

In this experiment, we investigate the contract preferences based on user type evaluations. Specifically, we analyze the miner's utility and users' payoff considering the various contracts as shown in Figs. 4a and 4b.[13] In Fig. 4a, we can observe that the miner's utility is an increasing function of the users' evaluation, which results from a higher willingness to pay. Users with high confidentiality and tolerance are more likely to pay since they value the mining service, which directly increases the miner's income when the transaction cost is fixed or minimal. Furthermore, as a user's evaluation increases, it becomes more profitable to serve users with higher evaluation than lower evaluation.

Miner's utility achieves the highest under the complete information scenario contract and the lowest under no contract. From Fig. 4b, the users' payoff is high under complete information scenario contract followed by incomplete information scenario contract and uniform contract.[14] This trend's deduction can be attributed to the suitable contract design for each user type under the complete information scenario to produce maximal utility and vice versa. Also, users can capitalize on misrepresentation of its type probability to reduce their cost, as observed in Figs. 4a and 4b. Hence, we can conclude from the results that a miner will prefer a complete information contract design at all times while users gravitate towards incomplete information scenario contract.

[13]Due to lack of space on the figures, we show complete information contracts as complete contracts and incomplete information contracts as incomplete contracts.

[14]Please refer to the Appendix for the uniform contract solution. We consider this as one of our contract analysis benchmarks.

However, our multi-dimensional contract design significantly improves the miner's utility and users' payoff.

#### 2) MINER'S UTILITY ANALYSIS

Fig. 4c shows the performance of our proposed mechanism design based on the miner's utility. In Fig. 4c, we evaluate the miner's utility from (1) considering our proposed WSS with FSS, A2MM, and MEV auction.[15] First, the miner's utility is an increasing function of the number of transactions and user evaluations but a decreasing function of the transaction cost, which is intuitive since a high transaction cost reduces the miner's utility. Our proposed WSS achieves approximately $78.42\% - 84.57\%$ increase in the miner's utility compared with FSS, A2MM, and MEV auction in blockchain networks.

As shown in Fig. 4c, offering a multi-dimensional contract in addition to the sorting algorithm delivers higher miner's utility. The reasons for this trend in results constitute: (i) as shown in WSS's multi-dimensional contract, we illustrate the optimal strategies for the miner to help maximize its utility, (ii) as we have shown from Fig. 4c, our contract design extracts the various user types for appropriate pricing, which improves the miner's utility, (iii) WSS weighs transactions based on user evaluations for transactions after contracting to increase the miner's utility. As the miner has an increased utility from our proposed mechanism, we can conclude that WSS is a promising approach to enhancing miners' utility on blockchain networks.

#### 3) USERS' PAYOFF ANALYSIS

This part of the experiment compares the users' payoff under WSS, FSS, A2MM, and MEV auction as shown in Fig. 4d. The users' payoff increases as transactions increase with WSS achieving up to 64.47% cost reduction compared with FSS, A2MM, and MEV auctions. From (4), the users' payoff is a function of workload evaluation and transaction cost. That is, a highly evaluated transaction presumably yields a higher revenue to users, and the more costly resources a miner will use, the higher the price users will be charged. Highly evaluated transactions often attract high transaction fees and vice versa. This approach provides a more efficient way of identifying and extracting profits for miners from users without indiscriminately increasing transaction fees or encouraging any gas price bidding wars among users. The users' payoff is high under WSS, followed by FSS, A2MM, and MEV auction, which suggests a superior performance for our proposed mechanism.

Similarly, in Fig. 4d, the better results from our proposed mechanism can be attributed to the following reasons: accurate representation of each user type for proper pricing via contracting and decisive transaction ordering based on the representation of each user type. Also, users do not need to worry about mining speed since getting a transaction into the mempool is guaranteed upon accepting contract items designed for each type. Therefore, users with low evaluation can send transactions with low gas prices, which are eventually mined with auditable records showing that transactions were not censored. Additionally, users will have a low evaluation of transactions if transaction fees exceed a threshold, requiring miners to process low-priority transactions. As a result, users' payoff increases based on reduced transaction fees. The DON also re-transmits scheduled user transactions to a relying contract with a high gas price, providing timely processing, and by using batching, the network keeps per-transaction gas costs low.

