

# SEAL: A Strategy-Proof and Privacy-Preserving UAV Computation Offloading Framework

Yuntao Wang, Zhou Su, Tom H. Luan, Jiliang Li, Qichao Xu, and Ruidong Li

**Abstract**—Due to the limited battery and computing resource, offloading unmanned aerial vehicles (UAVs)’ computation tasks to ground infrastructure, e.g., vehicles, is a fundamental framework. Under such an open and untrusted environment, vehicles are reluctant to share their computing resource unless provisioning strong incentives, privacy protection, and fairness guarantee. Precisely, without strategy-proofness guarantee, the strategic vehicles can overclaim participation costs so as to conduct market manipulation. Without the fairness provision, vehicles can deliberately abort the assigned tasks without any punishments, and UAVs can refuse to pay by the end, causing an exchange dilemma. Lastly, the strategy-proofness and fairness provision typically require transparent payment/task results exchange under public audit, which may disclose sensitive information of vehicles and make the privacy preservation a foremost issue. To achieve the three design goals, we propose SEAL, an integrated framework to address Strategy-proof, fair, and privacy-preserving UAV computation offloading. SEAL deploys a strategy-proof reverse combinatorial auction mechanism to optimize UAVs’ task offloading under practical constraints while ensuring economic-robustness and polynomial-time efficiency. Based on smart contracts and hashchain micropayment, SEAL implements a fair on-chain exchange protocol to realize the atomic completion of batch payments and computing results in multi-round auctions. In addition, a privacy-preserving off-chain auction protocol is devised with the assistance of the trusted processor to efficiently protect vehicles’ bid privacy. Using rigorous theoretical analysis and extensive simulations, we validate that SEAL can effectively prevent vehicles from manipulating, ensure privacy protection and fairness, improve the offloading efficiency, and reduce UAV’s energy costs and expenses with low overheads.

**Index Terms**—UAV, computation offloading, privacy protection, secure, vehicular fog computing.

## I. INTRODUCTION

Recently, unmanned aerial vehicles (UAVs) are gaining growing interest in enabling various smart city applications such as traffic surveillance and disaster rescue [1]–[3]. Thanks to the high agility, low cost, and line-of-sight (LoS) transmissions, UAVs equipped with wealthy sensors can be flexibly dispatched for data collection and environment perception in areas which are inaccessible or hazardous for humans in an on-demand manner [4]. For example, in disaster rescue, UAVs usually follow preset flying routes to visit all disaster sites and perform several missions (e.g., survivor detection and target tracking) at each location [5].

Due to the size and weight limitations, UAV’s battery capacity is usually constrained. For instance, the battery capacity of a small UAV with a payload of 300g is about 5200mAh, which supports a maximum flight endurance of 90 minutes [3]. Moreover, the real-time compute-intensive tasks such as image

Yuntao Wang, Zhou Su, Tom H. Luan, and Jiliang Li are with the School of Cyber Science and Engineering, Xi’an Jiaotong University, Xi’an, China

Qichao Xu is with the School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, China

Ruidong Li is with the College of Science and Engineering, Kanazawa University, Kanazawa, Japan

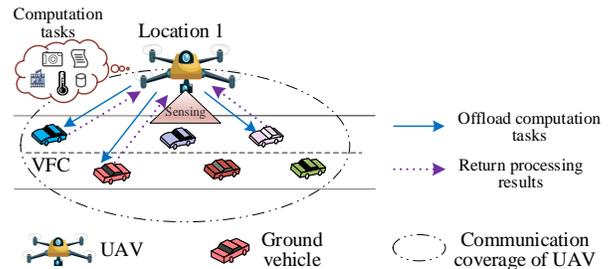


Fig. 1. An illustrating example of UAV computation offloading based on vehicular fog computing (VFC).

and video processing can greatly shorten UAVs’ endurance time, thereby restricting their flying time and distance. Thereby, efficient computation offloading is urgently needed for UAVs in performing persistent missions. In the literature, UAV’s computation missions are conventionally offloaded to the remote cloud for processing [6], [7], and then the computing results are delivered back to the UAV. Nevertheless, it incurs a long network delay in data transmission to/from the remote cloud, especially for latency-sensitive tasks such as traffic monitoring. To trade off the response latency and endurance time, a plausible solution is offloading UAVs’ heavy computations to fog infrastructures [8], [9] such as roadside units and Wi-Fi access points. However, it highly depends on and bears the high deployment cost of additional ground infrastructures. Besides, the computing capacity of fog nodes is also limited to perform the heavy offloaded tasks before the expiration time. Fortunately, vehicular fog computing (VFC) has been envisioned as a feasible and cost-effective solution by exploiting ground vehicles as moving fog nodes and offloading deadline-driven tasks to nearby vehicles with idle computation resources [2], [5], as shown in Fig. 1. Under the VFC paradigm, due to the proximity to end users, controllable vehicle mobility, and dense geographical distribution, it brings more convenience in offloading computation tasks produced by UAVs with reduced service delay.

To practically deploy VFC for collaborative computation offloading, incentive design is one of the fundamental issues. As participating in such offloading tasks usually consumes considerable computation and battery resources, vehicles will be reluctant to share their computing resource without satisfactory incentives. So far, various incentive mechanisms [10]–[12] have been proposed to motivate users’ cooperation in air-ground networking and lots of them are based on auction theory, in which the UAV is the service buyer and vehicles are service sellers. However, selfish and strategic vehicles may misreport their participation costs and submit untruthful bids to claim more compensations so as to conduct market manipulation [13], thereby damping the enthusiasm of honest vehicles and raising the necessity of strategy-proof (or truthful) auctions. In addition to strategy-proofness,

fairness is another essential concern that may hinder vehicles' participation [14]. In practice, vehicles may deliberately withdraw assigned tasks to seek high revenues. For example, a malicious vehicle with constrained computing resource may participate in multiple auctions for different UAVs, and it may abort the currently assigned task for a certain UAV and perform another assigned computing task for another UAV to gain higher benefits. Meanwhile, the ignorance of fairness in auction design may cause an exchange dilemma due to a lack of mutual trust, where neither the UAV nor vehicles are willing to initiate the payment/results transfer. Besides, the strategy-proofness and fairness provision typically require transparent exchange of payment and task results under public audit, which may leak a great deal of vehicle users' sensitive information (including cost type, task preference, and resource capacity). Thereby, vehicle's competitiveness and resource strategies can be estimated by other rivals or adversaries, making the privacy preservation a foremost issue.

In the literature, existing privacy-preserving auction approaches [15]–[20] are mainly based on the common assumption that participants will not abort the assigned tasks and there is no friction in exchanging the task results and rewards, which usually do not hold in realistic environments. The violation of fairness can cause the exchange dilemma (e.g., refuse to pay for computing results) and malicious dropout (e.g., abort assigned tasks), thereby causing loss to honest users and a task failure. Besides, existing mechanisms mainly offer single-parameter truthfulness (i.e., prevent strategic bid prices) for homogeneous (or identical) tasks/items. Under vehicle-assisted UAV computation offloading scenarios, the network environment can be time-varying, causing high heterogeneity and dynamics of tasks in terms of vehicular dwell time and offloading latency. In addition, UAVs usually require truthfulness for both computation resource supply and bid price of vehicles, resulting in a demand for combinatorial truthfulness. Hence, it remains an open and vital issue to design a combinatorial strategy-proof computation offloading framework with fairness and privacy guarantees for UAVs under the VFC.

In this paper, we propose SEAL, an integrated framework to promote Strategy-proof, fair, and privacy-prEserving UAV computation offloading for general UAV applications. Specifically, we first present a VFC-based collaborative architecture to facilitate UAVs' on-demand computation offloading, and then investigate a single-minded reverse combinatorial (SRC) auction framework for computation task scheduling with combinatorial truthfulness guarantees of resource supply and bid price. Afterward, to ensure fairness in the entire auction cycle, we resort to the smart contract technology and devise an on-chain fair exchange protocol to ensure both *exchange fairness* (i.e., prevent exchange dilemma in the payment/task results delivery) and *participation fairness* (i.e., prevent malicious dropout of assigned tasks) among distrustful parties. When implementing SRC auctions in smart contract scripts, it requires vehicles' truthful bid input to be public for audit, which violates the privacy of vehicles. An off-chain auction execution mechanism is further developed in smart contract systems to effectively preserve vehicles' bid privacy with the assistance of the trusted processor equipped on the auctioneer (i.e., UAV). In addition, for efficient on-chain and off-chain orchestration, we devise a commit-then-claim mechanism with batch payment in smart contracts to ensure transactional atomicity and enhance trading efficiency under frequent micropayments.

To summarize, the main contributions are three-fold.

- We propose SEAL to facilitate secure and efficient UAV computation offloading with two improvements: 1) a VFC-based SRC auction mechanism with high flexibility, on-demand deployment, and low response delay; and 2) an on-chain and off-chain cooperation mechanism to protect user privacy and enable fairness in the entire auction cycle with low system overheads.
- We consider the network dynamics due to the high mobility of UAVs and vehicles in the auction design to achieve both combinatorial strategy-proofness and near-optimal UAV cost minimization in practical offloading applications. We design a series of novel fair protocols based on smart contracts to forbid selfish bidders and ensure trust-free delivery of payments (to vehicles) and task results (to UAV). To further improve efficiency in multi-round SRC auctions, a commit-then-claim mechanism with hashchain-based batch payment is developed to reduce operational cost of smart contracts by sequentially delivering hash values (i.e., paywords) as micropayment commitments in an off-chain manner and claiming the due payment in an on-chain manner.
- We theoretically analyze the property of SEAL and rigorously prove its capability in privacy protection, fairness, combinatorial strategy-proofness, and computation efficiency. We also conduct extensive simulations to verify the feasibility and effectiveness of SEAL. Numerical results show that SEAL can reduce system overhead, alleviate UAV's cost, defend against strategic vehicles, and enhance offloading efficiency, compared with conventional schemes.

The remainder of this paper is organized as follows. Section II reviews the related work. Section III introduces the system model. We elaborate the detailed design of SEAL and theoretically analyze its property in Section IV. We present the numerical results in Section V. Section VI closes the paper with conclusions.

## II. RELATED WORKS

### A. Computation Offloading Mechanisms for UAVs

Various efforts on computation offloading have been made in wireless networks. Xiong *et al.* [23] model the offloading of proof-of-work (PoW) tasks from resource-limited miners to cloud/fog servers as a Stackelberg game and analyze the Stackelberg equilibrium. Ng *et al.* [24] present a double auction to match vehicles' required computation resources with edge servers under coded distributed computing paradigms. Gao *et al.* [25] propose a truthful auction mechanism to offload graph jobs efficiently under the vehicular cloud computing paradigm. The preceding works, however, are inapplicable to UAV networks with 3D mobility and complex aerial-ground dynamics. Furthermore, neither fairness nor the preservation of entities' privacy are taken into account.

Recently, a mass of works have been reported on computation offloading mechanisms for energy-limited UAVs. By formulating the dynamic offloading problem as a Markov process, Callegaro *et al.* [26] derive the optimal policies for UAVs to partially offload the computation missions to urban fog nodes with consideration of node competition and server congestion. Bai *et al.* [27] investigate a fog computing-assisted efficient task offloading mechanism for UAV swarms to extend their battery recharging time under partial and full offloading scenarios. Hou *et al.* [28]

TABLE I  
EXISTING TRUTHFUL AND PRIVACY-PRESERVING AUCTION APPROACHES: A COMPARATIVE SUMMARY

Ref.	Auction Type	Bid Price Truthfulness	Combinatorial Truthfulness	Bid Privacy Protection	Participation Fairness	Exchange Fairness	Task/Item Heterogeneity	Comp.&Comm. Overhead	Bid Utility (Availability)	Scenario
PISA [15]	single	✓	×	✓	×	×	×	High	High	spectrum market
ARMOR [16]	combinatorial	✓	×	✓	×	×	✓	High	High	spectrum market
PS-TAHES [17]	double	✓	×	✓	×	×	✓	High	High	spectrum market
V2GEx [18]	—	×	×	✓	×	✓	×	High	High	V2G
Liwang's [19]	reverse	✓	×	×	×	×	✓	Low	High	vehicular comp. offloading
BidGuard [20]	single	✓	×	✓	×	×	×	Low	Low	MCS
Wang's [21]	reverse	✓	×	✓	×	×	×	Low	Low	MCS
Trustee [22]	Vickrey	✓	×	✓	✓	×	—	Medium	High	General
SAFE [14]	single-round	✓	×	✓	✓	✓	—	Medium	High	General
<b>SEAL</b>	<b>SRC</b>	✓	✓	✓	✓	✓	✓	Low	High	UAV comp. offloading

Note 1: "✓" means support; "×" means not support; "—" means not mentioned; "comp." means computation; "comm." means communication.