#### 4) FRACTION OF USERS SERVED

This part of the experiment illustrates the fraction of users served as one of the system performance metrics of our proposed mechanism (WSS). In Figs. 4e and 4f, we compare the fraction of users served in WSS compared with FSS, A2MM, and MEV auction. The fraction of users served determines two main criteria: how many successful transactions are mined per sampled group of transactions and how many transaction types (e.g., VHP, HP, MP, and LP) are served per sampled number of users on the blockchain network. Figs. 4e and 4f describe the fraction of users served based on number of transactions and transaction types, respectively. From Fig. 4e, the fraction of users served increases as the number of transactions increase. This trend indicates that increasing transactions does not overload the blockchain network, which results from a relatively large block size and relatively small transaction processing time.

Also, our proposed mechanism achieves 72% more fraction of users served than FSS, A2MM, and MEV auctions. This trend can be attributed to the various user type characterization from contracting. Our contract designs ensure that any user type is well represented, which enhances the chance of processing more transactions. As deduced in Theorem 1, offering a positive contract to a particular user type yields the same payoff as many user types, making it more reasonable to serve more users instead of specific user types. Also, Proposition 1 can infer that the miner is not fixated on serving high-priority users since this approach does not always yield a higher utility the probability of that user type is low. As a result, the miner is motivated to accept various user types that optimize its utility. Finally, accepting the contract items designed for each user type guarantees that transactions will be mined within specified delay preferences. Failure to process transactions within the specified time duration leads to a negative payoff to the miner's utility. Hence miners are incentivized to execute transactions promptly and as scheduled. Therefore, we can conclude that our proposed mechanism offers a better fraction of user satisfaction than other proposed MEV mitigation schemes.

#### 5) USER PARTICIPATION RATE ANALYSIS

In this experiment, we evaluate the user participation rate based on the duration and number of users engaged on the platform using WSS, FSS, A2MM, and MEV auction.

[15] Implementations for FSS, A2MM, and MEV auction can be obtained from [7], [9], and [13], respectively.