Note 2: "Comp.&Comm. Overhead" is evaluated under our UAV computation offloading scenario with high-frequency resource trading.

study the optimal computation task assignment problem for UAV fleets under urban fog computing environment and design a genetic algorithm to obtain the near-optimal solution for energy consumption minimization. Liwang *et al.* [29] design a novel futures trading paradigm for onsite resource trading between UAVs and ground edge servers to relieve trading failures, latency, and unfairness. They also present two algorithms for optimal forward contract design and transmit power optimization. Sacco *et al.* [30] investigate a feasible reinforcement learning based method to offload UAVs' computation tasks to ground edge clouds.

Distinguished from the above works on UAV computation offloading to fixed fog servers or remote cloud servers, our work aims to design an efficient and on-demand scheme by harnessing idle computing resource shared by ground vehicles (referred to as VFC) to offload heavy computation missions produced by UAVs. In our previous work [5], we study a VFC-based task offloading framework for UAVs in disaster scenarios and design a stable one-to-one matching algorithm for task scheduling. Nevertheless, the fairness and privacy issues in task scheduling are ignored in [5].

### B. Strategy-Proof and Privacy-Preserving Auction Mechanisms

There have been a number of recent efforts on strategy-proof and privacy-preserving auction mechanisms, which mainly depend on advanced cryptographic mechanisms (e.g., zero-knowledge proof (ZKP) and homomorphic encryption (HE)) and differential privacy (DP). Chen *et al.* [16] propose ARMOR which leverages cryptographic tools including HE and garbled circuits to protect users' location and bid information in combinatorial spectrum auctions while considering spectrum reusability. Wang *et al.* [17] present a secure double auction named PS-TAHES for spectrum redistribution based on HE and garbled circuits to enable privacy-preserving bid multiplication, comparison, and sorting matrix operations. Wan *et al.* [18] develop V2GEx, a blockchain system with ZKP support to ensure exchange fairness and payment privacy for electric vehicles in vehicle-to-grid (V2G) energy services. Nevertheless, these approaches based on advanced cryptography may introduce large system overheads and consume considerable resource for energy-limited UAVs. Lin *et al.* [20] present a differentially private truthful auction architecture named BidGuard in mobile crowdsensing (MCS) to prevent adversaries from deducing users' private information via inference attacks. Wang *et al.* [21] propose a differentially private reverse auction framework for MCS under an untrusted auctioneer, where the exponential mechanism is employed to locally obfuscate users' bids to prevent inference attacks. However, these solutions built

on DP may result in a large bid utility decrease for practical use during the perturbation process, thereby deteriorating the auction efficiency in winner and payment determination.

Recently, there has been a surge in interest in combining smart contract and trusted execution environment (TEE) technologies to protect the bid privacy in auctions [14], [22], [31]–[33]. Galal and Youssef [22] develop Trustee to fully preserve bid privacy in Vickrey auctions by integrating the Ethereum and an Intel SGX enclave. In [22], bidders send their encrypted bids to the smart contract within the bidding interval, and the enclave executes the auction program and produces a signed transaction (including winners and payments) to the smart contract. Brandenburger *et al.* [31] identify that smart contracts run inside TEEs are vulnerable to rollback attacks and present a secure architecture implemented by Hyperledger Fabric for smart contract execution within Intel SGX with rollback prevention. Wang *et al.* [32] leverage the smart contract to replace the untrusted auctioneer to run spectrum auctions and utilize Intel SGX and Pedersen commitment to preserve bid privacy for public verification in smart contracts. Chen *et al.* [14] develop a general fair and privacy-preserving auction framework named SAFE based on smart contracts and Intel SGX, where four representative single-round auction formats are utilized as examples to show the framework design.

However, the above works [22], [31]–[33] do not consider the threats arising from the exchange fairness in the auction, which may discourage honest bidders from truthfully participating in auctions. Although the work [14] considers the exchange fairness in system design, it primarily applies to general single-round auctions and can result in high system costs (e.g., high Gas fee) for multi-round auctions, particularly for UAV computation offloading scenarios with highly frequent task offloading and highly dynamic network environment. In opposite, our SEAL implements a fair exchange protocol including a hashchain-based batch payment mechanism (to reduce operational cost of smart contracts) and a commit-then-claim mechanism (to coordinate on-chain fair exchange and off-chain payword delivery) for UAV computation offloading applications. Besides, for heterogeneous UAV computation tasks, we design a novel multi-round SRC auction with combinatorial truthfulness guarantee. Our SEAL ensures a wide range of security targets in a trust-free manner with much improved efficiency than its alternatives. A comparison of our work with other competing approaches is given in Table I.

### III. SYSTEM MODEL

In this section, we describe the system model by discussing the network model, mobility model, auction model, threat model, and

TABLE II  
SUMMARY OF NOTATIONS

Notation	Description
$\mathbb{I}$	Set of ground vehicles serving as VFC nodes.
$\mathbb{J}_n$	Set of computation tasks at location $n$ .
$\mathbb{W}$	Set of winners of task allocation.
$\mathbb{G}_i$	Set of allocated tasks to winner $i$ .
$\mathfrak{S}_{j,n}$	UAV's $j$ th task at location $n$ .
$\mathfrak{R}$	Radius of A2G/G2A communication coverage of UAV.
$s_{j,n}$	Data size of task $\mathfrak{S}_{j,n}$ .
$\varphi_{j,n}$	Task urgency degree or processing priority.
$\tau_{j,n}$	Task completion deadline.
$\zeta_{j,n}$	Computing intensity.
$\bar{h}_n$	Flying altitude of UAV at location $n$ .
$K$	Number of evenly divided time slots with interval $\Delta_t$ .
$\bar{v}_n[t_k]$	UAV's flying speed at time slot $k$ at segment $n$ .
$\bar{v}_{\text{veh}}$	Average vehicular speed.
$\mathcal{B}_i$	Combinatorial bid of vehicle $i$ .
$\Gamma_i$	Feasible task bundle of vehicle $i$ .
$\chi_i^j$	Amount of computing resources of vehicle $i$ in task $\mathfrak{S}_{j,n}$ .
$b_i^j$	Bidding price of vehicle $i$ for task $\mathfrak{S}_{j,n}$ .
$\beta_i^j$	Binary task allocation variable.
$p_i^j$	Payment to winner $i$ for $j$ th task.
$\Theta(\chi_i^j)$	Cost of vehicle $i$ in task offloading.
$E_n$	Energy consumption of the UAV at location $n$ .
$\varpi$	Weight factor of UAV's energy cost and payment.
$T_{j,n}$	Task completion time of mission $\mathfrak{S}_{j,n}$ .
$\tau_{j,n}^R$	Residual dwell time of vehicle $i$ in UAV's coverage.
$\mathbb{C}_{j,n}$	Feasible candidate set of task $\mathfrak{S}_{j,n}$ .
$F(\chi_i^j, b_i^j)$	Marginal cost factor of vehicle $i$ .
$\bar{b}_{i,j}$	Virtual bidding price of vehicle $i$ .

design goals. Table II summarizes the key notations in this paper.

#### A. Network Model

Fig. 1 illustrates a typical scenario where a UAV is dispatched for data sensing missions along a pre-determined transit route containing  $N$  locations of interest. The UAV starts from location 1, then sequentially moves to the next location by following the straight-line trajectory [8], and finally ends at location  $N$ . When the UAV arrives at the sensing location  $n$ , it hovers over this location to capture the sensory data (e.g., videos and images) of the task area with onboard sensors, then it generates a set of  $\mathbb{J}_n = \{1, \dots, j, \dots, J_n\}$  computing missions (e.g., image and video processing). For improved endurance capability, these missions can be offloaded to VFC nodes (i.e., ground vehicles with idle computing resources) for processing via air-to-ground (A2G) links, and the processing results are sent back to the UAV via ground-to-air (G2A) links<sup>1</sup>. The radius of A2G/G2A communication coverage of the UAV is denoted as  $\mathfrak{R}$ . Here, each mission  $j \in \mathbb{J}_n$  can be described by a 4-tuple:

$$\mathfrak{S}_{j,n} = \langle s_{j,n}, \varphi_{j,n}, \tau_{j,n}, \zeta_{j,n} \rangle, 1 \leq j \leq J_n, \quad (1)$$

where  $s_{j,n}$  (in bits) is the task size,  $\varphi_{j,n}$  is the urgency degree indicating the priority for task processing,  $\tau_{j,n}$  (in seconds) is the task completion deadline, and  $\zeta_{j,n}$  (in CPU cycles/bit) is the computing intensity.

#### B. Mobility Model

During the execution of task  $\mathfrak{S}_{j,n}$ , the UAV is assumed to hover in the sky at the constant flying altitude  $\bar{h}_n$  to avoid frequent ascending and descending for minimized energy consumption, where  $\bar{h}_n$  is the minimum altitude suitable for the working area and can avoid all collisions and blockages [8]. Thereby, we only

<sup>1</sup>When vehicles are not available in the UAV's communication range or the computing resources of vehicles are not sufficient, UAV's computation tasks can be alternatively offloaded to the remote cloud, as in conventional works [6], [7].

have to consider the mobility of vehicles in each sensing location. For simplicity, the time horizon  $T$  is evenly divided into  $K$  time slots [5], and each time slot has an interval of  $\Delta_t = \frac{T}{K}$ . Let  $V_n[t_k]$  be the UAV's flying speed at segment  $n$  between any two successive locations  $n$  and  $n+1$  ( $n, n+1 \leq N$ ) at time slot  $k$ . The set of ground vehicles serving as VFC nodes in the UAV's coverage at location  $n$  is denoted as  $\mathbb{I} = \{1, \dots, i, \dots, I\}$ . Specifically, the vehicle flow (i.e., average number of vehicles entering the UAV's coverage unit time) is denoted as  $\lambda$ , which is assumed to follow a Poisson process with mean arrival rate  $\lambda$  [34]. According to [34], [35], we have  $\lambda = \eta \bar{v}_{\text{veh}}$ , where  $\eta$  (in veh/km) is the vehicle density<sup>2</sup> and  $\bar{v}_{\text{veh}}$  is the average vehicle speed. According to field observations in [35], the vehicular speed-traffic relationship can be described as [34], [36]:

$$\bar{v}_{\text{veh}} = \max \left\{ v_{\text{veh}}^{\min}, (1 - \eta/\eta_{\max}) v_{\text{veh}}^{\max} \right\}, \quad (2)$$

where  $\eta_{\max}$  means the maximum traffic density (i.e., vehicle jam density at which traffic comes to a halt [34]–[36]).  $v_{\text{veh}}^{\min}$  and  $v_{\text{veh}}^{\max}$  are the minimum and maximum vehicle speed, respectively. As seen in Eq. (2), when vehicle density  $\eta$  grows from zero, the vehicle flow  $\lambda$  on the road also grows while the vehicle speed  $\bar{v}_{\text{veh}}$  declines (referred to as the free-flow phase). When the density reaches or above its threshold (i.e.,  $\eta \geq (1 - v_{\text{veh}}^{\min}/v_{\text{veh}}^{\max}) \eta_{\max}$ ), the vehicle traffic becomes congested, which entails low vehicle speed (referred to as the congested-flow phase).