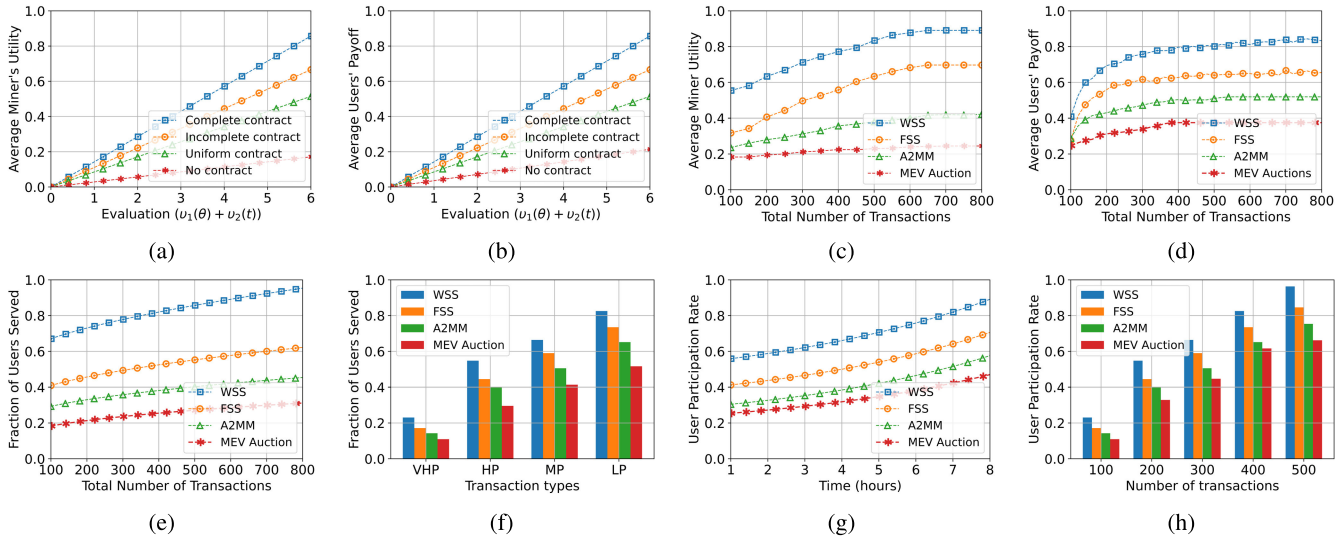


**FIGURE 4.** **Contract analysis: Fig. 4a and Fig. 4b show the contract evaluation considering various contract types. System performance: Fig. 4c - Fig. 4h represent the miner's utility, users' payoff, fraction of users served (number of transactions), fraction of users' served (transaction types), user participation rate (time), and user participation rate (number of transactions), respectively.**

User participation is an essential system performance metric for the sustainability of blockchain networks. Figs. 4g and 4h illustrate the user participation rate based on the experiment duration and number of transactions. From Figs. 4g and 4h, the user participation rate for our proposed system increases as the experiment time and number of transactions increase. WSS achieves up to 68% more user participation than FSS, A2MM, and MEV auction.

Based on the results obtained from Sections VI-C2 and VI-C3, we can conclude that the high user participation rate observed in our proposed mechanism is due to the enhanced miner's utility and user's payoff. The results also point to the potential of individual rationality and incentive compatibility, which provides users a non-negative payoff for any contract and a high payoff for choosing a contract designed for its type. As a result, users are more motivated to participate in blockchain transaction mining than other proposed MEV mitigation strategies.

#### 6) SECURITY ANALYSIS

In this part of the experiment evaluation, we present the security analysis of our proposed mechanism to support the results from Sections VI-C2 to Sections VI-C5. We consider two main concerns: 1) the possibility of miners disregarding the transaction mining order submitted by the chainlink nodes, and 2) MEV centralization and 51% attacks by chainlink nodes when sequencing transactions. Firstly, the miners risk obtaining a negative reward, which prevents them from defaulting in mining transactions beyond the required delay tolerance. Next, miners have minimal arbitrage opportunity ideas of the transactions before executing the mining because the private user information about any transactions is hidden. This concept of obfuscating transactions ensures that miners will not reorder transactions, which will cause penalties in future mining activities. Secondly, our proposed mechanism reduces the risk of MEV centralization and 51% attacks with decentralized transaction ordering by the chainlink nodes. Expressly, no miner or DON has exclusive access to WSS, which prevents the possibility of collusion on the network.

By mitigating these security challenges, WSS obtains a better fraction of users served and participation rate, as shown in Sections VI-C4 and VI-C5.

## VII. CONCLUSION

In this paper, we have studied an essential issue of incentive mechanism design for mitigating frontrunning and transaction reordering in DEXes. To the best of our knowledge, this is one of the first papers to address multi-dimensional private information considering different levels of information asymmetry and transaction ordering mechanisms for blockchain networks. We have introduced a weighted counting sort algorithm for ordering transactions on the blockchain network to avoid MEV centralization and extraction. To incentivize and encourage ethical behaviors, we have presented a multi-dimensional contract design for users with multi-dimensional private information. Moreover, we have implemented a DON to execute our transaction ordering and multi-dimensional contracts on the blockchain platform. We have revealed the effects of the various contract types on the miner's optimal strategies. One of our key contributions is to investigate how to implement our optimal multi-dimensional contract with a transaction ordering algorithm in decentralized blockchain environments. The experiment results demonstrate that our proposed solution offers a $78.42\% - 84.57\%$ increase in the miner's utility and 64.47% cost reduction for users, which incentivizes ethical behaviors on the blockchain network.

## VIII. ADDITIONAL INFORMATION

Appendix proof is available for this paper at: https://github.com/DanielDoe/WSS-appendix

## REFERENCES


[1] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *Int. J. Web Grid Services*, vol. 14, no. 4, pp. 352–375, Feb. 2018.