Here, the number of vehicles entering the UAV's coverage at time slot  $k$  is  $\mu^{\text{in}}[t_k] = \lambda \Delta_t = \eta \bar{v}_{\text{veh}} \Delta_t$ . Let  $\mu^{\text{out}}[t_k]$  denote the ratio of vehicles leaving the UAV's coverage. Then, the number of vehicles in the UAV's communication range at time slot  $k$  is denoted as:

$$I[t_k] = \begin{cases} (\mu^{\text{in}}[t_k] + I[t_{k-1}]) (1 - \mu^{\text{out}}[t_k]), & 1 < k \leq K; \\ \mu^{\text{in}}[t_1] (1 - \mu^{\text{out}}[t_1]), & k = 1. \end{cases} \quad (3)$$

#### C. Auction Model

The auction model is employed to schedule UAV's computation task offloading, where ground vehicles are the bidders while the UAV severs as the auctioneer. Each bidder  $i \in \mathbb{I}$  submits its combinatorial bid  $\mathcal{B}_i = \langle \Gamma_i, \vec{\chi}_i, \vec{b}_i \rangle$  including the feasible task bundle  $\Gamma_i$ , the computing resource profile  $\vec{\chi}_i$ , and the bidding price profile  $\vec{b}_i$ . Here,  $\vec{\chi}_i = \{\kappa_i^j \chi_i^j\}_{j=1}^{|\Gamma_i|}$ ,  $\vec{b}_i = \{b_i^j\}_{j=1}^{|\Gamma_i|}$ ,  $\kappa_i^j \in (0, 1]$ ,  $b_i^j \geq 0$  is vehicle  $i$ 's bidding price<sup>3</sup> for task  $\mathfrak{S}_{j,n}$ . For  $t_i^j \notin \Gamma_i$ , we have  $\chi_i^j = 0$  and  $b_i^j = \infty$ . We use  $\mathbf{b}_i^j$  to denote the combinatorial bid for  $j$ th task, i.e.,

$$\mathbf{b}_i^j = (t_i^j, \kappa_i^j \chi_i^j, b_i^j), \forall t_i^j \in \Gamma_i. \quad (4)$$

Then,  $\mathcal{B}_i$  can be rewritten as  $\mathcal{B}_i = \{\mathbf{b}_i^1, \dots, \mathbf{b}_i^j, \dots, \mathbf{b}_i^{|\Gamma_i|}\}$ . In the auction, each vehicle  $i$  bids  $b_i^j$  to sell  $\kappa_i^j \chi_i^j$  amount of computing resources for each task  $t_i^j \in \Gamma_i$ . In addition, vehicles are supposed to be *single-minded* [37], indicating that they can only sell the reported amount of computing resources (i.e.,  $\kappa_i^j = 1$ ) or lose the

<sup>2</sup>In this paper, for ease of analysis, vehicles are driving on the straight road beneath the UAV. The vehicle density in the UAV's coverage is computed as  $\eta = I/L$ , where  $I$  means the number of vehicles on the road segment in the UAV's coverage and  $L$  is the length of the road segment in the UAV's coverage.

<sup>3</sup>The bidding price indicates the bidder's intended payment to be received from the UAV for completing the offloaded computation task. The bidding price is generally determined by the valuation of vehicle user (which is measured by the cost in contributing computation resources).

auction.

**Definition 1 (Single-minded Reverse Combinatorial (SRC) Auction).** Given the task set  $\mathbb{J}_n$  and the bid profile  $\mathcal{B} = \{\mathcal{B}_1, \dots, \mathcal{B}_I\}$  with  $\kappa_i^j = 1$ , a SRC auction  $\mathcal{A}$  can be denoted as a pair of allocation rules  $\vec{\beta}(\mathcal{B})$  and payment rules  $\vec{p}(\mathcal{B})$ . Here,  $\vec{\beta}(\mathcal{B}) = (\beta_i^j)_{|I| \times |\mathbb{J}_n|}$ ,  $\vec{p}(\mathcal{B}) = (p_i^j)_{|I| \times |\mathbb{J}_n|}$ .  $\beta_i^j$  is a binary variable, where  $\beta_i^j = 1$  means bidder  $i$  wins to execute  $j$ th mission and  $\beta_i^j = 0$  means bidder  $i$  loses.  $p_i^j$  is the payment to winner  $i$  for  $j$ th task.

**Definition 2 (Payoff of ground vehicle).** The payoff (or utility) of each bidder  $i \in I$  (i.e., ground vehicle) for task  $\mathfrak{S}_{j,n}$  is the payment minus its monetary resource cost [10], [13], [15], i.e.,

$$\pi_i^j = \pi(\chi_i^j, b_i^j) = \beta_i^j (p_i^j - \Theta(\chi_i^j)), \quad (5)$$

where  $\Theta(\chi_i^j)$  is the cost of bidder  $i$  in sharing  $\chi_i^j$  amount of computing resource, which is private and unknown to others.

#### D. Threat Model

In truthful auctions, the truthful bid information can divulge bidders' true valuations and resource costs. If the private bids are unauthorizedly exposed to the public, malicious bidders may take advantage of these information to seek higher profits and even conduct market manipulation in current or future auctions. For privacy preservation, the UAV is equipped with a trusted processor and it executes the bid collection and SRC auction process inside the TEE, as illustrated in Fig. 2. The permissioned blockchain and smart contracts are exploited to improve the fairness of the exchange of payment and task result in the auction. The following security assumptions are made.

- **Bidders:** The bidders (i.e., ground vehicles) are assumed to be *strategic and selfish* during task offloading. They may deviate from the auction protocol to increase their payoffs and even manipulate the auction by submitting untruthful bids and aborting the assigned tasks deliberately.
- **Auctioneer:** The auctioneer (i.e., the UAV) is assumed to be *semi-honest* (i.e., *passive*). More specifically, the UAV will honestly follow the auction protocol but is curious about participants' private bid information and may refuse to pay after receiving the computing results from vehicles.
- **TEE:** The TEE enclave (deployed on the auctioneer) implemented by Intel SGX is supposed to be *secure*, and the correctness of its sealed data (e.g., bids and programs) can be verified via remote attestations [38].
- **Blockchain:** A permissioned blockchain is maintained by all participants (i.e., UAVs and vehicles), and a secure consensus algorithm is assumed to be executed by participants for blockchain maintenance. The transactions recorded in hash-chained blocks are supposed to be tamper-resistant.

#### E. Design Goals

The goal of our SEAL is to achieve the following desirable properties simultaneously.

**1) Combinatorial strategy-proofness guarantee.** Strategy-proofness (or truthfulness) is the fundamental basis for an auction mechanism [13], whose formal definition is given as below.

**Definition 3 (Combinatorial Strategy Proofness).** An auction mechanism is combinatorial strategy-proof if both combinatorial incentive compatibility and individual rationality are guaranteed.

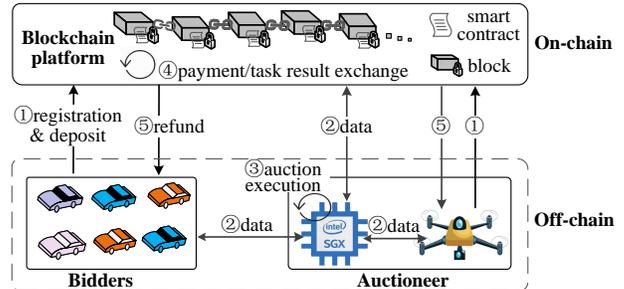


Fig. 2. An illustration of SEAL system.

- **Combinatorial incentive compatibility (CIC).** An auction is combinatorial incentive-compatible if reporting truthful combinatorial bid information  $\vec{\mathcal{B}}_i^{j*} = (\chi_i^j, \Theta(\chi_i^j))$  is the dominant strategy for every bidder  $i \in I$  to maximize its payoff regardless of other bidders' strategy profile  $\vec{\mathcal{B}}_{-i,j,n}$ , i.e.,  $\pi(\vec{\mathcal{B}}_i^{j*}, \vec{\mathcal{B}}_{-i}) \geq \pi(\vec{\mathcal{B}}^j, \vec{\mathcal{B}}_{-i}), \forall \vec{\mathcal{B}}^j \neq \vec{\mathcal{B}}_i^{j*}$ .
- **Individual rationality (IR).** An auction is individual-rational if the expected payoff of each bidder  $i \in I$  participating in the auction is no less than that under non-participation, i.e.,  $\pi_i^j \geq 0, \forall j \in \mathbb{J}_n$ .

**2) Privacy preservation for bidders.** Different from most of the existing auction approaches [15]–[20] that exploit heavy cryptographic tools (such as garbled circuits and HE), or integrate DP methods to preserve the bidding privacy, SEAL aims for an efficient privacy-preserving auction scheme in terms of system overheads and auction efficiency, by leveraging smart contracts with the aid of the trusted processor.

**3) Fair exchange between distrustful participants.** Fairness is another essential target for auction mechanism design to prevent bidders' malicious dropout and eliminate the exchange dilemma. The definition of fairness in the auction is given as follows [14].

**Definition 4 (Fairness).** A fair auction mechanism satisfies both participation fairness and exchange fairness.

- **Participation fairness.** An auction ensures participation fairness if any rational and selfish UAV is stimulated to honestly obey the auction protocols, i.e., they have no incentives to bid untruthfully or abort the auction.
- **Exchange fairness.** An auction ensures exchange fairness if the exchange can be faithfully realized between mutually untrusted entities in the auction.

**4) UAV cost minimization.** SEAL aims to minimize UAV's energy cost and payment in dynamic computation offloading environments with low computation and communication overheads.

## IV. SEAL: OUR DETAILED CONSTRUCTION

### A. Framework Overview

For UAV computation offloading services based on truthful auctions, the implementation of smart contracts requires the replication of all involved data (including the bid information that can reveal bidders' true types and valuations) to all participants for public audit and mutual supervision [39], [40], thereby leaking bidders' privacy. Besides, the high frequency of computation offloading behaviors and the corresponding huge number of micropayments may deteriorate the performance of the smart contract system. We make the following two improvements in SEAL to address these two challenges. One is to move the computation of auction process into an off-chain trusted processor

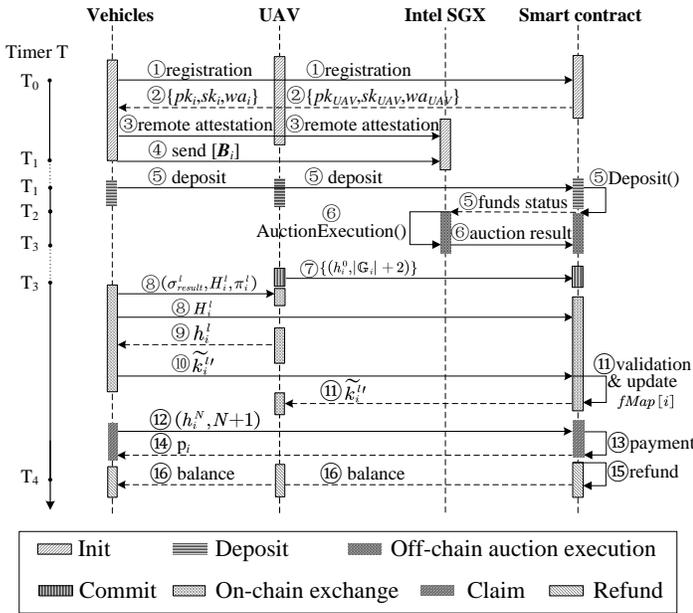


Fig. 3. Sequence diagram of SEAL.

by introducing the concept of local consensus; the other is the hashchain-based batch payment protocol to reduce the cost of supporting frequent on-chain payments. At a high level, as shown in Fig. 2, the design of SEAL orchestrates two parts: *on-chain* and *off-chain*.

- The **on-chain** part automatically executes the auction-based offloading services (i.e., automated delivery of computing results and due payment) by the smart contract with fairness and transparency guarantees, where the service transactions are immutably stored on the blockchain ledgers for audit.
- The **off-chain** part implements the strategy-proof SRC auction mechanism into the TEE to guarantee the auction correctness without violating bidders' privacy. Besides, the UAV makes micropayment commitments based on the hashchain and sequentially sends them to bidder vehicles as the payment authorization after seeing the proof-of-publication of task result delivery on the blockchain, and then the bidder claims the due payment based on the received commitments.

Concretely, as illustrated in Fig. 3, there are 8 phases need to undertake in the design of SEAL, i.e., *Init*, *Deposit*, *Off-chain auction execution*, *Commit*, *On-chain exchange*, *Claim*, *Refund*, and an additional *Timeout* operation. The **init** phase (step ①–④) executes on-chain registration and off-chain initialization. The TEE performs **off-chain auction execution** operations (step ⑥) using the local consensus to privately produce auction results after validating user deposits in the **deposit** phase (step ⑤). The fair exchange is realized by the commit-then-claim mechanism and **on-chain exchange** operations (step ⑧–⑩), where the auctioneer makes hashchain-based micropayments in the **commit** phase (step ⑦) and sequentially sends micropayments to the bidder as the payment authorization after on-chain task result delivery, and the bidder can claim the due payment in the **claim** phase (step ⑫–⑭). After that, the participant can obtain the remaining fund in the **refund** phase (step ⑮–⑯).

### B. Smart Contract Execution with TEE

#### Algorithm 1 Smart Contract with Off-Chain Auction Execution

- 1: **Init:**
- 2: Check the timer  $T$ . If  $T_0 < T < T_1$ , proceed to line 3, otherwise go to line 5;
- 3: bidder  $i \leftarrow \{pk_i, sk_i, wa_i\}$ , UAV  $\leftarrow \{pk_{UAV}, sk_{UAV}, wa_{UAV}\}$ ;
- 4: Each bidder  $i$  sends  $[B_i] \leftarrow \mathbf{Enc}_{pk_{TEE}}(B_i)$  to the TEE.
- 5: If  $T_1 < T < T_2$ , proceed to line 6, otherwise terminate;
- 6: Each seller and buyer invokes **Deposit**().
- 7: **Deposit:**
- 8: Each participant  $m$  sends (deposit,  $\$val_m$ ) to the smart contract.
- 9: **if**  $balance_m \geq \$val_m$  **then**
- 10: Smart contracts transfer  $\$val_m \rightarrow esPool$  from  $wa_m$ ;
- 11: Smart contracts update  $Depo[m] \leftarrow \$val_m$ ;
- 12: **end if**
- 13: **Off-Chain Auction Execution:** # Called by the TEE. #
- 14: If  $T_2 < T < T_3$ , proceed to line 15, otherwise terminate;
- 15: Decrypt  $B_i \leftarrow \mathbf{Dec}_{sk_{TEE}}([B_i])$ ;
- 16: Check the deposit  $Depo[m]$  of each bidder and the UAV;
- 17: Decide the winners and payments and publish  $(\vec{\beta}, \vec{p})$  on blockchain.
- 18: If  $T_3 < T < T_4$ , proceed to line 19, otherwise terminate;
- 19: **Commit:**
- 20: The UAV generates a hashchain  $HashChain_i$  of length  $|\mathbb{G}_i| + 2$  for each winner  $i \in \mathbb{W}$ ;
- 21: The UAV sends the signed metadata  $\{(h_i^0, |\mathbb{G}_i| + 2)\}_{i \in \mathbb{W}}$  to the smart contract.
- 22: **On-Chain Exchange:**
- 23: # Each winning bidder  $i$  executes lines 24-25 for  $l = 1, 2, \dots, |\mathbb{G}_i|$ . #
- 24: Compute  $\sigma_{result}^l \leftarrow \mathbf{Enc}_{pk_{UAV}}(\mathbf{Enc}_{sk_i}(result_i^l))$ ,  $H_i^l \leftarrow \mathbf{Hash}(k_i^l || nonce_i)$ , and a zero knowledge proof  $\pi_i^l$ ;
- 25: Send  $(\sigma_{result}^l, H_i^l, \pi_i^l)$  to the UAV and publish  $H_i^l$  on the blockchain.
- 26: # The UAV executes lines 27-31. #
- 27: **if**  $\mathbf{Verify}(\pi_i^l) = true$  **then**
- 28: Sequentially transmit the signed hash value  $h_i^l$  of  $HashChain_i$ ;
- 29: **else**
- 30: Report the misbehavior of the bidder  $i$  to the smart contract;
- 31: **end if**
- 32: Winner  $i$  sends  $\tilde{k}_i^l \leftarrow \mathbf{Enc}_{sk_i}(k_i^l || nonce_i)$  to the smart contract.
- 33: # The smart contracts execute lines 34-39. #
- 34: Decrypt  $(k_i^l, nonce_i) \leftarrow \mathbf{Dec}_{pk_i}(\tilde{k}_i^l)$ ;
- 35: **if**  $\mathbf{Hash}(k_i^l || nonce_i) = H_i^l$  &  $T \leq \tau_{l,n}$  **then**
- 36: Deliver the key  $k_i^l$  to the UAV;
- 37: **else**
- 38: Update the failed task map of winner  $i$  as  $fMap[i] \leftarrow \{l\}$ ;
- 39: **end if**
- 40: The UAV obtains  $result_i^l \leftarrow \mathbf{Dec}_{sk_{UAV}}(\mathbf{Dec}_{sk_i}(\sigma_{result}^l))$ .
- 41: **Claim:**
- 42: Winner  $i$  sends  $(h_i^N, N+1)$  to the smart contract for validation;
- 43: Smart contracts compute  $p_i = \sum_{l=1}^N p_i^l - \sum_{k \in fMap[i]} p_i^k$ ;
- 44: Smart contracts transfer  $p_i \rightarrow wa_i$  from  $esPool$ ;
- 45: Smart contracts update  $Depo[UAV] \leftarrow Depo[UAV] - p_i$ .
- 46: **Refund:**
- 47: Participant  $m$  sends (refund,  $\$val'_m$ ) to the smart contract.
- 48: **if**  $Depo[m] = \$val'_m$  **then**
- 49: Smart contracts transfer  $\$val'_m \rightarrow wa_m$  from  $esPool$ ;
- 50: Smart contracts update  $Depo[m] \leftarrow Depo[m] - \$val'_m$ ;
- 51: **end if**

Algorithm 1 summarizes the workflow of smart contract execution with the following eight phases.

**Init** (lines 2–6). In the on-chain part, after membership registration at the certificate authority (CA), each registered entity  $m$  (i.e., UAV and vehicle) maintains an account  $account_m$  including its public/private key-pair  $(pk_m, sk_m)$ , wallet address  $wa_m$ , and certificate  $Cer_m = \mathbf{Enc}_{sk_{CA}}(pk_m || T_{stamp} || T_{exp})$  in the permis-

sioned blockchain network. Here,  $\mathbf{Enc}(\cdot)$  is the encryption function,  $sk_{CA}$  is the secret key of the CA,  $T_{stamp}$  is the timestamp of certificate creation, and  $T_{exp}$  is the expiration time. In the off-chain initialization, each bidder can serve as the challenger and verify the correctness of the loaded auction program in the TEE via remote attestations. After successful remote attestations, each bidder  $i$  submits its sealed bid information  $[\mathcal{B}_i] = \mathbf{Enc}_{pk_{TEE}}(\mathcal{B}_i)$  to the TEE before the auction.  $pk_{TEE}$  is the public key of TEE.

**Deposit** (lines 8–12). Each bidder  $i$  sends a deposit transaction to the smart contract:

$$tx_{deposit} = \langle deposit, \$val_m, Depo[m], pk_m, T_{stamp}, \sigma_d \rangle, \quad (6)$$

where  $\$val_i$  is the deposit value sent to the blockchain,  $Depo[m]$  is node  $m$ 's deposit record,  $T_{stamp}$  is the timestamp for transaction creation, and  $\sigma_d$  is the signature on the hash digest of  $tx_{deposit}$ . After checking the account balance  $balance_i$ , the deposit  $\$val_i$  is transferred from  $account_i$  to the escrow pool  $esPool$ .

**Off-chain auction execution** (lines 14–18). In this phase, a critical challenge is the correctness of smart contract execution within TEE. Traditional smart contracts that require the acknowledgment of all entities may contradict the privacy targets if users' private bids are publicly accessible. We observe that the correctness of off-chain auction execution only matters to the contracting parties. Instead, we introduce the *local consensus* by narrowing the universal consensus only to contracting parties, where each involved party in the auction can authenticate the loaded program and data via software attestations.

Concretely, the TEE first decrypts the received encrypted combinatorial bids from all bidders by using its secret key  $sk_{TEE}$ , i.e.,  $\mathcal{B}_i = \mathbf{Dec}_{sk_{TEE}}([\mathcal{B}_i])$ . Then, each participant in the auction can verify the correctness and authenticity of the loaded program and bid information in the TEE enclave by software attestations via the EPID protocol. Only if all involved parties reach an agreement on the loaded auction program and data within the TEE enclave, the off-chain auction execution will continue. During off-chain auction execution, the TEE verifies the deposit value of each participant, and then decides winners  $\vec{\beta}$  and payments  $\vec{p}$  based on the private input (i.e., users' bid information) and the strategy-proof SRC auction program (i.e., Algorithm 2 in Sect. IV-D). Next, the auction results  $(\vec{\beta}, \vec{p})$  are uploaded and immutably stored in the blockchain ledgers. Based on  $\vec{\beta}$ , both the winner set  $\mathbb{W}$  and the set of allocated tasks  $\mathbb{G}_i$  to each winner  $i^* \in \mathbb{W}$  can be computed.

Another critical issue is to ensure the atomicity of transactions, i.e., either both the payment (to vehicles) and the release of task results (to the UAV) are completed simultaneously, or none of them at the cost of their deposits. By using the *commit-then-claim* mechanism and *hashchain micropayment* method, we develop a fair exchange protocol with atomic completion guarantees and batch payment functions in smart contracts, containing the commit phase, on-chain exchange phase, and claim phase, as shown in lines 20–45 in Algorithm 1.

**Commit** (lines 20–21). For every winner  $i \in \mathbb{W}$ , the UAV produces a hashchain with length  $|\mathbb{G}_i| + 2$  and root  $h_i^0$ , i.e.,

$$HashChain_i = \{h_i^{|\mathbb{G}_i|+1} \rightarrow h_i^{|\mathbb{G}_i|} \rightarrow \dots \rightarrow h_i^1 \rightarrow h_i^0\}, \quad (7)$$

where  $|\mathbb{G}_i|$  is the total number of allocated tasks to winner  $i$ . In conventional hashchain approaches [41], [42], the root  $h_i^0$  is public and any element  $h_i^z, \forall z \geq 1$  satisfies  $h_i^z = \mathbf{Hash}(h_i^{z+1})$ . Thereby,

any element  $h_i^z, \forall z \geq 1$  can be employed as the commitment for a constant micropayment unit, and the due payment can be validated by revealing the received commitments and calculated based on the number of received commitments, which is more efficient than those using signatures for verification.

However, in conventional hashchain-based micropayment methods [41], [42], each element in the hashchain represents a commitment for a fixed task payment, which is only applicable for services with constant micropayment. In our work, due to the heterogeneity of tasks in terms of required computation resources and task deadline, the rewards (i.e., payments) for executing distinct tasks are different. Here, we design a novel hashchain micropayment method under the heterogeneous task setting. Specifically, for the first element, we have  $h_i^0 = \mathbf{Hash}(h_i^1)$ . For the remaining elements with  $z \geq 1$ , we further consider the heterogeneous payment in designing the hashchain, i.e.,  $h_i^z = \mathbf{Hash}(h_i^{z+1} || p_i^z)$ .  $p_i^z$  is the micropayment to winner  $i$  for task  $\mathbb{S}_{z,n}, \forall z \in \mathbb{G}_i$ . The element in the hashchain is summarized as:

$$h_i^z = \begin{cases} \mathbf{Hash}(h_i^{z+1} || p_i^z), & 1 \leq z \leq |\mathbb{G}_i|; \\ \mathbf{Hash}(h_i^1), & z = 0. \end{cases} \quad (8)$$

After the construction of all the hashchains for winners (i.e.,  $HashChain_i, i \in \mathbb{W}$ ), the UAV publishes the following claim transaction and delivers it to the smart contract:

$$tx_{commit} = \langle commit, (meta_i, \{p_i^z\}_{z=1}^{|\mathbb{G}_i|}, pk_i)_{i \in \mathbb{W}}, pk_{UAV}, T_{stamp}, T_{exp}, \sigma_{com} \rangle, \quad (9)$$

where  $meta_i = \{h_i^0, |\mathbb{G}_i| + 2\}$  is the metadata of the hashchain,  $T_{exp}$  is the expiry time, and  $\sigma_{com}$  is the signature on the hash digest of  $tx_{commit}$ . Once vehicle  $i$  completes the computation task  $\mathbb{S}_{z,n}$  ( $1 \leq z \leq |\mathbb{G}_i|$ ), the UAV sequentially transmits the (signed) hash value  $h_i^{z+1}$  on the hashchain to bidder  $i$ , as a verifiable commitment for the micropayment  $p_i^z$  in task  $\mathbb{S}_{z,n}$ . Then, every winner  $i$  can use the received hash values in its hashchain as payment authorizations to claim its due reward in conducting the tasks in  $\mathbb{G}_i$  (as analyzed in the Claim phase).