[2] A. Aspris, S. Foley, J. Svec, and L. Wang, "Decentralized exchanges: The 'wild west' of cryptocurrency trading," *Int. Rev. Financial Anal.*, vol. 77, Oct. 2021, Art. no. 101845.

[3] P. Daian, S. Goldfeder, T. Kell, Y. Li, X. Zhao, I. Bentov, L. Breidenbach, and A. Juels, "Flash Boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability," in *Proc. IEEE Symp. Secur. Privacy (SP)*, San Francisco, CA, USA, May 2020, pp. 910–927.

[4] K. Qin, L. Zhou, and A. Gervais, "Quantifying blockchain extractable value: How dark is the forest?" 2021, *arXiv:2101.05511*.

[5] J. Ghospel. (Apr. 2022). *Maximal Extractable Value (MEV)*. [Online]. Available: https://ethereum.org/en/developers/docs/mev/

[6] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on Ethereum smart contracts (SoK)," in *Proc. Int. Conf. Princ. Secur. Trust*. Berlin, Germany: Springer, Apr. 2017, pp. 164–186.

[7] A. Juels, L. Breidenbach, and F. Tramer. (Sep. 2020). *Fair Sequencing Services: Enabling a Provably Fair DeFi Ecosystem*. [Online]. Available: https://blog.chain.link/chainlink-fair-sequencing-services-enabling-a-provably-fair-defi-ecosystem/

[8] G. Angeris, A. Evans, and T. Chitra, "A note on bundle profit maximization," Jun. 2021. [Online]. Available: https://angeris.github.io/papers/flashbots-mev.pdf

[9] M. Moosavi and J. Clark, "Lissy: Experimenting with on-chain order books," *Cryptogr. Secur.*, vol. 13412, pp. 7–8, Jan. 2021.

[10] *MEV Auctions Considered Harmful*. Accessed: Apr. 12, 2022. [Online]. Available: https://medium.com/offchainlabs/mev-auctions-considered-harmful-fa72f61a40ea

[11] *MEV Auctions Will Kill Ethereum*. Accessed: Apr. 12, 2022. [Online]. Available: https://ethresear.ch/t/mev-auctions-will-kill-ethereum/9060

[12] J. Xu et al., "SoK: Automated market maker (AMM) based decentralized exchanges (DEXs)," vol. 55, pp. 1–50, Mar. 2021, *arXiv:2103.12732*.

[13] L. Zhou, K. Qin, and A. Gervais, "A2MM: Mitigating frontrunning, transaction reordering and consensus instability in decentralized exchanges," 2021, *arXiv:2106.07371*.

[14] M. Bartoletti, J. H.-Y. Chiang, and A. Lluch-Lafuente, "A theory of automated market makers in DeFi," in *Proc. Int. Conf. Coordination Lang. Models*. Cham, Switzerland: Springer, Jun. 2021, pp. 168–187.

[15] H. Lee, K. Sung, K. Lee, J. Lee, and S. Min, "Economic analysis of blockchain technology on digital platform market," in *Proc. IEEE 23rd Pacific Rim Int. Symp. Dependable Comput. (PRDC)*, Taipei, Taiwan, Dec. 2018, pp. 94–103.

[16] P. Bolton and M. Dewatripont, *Contract Theory*. Cambridge, MA, USA: MIT Press, 2005.

[17] J. Moore, "Implementation, contracts, and renegotiation in environments with complete information," *Adv. Econ. Theory*, vol. 1, pp. 182–282, Jan. 1992.

[18] S. Xuan, L. Zheng, I. Chung, W. Wang, D. Man, X. Du, W. Yang, and M. Guizani, "An incentive mechanism for data sharing based on blockchain with smart contracts," *Comput. Electr. Eng.*, vol. 83, May 2020, Art. no. 106587.

[19] M. H. Manshaei, M. Jadliwala, A. Maiti, and M. Fooladgar, "A game-theoretic analysis of shard-based permissionless blockchains," *IEEE Access*, vol. 6, pp. 78100–78112, 2018.