**Remark.** As this exchange process is performed off-chain, there are no intermediate transactions to be processed by the blockchain. Besides, only the final payment between the UAV and the winner is settled by the blockchain platform when the winner completes all assigned computation tasks. In this manner, the efficiency of our SEAL scheme can be significantly enhanced by using the batch technique. Besides, for failed tasks that are aborted by the bidder or failed to complete in time, a failed task map  $fMap[i]$  is maintained on the blockchain ledger for punishment execution and financial settlement with improved robustness.

**On-chain exchange** (lines 23–40). For each task  $\mathbb{S}_{l,n}, l \in \mathbb{G}_i$ , the winner  $i$  first computes its encrypted processing result with a symmetric key  $k_i^l$ , i.e.,  $\mathbf{Enc}_{k_i^l}(result_i^l)$ . Then, it generates a hash value  $H_i^l = \mathbf{Hash}(k_i^l || nonce_l)$ , and produces a ZKP  $\pi_i^l$  using the zero-knowledge succinct noninteractive arguments of knowledge (ZK-SNARK) protocol to commit both the hash value and encrypted result. Here,  $nonce_l = nonce_0 + l$  is used to prevent message reply attacks. Next, winner  $i$  sends a message

$$ResMsg = \langle \mathbf{Enc}_{pk_{UAV}}(\mathbf{Enc}_{k_i^l}(result_i^l)), H_i^l, \pi_i^l \rangle \quad (10)$$

to the UAV, and publishes  $H_i^l$  on the blockchain (line 25). Upon validating the proof  $\pi_i^l$ , the UAV successively reveals the

signed payword  $h_i^l$  of  $HashChain_i$  to winner  $i$  (line 28) as a commitment for payment if the decryption key committed by  $\pi_i^l$  is released within task deadline. Otherwise, if the validation fails, the misbehavior of the corresponding vehicle in delivering a wrong  $ResMsg$  message (which causes a failure of exchange) is reported to the smart contract (line 30). Upon receiving the payword, the vehicle sends its signature of  $k_i^l$  to the smart contract (line 32). Only if the hash of the key matches the commitment, i.e.,

$$\text{Hash}(k_i^l || nonce_l) \stackrel{?}{=} H_i^l, \quad (11)$$

and the task deadline  $\tau_{l,n}$  is not violated, the smart contract releases the key  $k_i^l$  to the UAV (line 36). Otherwise, the task  $l$  is added into the failed task map  $fMap[i]$  of winner  $i$  (line 38). With the key  $k_i^l$ , the UAV can acquire the task results by decrypting the encrypted processing result (line 40).

**Claim** (lines 42–45). Each winning bidder  $i \in \mathbb{W}$  utilizes  $h_i^N$  ( $1 \leq N \leq |\mathbb{G}_i|$ ) received from the UAV and generates the following claim transaction to claim its due payment  $p_i$ :

$$tx_{\text{claim}} = \langle \text{claim}, h_i^N, N + 1, p_i, pk_i, T_{\text{stamp}}, \sigma_{\text{cla}} \rangle, \quad (12)$$

where  $\sigma_{\text{cla}}$  is the signature on the hash digest of  $tx_{\text{claim}}$ . Based on the published root  $h_i^0$ , the failed task maps, and the announced auction results, the smart contract first validates  $(h_i^N, N + 1)$  and then computes the feasible due payment  $p_i$  to winner  $i$  if the validation succeeds, i.e.,

$$p_i = \sum_{l=1}^N p_i^l - \sum_{k \in fMap[i]} p_i^k. \quad (13)$$

Next, it releases the due payment  $p_i$  to winner  $i$  from the escrow pool  $esPool$  and updates UAV's deposit value.

**Refund** (lines 47–51). The smart contract closes the auction by invoking the  $refund()$  to release balances to participants and update system states.

**Timeout** (lines 2, 5, 14, 18). In the smart contract, a timer is set to examine the current time and invoke corresponding functions if the time (e.g.,  $T_0, T_1, T_2, T_3$ , or  $T_4$ ) has expired.

### C. UAV Cost Analysis

In its transit route, the UAV needs to schedule its computation tasks offloaded to VFC nodes for improved endurance time and service quality. The overall cost of a UAV includes its energy consumption and payment at location  $n$ , i.e.,

$$C_n = \varpi E_n + (1 - \varpi) \lambda_p \sum_{i \in \mathbb{W}} p_i^j, \quad (14)$$

where  $\varpi \in (0, 1)$  is the weight factor to balance the energy cost and payment,  $\lambda_p > 0$  is an adjustment parameter,  $\mathbb{W}$  is the set of winners of all the  $J_n$  tasks, and  $E_n$  is the total energy consumption of the UAV at segment  $n$ .

**Energy consumption analysis.** The energy consumption of the UAV at each segment consists of hovering energy cost, propulsion energy cost, and transmission energy cost. Based on [8], the hovering energy cost at location  $n$  can be simplified as  $E_n^{\text{hov}} = P^{\text{hov}} \sum_{j=1}^{J_n} T_{j,n}$ , where  $P^{\text{hov}}$  is UAV's hovering power and  $T_{j,n}$  is the task completion time of mission  $\mathfrak{S}_{j,n}$ . The UAV's propulsion energy cost in flying from location  $n$  to location  $n+1$  is  $E_n^{\text{fly}} = P_n^{\text{fly}} L_n / V_n$ , where  $L_n$  is the distance between locations  $n$  and  $n+1$ . Finally, the transmission energy cost of mission  $\mathfrak{S}_{j,n}$  at location  $n$  is  $E_{i,j}^{\text{tr}} = P^{\text{A2G}} T_{i,j}^{\text{tr}}$ , where  $P^{\text{A2G}}$  is UAV's

transmit power. To summarize, the overall energy cost of the UAV at segment  $n$  is

$$\begin{aligned} E_n &= E_n^{\text{fly}} + E_n^{\text{hov}} + \sum_{i \in \mathbb{W}} E_{i,j}^{\text{tr}} \\ &= P_n^{\text{fly}} \frac{L_n}{V_n} + P^{\text{hov}} \sum_{j \in \mathbb{J}_n} T_{j,n} + \sum_{i \in \mathbb{W}} P^{\text{A2G}} T_{i,j}^{\text{tr}}. \end{aligned} \quad (15)$$

**Task delay analysis.** The task completion time of mission  $\mathfrak{S}_{j,n}$ , i.e.,  $T_{j,n}$ , consists of uplink transmission time  $T_{i,j}^{\text{tr}}$ , task processing time  $T_{i,j}^{\text{comp}}$ , and downlink transmission time. As the size of processing results is relatively small, the downlink G2A transmission time can be neglected [4]. The uplink A2G transmission delay is related to the task size  $s_{j,n}$  and data transmission rate  $\gamma_i^j$ , so we have  $T_{i,j}^{\text{tr}} = s_{j,n} / \gamma_i^j$ . Besides, according to [4], the task processing latency is associated with the shared computing resource  $\chi_i^j$  (in CPU cycles per second), the computing intensity  $\zeta_{j,n}$ , and task size  $s_{j,n}$ . We have  $T_{i,j}^{\text{comp}} = s_{j,n} \zeta_{j,n} / \chi_i^j$ . Therefore, the task completion time can be denoted as

$$T_{j,n} = s_{j,n} \left( \frac{1}{\gamma_i^j} + \frac{\zeta_{j,n}}{\chi_i^j} \right). \quad (16)$$

Due to the mobility of vehicles, the residual dwell time of vehicle  $i \in \mathbb{I}$  in UAV's coverage should satisfy  $\tau_i^R \geq T_{j,n}$  during task offloading. For ease of analysis, it is assumed that the road in the UAV's coverage is straight, and the center of UAV's communication coverage lies on the central axis of the road. Then, we can obtain

$$\tau_i^R = \frac{\mathfrak{R} + \varsigma_{i,n} d_{i,n}}{\bar{v}_{\text{veh}}}, \quad (17)$$

where vehicle  $i$  is driving at a constant speed  $\bar{v}_{\text{veh}}$  in UAV's coverage area (which is regarded as a circle with radius  $\mathfrak{R}$  [5]). In Eq. (17),  $d_{i,n}$  is the horizontal distance between vehicle  $i$  and the UAV at location  $n$ .  $\varsigma_{i,n}$  is vehicle  $i$ 's heading direction, i.e.,

$$\varsigma_{i,n} = \begin{cases} +1, & \text{if vehicle } i \text{ is driving towards location } n; \\ -1, & \text{if vehicle } i \text{ is driving away location } n. \end{cases} \quad (18)$$

**Cost minimization problem.** The optimization problem of the UAV is to minimize its operational cost under the following practical constraints ( $\forall i \in \mathbb{I}, \forall j \in \mathbb{J}_n$ )<sup>4</sup>.

$$\begin{aligned} \min_{\vec{\beta}, \vec{p}} \varpi & \left\{ \sum_{j \in \mathbb{J}_n} \sum_{i \in \mathbb{I}} \left( \frac{P^{\text{hov}} \zeta_{j,n}}{\chi_i^j} + \frac{P^{\text{A2G}} + P^{\text{hov}}}{\gamma_i^j} \right) s_{j,n} \beta_i^j \right. \\ & \left. + P_n^{\text{fly}} L_n / V_n \right\} + (1 - \varpi) \lambda_p \sum_{j \in \mathbb{J}_n} \sum_{i \in \mathbb{I}} p_i^j \beta_i^j \end{aligned} \quad (19)$$

$$T_{j,n} \beta_i^j \leq \min\{\tau_{j,n}, \tau_i^R\}, \quad (20)$$

$$\pi(\chi_i^j, b_i^j) \geq 0, \quad (21)$$

$$\text{s.t. } \pi(\vec{\mathcal{B}}_i^{j*}, \vec{\mathcal{B}}_{-i}^j) \geq \pi(\vec{\mathcal{B}}', \vec{\mathcal{B}}_{-i}^j), \forall \vec{\mathcal{B}}' \neq \vec{\mathcal{B}}_i^{j*}, \quad (22)$$

$$\sum_{i \in \mathbb{I}} \beta_i^j = 1, \quad (23)$$

$$\beta_i^j \in \{0, 1\}, p_i^j \geq 0. \quad (24)$$

The decision variables in the above problem are  $\vec{\beta}$  and  $\vec{p}$ . Constraint (20) is the task deadline constraint of vehicles, indicating that the task completion time  $T_{j,n}$  should be less than both

<sup>4</sup>In the future, we will further investigate UAV's global energy consumption minimization problem across  $N$  sensing locations under battery supply limits. As the energy consumption at each location affects the energy budget at subsequent locations, an energy deficit (indicating the energy consumption deviation from the average energy budget) [8] can be further considered to break the energy supply linkage across  $N$  locations, thereby facilitating the optimization process.

the task completion deadline  $\tau_{j,n}$  and the vehicular residual dwell time  $\tau_i^R$ . Constraints (21) and (22) are IR and CIC constraints for each vehicle, respectively. Constraint (23) indicates that each task can be assigned to at most one vehicle.

### D. Multi-Round SRC Auction Design

Note that the relaxed problem in formula (19) with fixed payment  $\vec{p}$  and constraints (23)–(24) is a typical weighted set cover problem and is NP-hard. Thereby, the problem in formula (19) is NP-hard, and it is nontrivial to attain the optimal solution in polynomial time. Besides, conventional single-parameter truthful auctions [15]–[20] cannot be directly applied, as our SRC auction is a double-parameter truthful auction with both bidding price and computation resource supply truthfulness for heterogeneous tasks. Next, we introduce the sufficient and necessary conditions for satisfying CIC in SRC auctions in the following theorem.

**Theorem 1.** A SRC auction mechanism is combinatorial incentive-compatible if the following two properties hold:

- **Monotonicity.** For each task  $\mathfrak{S}_{j,n}$ , given that other bidders' strategies are fixed, any bidder  $i \in \mathbb{I}$  wins the auction with bid  $(\chi_i^j, b_i^j)$  still wins by bidding  $(\chi_i^{j'}, b_i^{j'})$  with  $\chi_i^{j'} > \chi_i^j$  and  $b_i^{j'} < b_i^j$ .
- **Critical payment.** Any winner  $i$  with bid  $(\chi_i^j, b_i^j)$  of task  $\mathfrak{S}_{j,n}$  is paid the critical payment, i.e., the supremum of all bidding prices  $b_i^j$ 's such that  $(\chi_i^j, b_i^j)$  still wins, i.e.,  $p_{i^*,j}^{\text{CP}} = \sup\{b_i^j \mid \beta_i^j = 1\}$ , given the bids of others remain unchanged.

*Proof:* A similar proof can refer to [37] (in Sect. 5.4). We move the detailed proof to Appendix A in the supplementary material due to the page limitation. ■

Utilizing the rationale provided in Theorem 1, we design a multi-round SRC auction mechanism for approximate UAV's cost minimization with strategy proofness and computational efficiency. Algorithm 2 summarizes the auction workflow, which includes three consecutive phases: candidate group formulation, optimal worker selection, and payment determination.

1) **Candidate group formulation** (lines 4–10). In the task area, due to the high mobility of vehicles and UAVs and the heterogeneity of tasks in terms of task delay and resource demand, part of vehicles may not complete the assigned tasks in time or lack sufficient computation resource for task processing, causing a failure of task offloading. Therefore, efficient task allocation is needed with consideration of the heterogeneity of both tasks and vehicles. Generally, the task with a higher urgency degree needs to be offloaded with a higher priority. By sorting the urgency degree of all tasks in set  $\mathbb{J}_n$  in decreasing order, a new task set is obtained as  $\mathbb{J}'_n$  (line 4).

**Definition 5 (Feasible Task Set).** For each vehicle  $i \in \mathbb{I}$ , its feasible task set  $\Gamma_i$  can be formed by sequentially adding every feasible task from  $\mathbb{J}'_n$  (line 6) as below:

$$\Gamma_i = \left\{ j \mid s_{j,n} \left( \frac{1}{\gamma_i^j} + \frac{\zeta_{j,n}}{\chi_i^j} \right) \leq \min\{\tau_{j,n}, \tau_i^R\} \text{ and } \sum_{j \in \Gamma_i} \chi_i^j \leq \bar{\chi}_i, \forall j \in \mathbb{J}'_n \right\}, \quad (25)$$

where  $\bar{\chi}_i$  is the available computing resource of vehicle  $i$ .

**Definition 6 (Feasible Candidate Set).** For each task  $\mathfrak{S}_{j,n}$ , the

### Algorithm 2 Task Offloading Scheduling with Winner Selection and Pricing in Multi-round SRC Auction

---

```

1: Input:  $\mathbb{I}, \mathbb{J}_n, \mathfrak{S}_{j,n}, \mathcal{B}_i, \mathfrak{R}, \varsigma_{i,n}, d_{i,n}, h_n, \bar{\vartheta}_{\text{veh}}, \bar{\chi}_i, P^{\text{A2G}}, P^{\text{hov}}$ ;
2: Output:  $\vec{\beta}, p^{\text{CP}}$ ;
3: Initialize:  $\beta_i^j \leftarrow 0, p_i^j \leftarrow 0, \Gamma_i \leftarrow \emptyset, \mathbb{W} \leftarrow \emptyset, \mathbb{G}_i \leftarrow \emptyset$ ;
4: Sort all tasks in set  $\mathbb{J}_n$  in decreasing order of the urgency degree and obtain a new task set  $\mathbb{J}'_n$ ;
5: for  $i = 1, 2, \dots, |\mathbb{I}|$  do
6:   Obtain the feasible task set  $\Gamma_i$  using Eq. (25);
7: end for
8: for  $j = 1, 2, \dots, |\mathbb{J}'_n|$  do
9:   # Select feasible workers as candidates.#
10:  Obtain the feasible candidate set  $\mathbb{C}_{j,n}$  using Eq. (26);
11:  for  $i \in \mathbb{C}_{j,n}$  do
12:    Compute the marginal cost factor  $F(\chi_i^j, b_i^j)$  using Eq. (27);
13:  end for
14:  # Find the bidder with the minimum marginal cost factor.#
15:   $i^* = \arg \min_i \{F(\chi_i^j, b_i^j) : i \in \mathbb{C}_{j,n}\}$ ;
16:   $\beta_{i^*,j,n} \leftarrow 1, \mathbb{W} \leftarrow \mathbb{W} \cup \{i^*\}$ ;
17:   $\mathbb{G}_{i^*} \leftarrow \mathbb{G}_{i^*} \cup \{j\}$ ;
18:   $\Gamma_i \leftarrow \Gamma_i \setminus \{j\}, \forall i \in \mathbb{C}_{j,n}$ ;
19:  # Calculate residual computing resource.#
20:   $\bar{\chi}_{i^*} \leftarrow \bar{\chi}_{i^*} - \chi_{i^*}^j$ ;
21: end for
22: for  $j = 1, 2, \dots, |\mathbb{J}'_n|$  do
23:   $\mathbb{C}'_{j,n} \leftarrow \mathbb{C}_{j,n} \setminus \{i^*\}$ ;
24:  Perform winner selection process in lines 11–15 with input  $\mathbb{C}'_{j,n}$  and compute a new winner  $k$ ;
25:  Compute the virtual bidding price  $\tilde{b}_{i^*,j}$  using Eq. (32);
26:  # Calculate critical payment.#
27:   $p_{i^*,j}^{\text{CP}} \leftarrow \tilde{b}_{i^*,j}$ ;
28: end for

```

---

vehicles that satisfy constraint (20) can be included into the feasible candidate set  $\mathbb{C}_{j,n}$  (line 10), which is defined as:

$$\mathbb{C}_{j,n} = \left\{ i \mid s_{j,n} \left( \frac{1}{\gamma_i^j} + \frac{\zeta_{j,n}}{\chi_i^j} \right) \leq \min\{\tau_{j,n}, \tau_i^R\}, \forall j \in \Gamma_i, i \in \mathbb{I} \right\}. \quad (26)$$

2) **Optimal worker selection** (lines 11–20). For every task in  $\mathbb{J}'_n$ , the winner to perform the task is determined based on the marginal cost factor (MCF), which indicates the marginal cost increment (including the energy cost and payment cost) of bidder  $i$  to the UAV in the task (line 12).

**Definition 7 (Marginal Cost Factor).** For every candidate in  $\mathbb{C}_{j,n}$ , its MCF  $F(\chi_i^j, b_i^j)$  in performing task  $\mathfrak{S}_{j,n}$  is defined as:

$$F(\chi_i^j, b_i^j) = \varpi s_{j,n} \left( \frac{P^{\text{hov}} \zeta_{j,n}}{\chi_i^j} + \frac{P^{\text{A2G}} + P^{\text{hov}}}{\gamma_i^j} \right) + (1 - \varpi) \lambda_p b_i^j. \quad (27)$$

In every iteration, the UAV calculates the optimal candidate  $i^*$  for task  $\mathfrak{S}_{j,n}$  that incurs the lowest MCF over  $\mathbb{C}_{j,n}$  as the winner (line 15), i.e.,

$$\beta_{i^*}^j = \begin{cases} 1, & \text{if } i^* = \arg \min_i \{F(\chi_i^j, b_i^j) : i \in \mathbb{C}_{j,n}\}; \\ 0, & \text{otherwise.} \end{cases} \quad (28)$$

In our work, the auction for each task in the set  $\mathbb{J}'_n$  is carried in a sequential manner, where the tasks with higher urgency degrees are executed earlier. After the auction processes of all tasks in  $\mathbb{J}'_n$  finish, the set of winners (line 16) can be derived as

$$\mathbb{W} = \left\{ i \mid \beta_i^j = 1, j \in \mathbb{J}'_n \right\}. \quad (29)$$

Besides, the set of allocated tasks to winner  $i^* \in \mathbb{W}$  (line 17) is

$$\mathbb{G}_{i^*} = \left\{ j \mid \beta_{i^*}^j = 1, j \in \mathbb{J}'_n \right\}, \quad (30)$$

where  $\mathbb{G}_i \cap \mathbb{G}_{i'} = \emptyset, \forall i \neq i'$ .

3) Payment determination (lines 22–28). After determining the winner of each task, we develop a pricing method by employing the critical payment defined in Theorem 1 to decide the payment to each winner with combinatorial truthfulness guarantees.

**Definition 8 (Critical Bidder).** *The critical bidder is defined as the virtual winner  $k$  that wins the auction of task  $\mathfrak{S}_{j,n}$  when excluding the original winner  $i^*$  from the candidate set, i.e.,  $k = \arg \min_i \{F(\chi_i^j, b_i^j) : i \in \mathbb{C}_{j,n} \setminus \{i^*\}\}$ .*

Particularly, for each mission  $\mathfrak{S}_{j,n}$ , a new winner selection procedure is executed over all candidates in  $\mathbb{C}_{j,n}$  except the winner  $i^*$  (line 23). Then, a new winner  $k \in \mathbb{C}_{j,n} \setminus \{i^*\}$  (i.e., critical bidder) can be chosen (line 24). Let  $\tilde{b}_{i^*,j}$  be the *virtual bidding price* of bidder  $i^*$ , which is defined as its maximum bidding price that substitutes bidder  $k$  as the winner (line 25). It indicates that

$$F(\chi_k^j, b_k^j) = F(\chi_{i^*}^j, \tilde{b}_{i^*,j}). \quad (31)$$

By solving the above equation,  $\tilde{b}_{i^*,j}$  can be derived as:

$$\begin{aligned} \tilde{b}_{i^*,j} = & \frac{\varpi}{(1-\varpi)\lambda_p} s_{j,n} \left[ P^{\text{phov}} \zeta_{j,n} \left( \frac{1}{\chi_k^j} - \frac{1}{\chi_{i^*}^j} \right) + \right. \\ & \left. (P^{\text{A2G}} + P^{\text{phov}}) \left( \frac{1}{\gamma_k^j} - \frac{1}{\gamma_{i^*}^j} \right) \right] + b_k^j. \end{aligned} \quad (32)$$

Then, the critical payment  $p_{i^*,j}^{\text{CP}}$  is derived for every winner  $i^*$  by setting its value equal to the virtual bidding price  $\tilde{b}_{i^*,j}$  (line 27).

**Remark.** For each task to be offloaded, we iteratively execute the SRC auction mechanism to obtain the optimal winners and their corresponding payments in each auction round.

### E. Theoretical Analysis

In this subsection, we first show that SEAL satisfies CIC (Lemma 1) and IR (Lemma 2). Based on these two lemmas, we then prove its combinatorial strategy-proofness in Theorem 2. Next, we prove the desired properties of SEAL including fairness and privacy protection in Theorems 3 and 4, respectively. Finally, we analyze the complexities of SEAL in Theorem 5.