[20] Y. Sun, R. Xue, R. Zhang, Q. Su, and S. Gao, "RTChain: A reputation system with transaction and consensus incentives for e-commerce blockchain," *ACM Trans. Internet Technol.*, vol. 21, no. 1, pp. 1–24, Feb. 2021.

[21] S. Shyamsukha, P. Bhattacharya, F. Patel, S. Tanwar, R. Gupta, and E. Pricop, "PoRF: Proof-of-reputation-based consensus scheme for fair transaction ordering," in *Proc. 13th Int. Conf. Electron., Comput. Artif. Intell. (ECAI)*, Jul. 2021, pp. 1–6.

[22] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22328–22370, 2019.

[23] A. Begum, A. Tareq, M. Sultana, M. Sohel, T. Rahman, and A. Sarwar, "Blockchain attacks analysis and a model to solve double spending attack," *Int. J. Mach. Learn. Comput.*, vol. 10, no. 2, pp. 352–357, Feb. 2020.

[24] M. Kaleem and W. Shi, "Demystifying Pythia: A survey of ChainLink Oracles usage on Ethereum," 2021, *arXiv:2101.06781*.

[25] L. Ma, K. Kaneko, S. Sharma, and K. Sakurai, "Reliable decentralized Oracle with mechanisms for verification and disputation," in *Proc. 7th Int. Symp. Comput. Netw. Workshops (CANDARW)*, Nagasaki, Japan, Nov. 2019, pp. 346–352.

[26] Chainlink. Accessed: Apr. 12, 2022. [Online]. Available: https://docs.chain.link/docs/chainlink-keepers/introduction/

[27] L. Merkin, R. Rezin, and N. Vasilyev, "Architecture of INNOCHAIN, a formally-verified distributed ledger system," in *Proc. Comput. Sci. On-Line Conf.* Cham, Switzerland: Springer, Jul. 2021, vol. 1, no. 1, pp. 96–113.

[28] A. Judmayer, N. Stifter, P. Schindler, and E. Weippl, "Estimating (miner) extractable value is hard, let's go shopping!" *Cryptol. ePrint Arch.*, pp. 1–29, Sep. 2021.

[29] D. Boneh, J. Bonneau, B. Bünz, and B. Fisch, "Verifiable delay functions," in *Proc. Annu. Int. Cryptol. Conf.* Cham, Switzerland: Springer, Aug. 2018, pp. 757–788.

[30] D. Vujicic, D. Jagodic, and S. Randic, "Blockchain technology, Bitcoin, and Ethereum: A brief overview," in *Proc. 17th Int. Symp. INFOTEH-JAHORINA (INFOTEH)*, East Sarajevo, Bosnia and Herzegovina, Mar. 2018, pp. 1–6.

[31] Y. Jiao, P. Wang, D. Niyato, and K. Suankaewmanee, "Auction mechanisms in cloud/fog computing resource allocation for public blockchain networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 9, pp. 1975–1989, Sep. 2019.

[32] N. Houy, "The Bitcoin mining game," *Available SSRN 2407834*, vol. 1, no. 13, pp. 1–9, Mar. 2014.

[33] H. Nakashima and M. Aoyama, "An automation method of SLA contract of Web APIs and its platform based on blockchain concept," in *Proc. IEEE Int. Conf. Cogn. Comput. (ICCC)*, Honolulu, HI, USA, Jun. 2017, pp. 32–39.

[34] Y. Li, S. Jiang, J. Shi, and Y. Wei, "Pricing strategies for blockchain payment service under customer heterogeneity," *Int. J. Prod. Econ.*, vol. 242, no. 1, Dec. 2021, Art. no. 108282.

[35] A. H. Elkahlout and A. Y. Maghari, "A comparative study of sorting algorithms comb, cocktail and counting sorting," *Int. Res. J. Eng. Technol. (IRJET)*, vol. 4, no. 1, pp. 108–282, Jan. 2017.

[36] A. R. Usmani, "A novel time and space complexity efficient variant of counting-sort algorithm," in *Proc. Int. Conf. Innov. Comput. (ICIC)*, Lahore, Pakistan, vol. 1, Nov. 2019, pp. 1–6.