**Lemma 1.** *SEAL satisfies combinatorial incentive compatibility (CIC) for both computation resource supply and bidding price.*

*Proof:* Please refer to Appendix B. ■

**Lemma 2.** *SEAL satisfies individual rationality (IR).*

*Proof:* Please refer to Appendix C. ■

**Theorem 2.** *SEAL is a combinatorial strategy-proof auction mechanism.*

*Proof:* According to Lemmas 1 and 2, both CIC and IR properties are satisfied. Based on Definition 1, our SRC auction mechanism ensures combinatorial strategy-proofness. Besides, numerical results in Fig. 9 also validate it. ■

**Theorem 3.** *SEAL can preserve participants' bidding privacy.*

*Proof:* Please refer to Appendix D. ■

**Theorem 4.** *SEAL is a fair auction mechanism, i.e., it guarantees both participation fairness and exchange fairness.*

*Proof:* Please refer to Appendix E. ■

**Theorem 5.** *The computational complexity and communication complexity of SEAL are  $\mathcal{O}(J_n I \log(I))$  and  $\mathcal{O}(I(M \cdot \text{bit}_\pi + \text{bit}_c))$ , respectively.*

*Proof:* We move the detailed proof to Appendix F, as well as summarize the computation and communication complexities in each phase of SEAL in Table III. ■

TABLE III  
COMPUTATION AND COMMUNICATION COMPLEXITY IN SEAL

	Comp. Complexity	Comm. Complexity
Init	$\mathcal{O}(I)$	$\mathcal{O}(I \cdot \text{bit}_c)$
Deposit	$\mathcal{O}(I)$	$\mathcal{O}(I \cdot M)$
Off-chain auction	$\mathcal{O}(\sum_{j=1}^{J_n} I_j + J_n  \mathbb{C}_{j,n}  \log( \mathbb{C}_{j,n} ))$	$\mathcal{O}(M \cdot \text{bit}_p)$
Commit	$\mathcal{O}(\sum_{i=1}^{ \mathbb{W} }  \mathbb{G}_i )$	$\mathcal{O}(M)$
On-chain exchange	$\mathcal{O}(\sum_{i=1}^{ \mathbb{W} }  \mathbb{G}_i )$	$\mathcal{O}(M  \mathbb{W}  \text{bit}_\pi)$
Claim	$\mathcal{O}(\sum_{i=1}^{ \mathbb{W} }  \mathbb{G}_i )$	$\mathcal{O}(M  \mathbb{W} )$
Refund	$\mathcal{O}(I)$	$\mathcal{O}(I \cdot M)$
<b>SEAL</b>	$\mathcal{O}(J_n I \log(I))$	$\mathcal{O}(I(M \cdot \text{bit}_\pi + \text{bit}_c))$

TABLE IV  
SIMULATION PARAMETERS

Param	Value	Param	Value	Param	Value
$N$	30	$K$	1000	$J_n$	[100, 300]
$h_n$	50m	$s_{j,n}$	[3, 9] Mb	$\tau_{j,n}$	[1.0, 2.5]s
$\bar{\chi}_i$	[0.5, 2.0]GC/s	$\varpi$	0.5	$\zeta_{j,n}$	50 C/Mb
$\varphi_{j,n}$	[0.1, 1]	$\mathfrak{R}$	250m	$L_n$	500m
$\gamma_{i,n}$	6 Mbps	$V_{\min}$	2 m/s	$V_{\max}$	20 m/s
$\lambda_p$	40	$\vartheta_{\text{veh}}^{\min}$	30 km/h	$\vartheta_{\text{veh}}^{\max}$	80 km/h
$\phi_i$	[1, 9]	$P^{\text{A2G}}$	0.2W	$P^{\text{phov}}$	500W

## V. PERFORMANCE EVALUATION

### A. Simulation Settings

We conduct simulations on a real-world data set from the mobility traces of taxi cabs in San Francisco [43], which contains GPS coordinates of about 500 taxis gathered over a month in the San Francisco Bay Area. There are 30 sensing locations uniformly distributed in the area. A UAV flies at a fixed altitude 50m and sequentially visits each location via a straight-line trajectory. The number of computation missions produced at each location is randomly selected between 100 and 300. The task size and task deadline follow the uniform distribution within [3, 9]Mb and [1.0, 2.5]s, respectively. The computation intensity is  $\zeta = 50$  CPU cycles/Mb. The idle computing resource of ground vehicles is randomly selected from [0.5, 2.0] GC/s (GC =  $10^9$  CPU cycles) [44]. We define  $\Theta(\chi_i^j) = \phi_i \chi_i^j + c_0$  as the private cost valuation of bidder  $i$  [19], where  $\phi_i$  is its unit cost of computing resource and  $c_0$  is the fixed cost.

The Intel SGX SDK<sup>5</sup> is adopted to implement the SRC auction algorithm, where the SGX enclave serves as the auctioneer and runs the auction algorithm. The software attestation process is implemented where bidders verify whether the auction program is correctly coded and loaded in the enclave. After attestation, bidders send their encrypted combinatorial bids to the SGX enclave which loads the ciphertexts via the *ecall* function. We implement the smart contracts in JavaScript in a local simulated environment using Hyperledger Caliper<sup>6</sup>, which is a widely used customizable benchmarking tool for blockchains such as Hyperledger Fabric and Ethereum. A Caliper adaptor is programmed via Fabric Client SDK using Node.js to interact with the blockchain platform. As the reference implementation, 3 ordering service nodes (OSNs) run the Apache Kafka protocol to reach consensus on generated transactions. Besides, the ZKP is realized based on the open source library libsnark<sup>7</sup>, and the Keccak-256 is adopted as the one-way hash function for efficient hashchain creation. The simulation parameters are summarized in Table IV [8], [44].

<sup>5</sup><https://software.intel.com/content/www/us/en/develop/topics/software-guard-extensions/sdk.html>

<sup>6</sup><https://hyperledger.github.io/caliper/>

<sup>7</sup><https://github.com/scipr-lab/libsnark>

TABLE V  
COMPUTATION AND COMMUNICATION OVERHEADS IN SEAL UNDER SMALL-SCALE AND LARGE-SCALE AUCTIONS

	$J = 10, I = 10$		$J = 100, I = 50$	
	Comp.(ms)	Comm.(KB)	Comp.(ms)	Comm.(KB)
Init	2.6	14.36	2.6	67.01
Off-chain auction	2.8	0.15	45	2.42
Commit	1.3	7.85	7.1	32.53
On-chain exchange	61	9.5	133	93.89
Claim	29	1.59	29	7.97
Total	96.7	33.45	216.7	203.82

TABLE VI  
COMPARISON OF COMPUTATION OVERHEADS IN FOUR SCHEMES ( $J = 100$ )

	Computation overhead (ms)				
	$I = 10$	$I = 20$	$I = 30$	$I = 40$	$I = 50$
ARMOR	$2.6 \times 10^3$	$9.7 \times 10^3$	$23.3 \times 10^3$	$35.2 \times 10^3$	$48.8 \times 10^3$
BidGuard	13.6	24.5	37.4	49.8	62.5
SAFE	188.2	199.4	210.1	229.1	357.8
SEAL	100.1	121.6	147.0	177.5	216.7

TABLE VII  
COMPARISON OF COMMUNICATION OVERHEADS IN FOUR SCHEMES ( $J = 100$ )

	Communication overhead (KB)				
	$I = 10$	$I = 20$	$I = 30$	$I = 40$	$I = 50$
ARMOR	$2.1 \times 10^3$	$4.7 \times 10^3$	$10.2 \times 10^3$	$15.8 \times 10^3$	$21.3 \times 10^3$
BidGuard	19.75	36.05	53.35	69.66	85.96
SAFE	104.3	132.8	178.2	235.8	292.3
SEAL	36.49	68.99	101.08	133.17	165.25

TABLE VIII  
COMMUNICATION COMPLEXITY IN COMMIT-THEN-COMMIT OPERATIONS IN SEAL AND SAFE SCHEMES UNDER HIGH-FREQUENCY TRADING

	Comm. complexity in commit-then-commit operations under high-frequency payment ( $ \mathbb{W}  \ll J$ )
SEAL	$\mathcal{O}(M \cdot  \mathbb{W}  \cdot bit_\pi) \approx \mathcal{O}(M \cdot bit_\pi)$
SAFE	$\mathcal{O}(M \cdot J \cdot bit_\pi)$

## B. System Overheads

We compare the SEAL with the following representative privacy-preserving auction schemes in terms of system overhead.

- *ARMOR scheme* [16]: it utilizes cryptographic tools including HE and garbled circuits to preserve users' privacy in combinatorial spectrum auctions.
- *BidGuard scheme* [20]: it leverages the exponential mechanism in DP for bid perturbation to prevent bid privacy inference in truthful MCS auctions.
- *SAFE scheme* [14]: it leverages TEE for bid privacy protection in general single-round auctions on smart contract systems, where the batch payment and double-parameter truthfulness in multi-round auctions are not supported.

Note that these schemes focus on distinct auction formats in different applications. To be objective and fair, we implement the multi-round SRC auction in the above three schemes under UAV computation offloading scenarios, and other operations follow the original schemes.

**1) Computation & communication overheads.** Table V shows the computation and communication overheads in each auction phase of SEAL for small-scale and large-scale auctions. It can be seen that the execution time and communication cost of SEAL are very low under both small-scale and large-scale scenarios, as the combinatorial bids are processed in plaintext inside the trusted enclave. The on-chain exchange phase occupies a majority of the time and communication cost owing to the creation and verification of ZKP for task result/payment delivery.

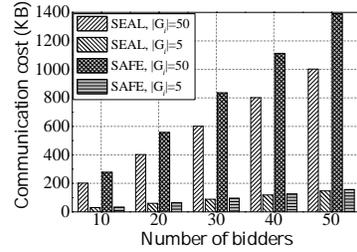


Fig. 4. Communication cost of SEAL and SAFE schemes in commit-then-commit operations under low-frequency and high-frequency payment scenarios.

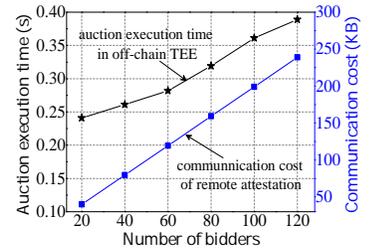


Fig. 5. Off-chain auction execution time in TEE and communication cost of remote attestations with different number of bidders.

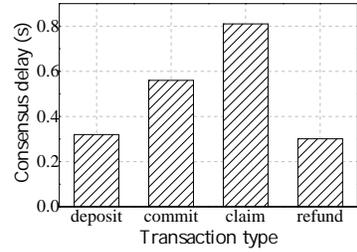


Fig. 6. Consensus delay for different transaction types in the smart contract.

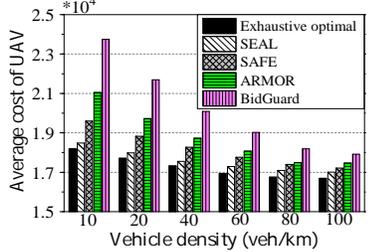


Fig. 7. Average cost of UAV vs. vehicle density in five schemes.

Then we compare the computation and communication overheads of SEAL with other three schemes in Tables VI and VII, respectively. Obviously, BidGuard has the smallest system overheads as it only sends perturbed bids to the auctioneer for winner and payment determination. Meanwhile, a large bid utility decrease may occur in BidGuard for practical use, especially requiring strong privacy provisions (as shown in Fig. 7). Besides, SEAL outperforms ARMOR and SAFE in attaining smaller computation and communication overheads given different number of bidders. For example, when  $I = 30$ , SEAL needs 147ms with about 101KB communication cost, SAFE requires about 210ms with near 178KB communication overheads, while ARMOR spends over 23s with about 10MB communication costs.