[37] B. Kalkanci, K.-Y. Chen, and F. Erhun, "Contract complexity and performance under asymmetric demand information: An experimental evaluation," *Manage. Sci.*, vol. 57, no. 4, pp. 689–704, Apr. 2011.

[38] A. B. Tran, Q. Lu, and I. Weber, "Lorikeet: A model-driven engineering tool for blockchain-based business process execution and asset management," in *Proc. Bus. Process Manag., Dissertation/Demos/Industry*, Sep. 2018, pp. 56–60.

[39] Sushiswap. Accessed: Apr. 12, 2022. [Online]. Available: https://sushi.com/

[40] Uniswap. Accessed: Apr. 12, 2022. [Online]. Available: https://uniswap.org/

[41] A. Vipul and P. Sonpatki, *ReactJS by Example—Building Modern Web Applications with React*, vol. 2. Birmingham, U.K.: Packt Publishing, Apr. 2016.


**DANIEL MAWUNYO DOE** received the bachelor's degree in computer engineering from the Kwame Nkrumah University of Science and Technology, Kumasi, Ghana, in 2018, and the M.Sc. degree in computer science and engineering from the University of Electronic Science and Technology of China (UESTC), in 2021. He is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering Department, University of Houston, Houston, TX, USA.

His research interests include blockchain, the Internet of Things, cryptography, and game theory.

**JING LI** received the B.S. and M.S. degrees in computer science from North China Electric Power University, Beijing, China, in 2014 and 2018, respectively. She is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering Department, University of Houston, Houston, TX, USA.

Her research interests include blockchain, the Internet of Things, cryptography, and game theory.

**NIYATO DUSIT** (Fellow, IEEE) received the B.Eng. degree from the King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, in 1999, and the Ph.D. degree in electrical and computer engineering from the University of Manitoba, Winnipeg, MB, Canada, in 2008. He is currently a Full Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His current research interests include energy harvesting for wireless communication, the Internet of Things, and sensor bio-adjust networks.

**LI WANG** received the Ph.D. degree from the Beijing University of Post and Telecommunications (BUPT), in 2009. She is currently an Associate Professor with the School of Electronic Engineering, BUPT, where she directs the Laboratory of High Performance Computing and Networks. She received the 2013 Beijing Young Elite Educator for Higher Education Award. From December 2013 to January 2015, she was a Visiting Researcher with the School of Electrical and Computer Engineering, Georgia Tech, Atlanta. Her research interests include wireless networking, secure communications, device-to-device communication systems, and peer-to-peer networks. She has served on the Technical Program Committees of IEEE CCNC 2009, IEEE CCNC 2010, IEEE WCSP 2013, IEEE GLOBECOM 2014, IEEE WCNC 2015, IEEE ICC 2015, IEEE ICNC 2015, and IEEE ICCC 2015.

**ZHU HAN** received the B.S. degree in electronic engineering from Tsinghua University, in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, in 1999 and 2003, respectively. From 2000 to 2002, he was a Research and Development Engineer with JDSU, Germantown, Maryland. From 2003 to 2006, he was a Research Associate with the University of Maryland. From 2006 to 2008, he was an Assistant Professor with Boise State University, ID, USA. He is currently a John and Rebecca Moores Professor with the Electrical and Computer Engineering Department and the Computer Science Department, University of Houston, TX, USA. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid. He received a NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the EURASIP Journal on Advances in Signal Processing in 2015, IEEE Leonard G. Abraham Prize in the field of communications systems (Best Paper Award in IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS) in 2016, and several best paper awards in IEEE conferences. He was an IEEE Communications Society Distinguished Lecturer, from 2015 to 2018. He has been an AAAS Fellow, since 2019, and an ACM Distinguished Member, since 2019. He has been a 1% Highly Cited Researcher, since 2017, according to Web of Science. He is also the Winner of the 2021 IEEE Kiyo Tomiyasu Award for outstanding early to mid-career contributions to technologies holding the promise of innovative applications with the following citation for contributions to game theory and distributed management of autonomous communication networks.