Next, we show the performance of hashchain micropayment in SEAL. As seen in Table VIII, under high-frequent payment scenarios, SEAL enjoys a much smaller communication complexity (i.e.,  $\mathcal{O}(M \cdot bit_\pi)$ ) than that in SAFE (i.e.,  $\mathcal{O}(M \cdot J \cdot bit_\pi)$ ) in commit-then-commit operations (i.e., the commit, on-chain exchange, and claim phases). Besides, Fig. 4 compares the communication cost with the SAFE scheme in commit-then-commit operations in multi-round SRC auctions, under both low-frequency ( $|\mathbb{G}_i| = 5$ ) and high-frequency ( $|\mathbb{G}_i| = 50$ ) trading scenarios. Here, the number of bidders varies from 10 to 50.  $|\mathbb{G}_i|$  means the number of winning tasks of bidder  $i$ , and  $|\mathbb{G}_i| + 2$  is the length of  $HashChain_i$ . In Fig. 4, we can observe that SEAL outperforms SAFE in attaining a lower communication overhead especially when the trading frequency is high. The reason is that SEAL integrates the hashchain-based micropayment mechanism to support batch payment to each winning bidder instead of paying at each auction round, thereby alleviating the communication burden in multi-round frequent micropayments.

The auction execution time and communication cost of off-chain auction execution in Intel SGX are evaluated in Fig. 5. It can be seen that, given the number of bidders ranging from 20 to 120, the auction time in TEE is less than 0.4s and the communication

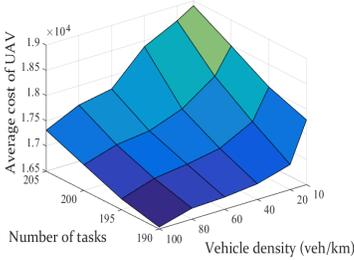


Fig. 8. Average cost of UAV vs. vehicle density, with different number of computation tasks.

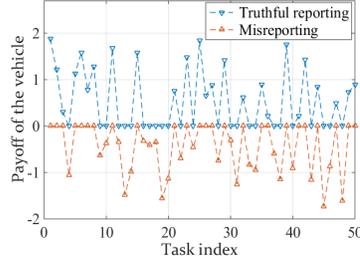


Fig. 9. Payoff of a randomly selected vehicle when it submits truthful bids vs. strategic bids in different tasks.

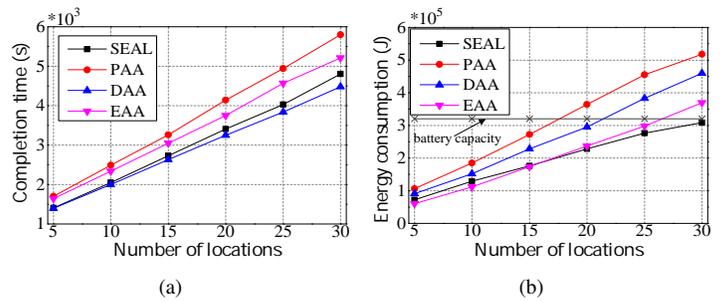


Fig. 10. Comparison of (a) journey completion time of UAV and (b) energy consumption of UAV vs. number of locations in four schemes.

cost in remote attestations between participants and the enclave is less than 250KB, which is efficient for practical deployment.

**2) Consensus delay.** Fig. 6 shows the consensus delay for different transaction types defined in Algorithm 1. Here, for a transaction, its *consensus delay* refers to the latency from being pending to be confirmed in the blockchain. As seen in Fig. 6, the consensus delay is about 0.3~0.8 seconds for different types of transactions, and it is less than 0.81 seconds for all transaction types, which is efficient for practical offloading services.

### C. Economic Efficiencies

**1) Average cost of UAV.** Fig. 7 compares UAV's average cost among five schemes under different vehicle densities. Here, the number of tasks is set as  $J = 200$ , and the linear score function **LIN** is adopted in BidGuard. As seen in Fig. 7, SEAL outperforms BidGuard, ARMOR, and SAFE in acquiring a smaller gap with the exhaustive optimal solution. This is because in BidGuard, users need to upload the perturbed combinatorial bids, instead of the real ones, via the exponential mechanism to ensure DP and prevent inference attacks. Thereby, a large auction efficiency decrease can occur in determining optimal winners and payments based on the perturbed bids. In ARMOR, it consumes considerable computation energy for UAVs in HE operations for bid privacy protection. In SAFE, as analyzed in Table VIII and Fig. 4, it involves higher communication cost for UAVs especially under high  $|\mathcal{G}_i|$  when the vehicle density is low. On contrary, SEAL determines the winners and due payments based on the true bids in plaintext and batch payment method with the help of smart contracts and the trusted processor for frequent micropayments, resulting in a higher offloading efficiency in the auction.

Fig. 8 shows the average cost of UAV in its flying route when the vehicle density increases from 10 to 100 veh/km and the number of tasks increases from 190 to 205. It can be seen that the average cost of UAV decreases with the increase of the vehicle density, and it increases with the increase of the number of tasks. The reason is that the UAV can choose vehicles that offer lower costs when the number of candidate vehicles increases. Besides, according to the objective function in Eq. (19), given the fixed vehicle density, the higher number of tasks can result in a higher cost of the UAV.

**2) Strategy proofness.** Fig. 9 shows the payoff of a candidate vehicle in different tasks. It can be seen that the vehicle's payoff under honest participation (i.e., bidding truthfully) is always non-negative and higher than that under strategic bids (i.e., misreporting an arbitrary bid vector), which validates the strategy-proofness of SEAL in vehicles' bids.

**3) Latency & energy cost in offloading.** The following conventional offloading schemes are used for comparison with SEAL in terms of auction efficiency.

- *Energy-Aware Auction (EAA) scheme* [28]: every task is assigned to the candidate bidder with the minimum energy cost, and the UAV flies at the minimum flying speed  $V_{\min}$ .
- *Delay-Aware Auction (DAA) scheme*: each task is allocated to the bidder with the minimum completion delay, and the UAV's flying speed is fixed at the maximum value.
- *Price-Aware Auction (PAA) scheme*: every task is assigned to the candidate bidder with the minimum bidding price, and UAV's flying speed is randomly selected from  $[V_{\min}, V_{\max}]$ .
- *Cloud-based offloading scheme* [6]: the computing tasks of the UAV are offloaded to the remote cloud for processing at each location. Here,  $\phi_{\text{cloud}} = 8$  and  $\chi_{\text{cloud}}^j = 10$  GC/s.
- *Fog-based offloading scheme* [26]: the computing tasks of the UAV are offloaded to the fixed fog server for processing at each location. Here,  $\phi_{\text{fog}} = 9$  and  $\chi_{\text{fog}}^j = 3$  GC/s [44].

In the above baselines, as the bid privacy protection and fairness are not considered, these offloading schemes have lower computation and communication overheads than our SEAL scheme. To be objective and fair, we only compare them with our SEAL scheme in terms of auction efficiency (e.g., task completion latency and energy cost) in Figs. 10-11.

Fig. 10 shows the task completion time and energy consumption in four schemes with different number of locations (i.e.,  $N$ ). In Fig. 10, as more locations cause higher latency in task processing and UAV's transition, both the completion time and energy consumption in four schemes increase with  $N$ . In Fig. 10(a), DAA attains the smallest task processing delay, and SEAL performs close to DAA when  $N$  is small. In Fig. 10(b), when  $N$  becomes large, both DAA and PAA incur high energy cost and violate UAV's battery limit. Besides, EAA performs better than SEAL when  $N$  is small, while EAA incurs a higher growth rate than SEAL and even violates the battery constraint when  $N$  is large. The reason is that a low flying speed raises UAV's energy consumption for lifting against the force of gravity, while a high speed raises the propulsion energy for moving between locations. In PAA, DAA and EAA, UAV flies at the random, maximum, and minimum speed, thereby increasing the energy cost in transitions and violating the battery constraint when  $N$  is large.

Fig. 11 shows the task completion time and energy consumption in offloading in four schemes with different number of tasks (i.e.,  $J$ ). In Fig. 11(a), the local computing scheme has the smallest completion time when  $J$  is small, and it incurs a higher growth rate than other schemes when  $J$  becomes large. The reason is that

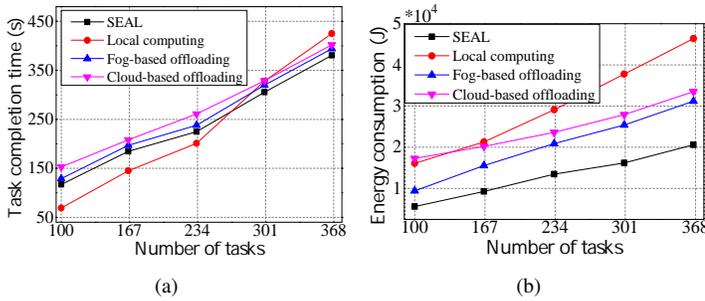


Fig. 11. Comparison of (a) task completion time and (b) energy consumption of UAV vs. number of tasks in four schemes.

UAV's computation capacity is limited and can be fully occupied by the increasing tasks to be processed. Besides, our SEAL attains the smallest task delay when  $J$  is large (in Fig. 11(a)) and the smallest energy cost in offloading (in Fig. 11(b)). It is because that vehicles are provisioned with sufficient computing resources and are more close to the UAV for task execution.

#### D. Key Insights

- Compared with existing representative privacy-preserving auction approaches based on cryptosystems and TEE, our SEAL achieves lower computation/communication complexity, particularly in high-frequency trading settings. Moreover, our SEAL acquires lower average costs for UAVs than existing representative privacy-preserving auctions. Besides, our SEAL enjoys low off-chain auction execution time within TEE and low consensus delay in the blockchain.
- Our SEAL ensures fairness, strategy proofness, and privacy preservation simultaneously. Besides, our SEAL enforces high offloading efficiency in terms of low UAV's cost, low task completion delay, and low energy consumption.

## VI. CONCLUSION

In this paper, we have presented SEAL to address efficient, fair, and privacy-preserving computation offloading for UAV applications. First, we have introduced a VFC-oriented auction-based collaborative mechanism to efficiently offload UAVs' intensive computation missions to ground vehicles while guaranteeing economic robustness. Then, we have implemented a fair exchange protocol in smart contracts to enforce both participation fairness and exchange fairness between distrustful entities. By further integrating TEE into smart contracts, an off-chain auction mechanism has been devised to preserve vehicles' privacy in an efficient manner. At last, simulation results have validated the effectiveness of SEAL in terms of offloading efficiency, cost saving, and system overheads. For future work, we will further extend SEAL to be resistant to collusive vehicles with bid manipulation prevention.

#### APPENDIX A PROOF OF THEOREM 1

**Theorem 1.** A SRC auction mechanism is combinatorial incentive-compatible if the following two properties hold:

- **Monotonicity.** For each task  $\mathfrak{S}_{j,n}$ , given that other bidders' strategies are fixed, any bidder  $i \in \mathbb{I}$  wins the auction with bid  $(\chi_i^j, b_i^j)$  still wins by bidding  $(\chi_i^{j'}, b_i^{j'})$  with  $\chi_i^{j'} > \chi_i^j$  and  $b_i^{j'} < b_i^j$ .

- **Critical payment.** Any winner  $i$  with bid  $(\chi_i^j, b_i^j)$  of task  $\mathfrak{S}_{j,n}$  is paid the critical payment, i.e., the supremum of all bidding prices  $b_i^j$ 's such that  $(\chi_i^j, b_i^j)$  still wins, i.e.,  $p_{i^*,j}^{\text{CP}} = \sup\{b_i^{j'} | \beta_i^{j'} = 1\}$ , when the bids of others remain unchanged.

*Proof:* Let  $(\chi_i^j, \Theta(\chi_i^j))$  be the truthful combinatorial bid of vehicle  $i$ . Obviously, if vehicle  $i \in \mathbb{C}_{j,n}$  loses the auction with untruthful combinatorial bid  $(\chi_i^{j'}, b_i^{j'})$  where  $\chi_i^{j'} \neq \chi_i^j$  or  $b_i^{j'} \neq \Theta(\chi_i^j)$ , its payoff is non-positive. Besides, rational bidders will not receive negative payoffs. Therefore, only the case that bidder  $i$  wins with bid  $(\chi_i^{j'}, b_i^{j'})$  needs to be considered.

First, we prove that for a strategic bidder that bids an untruthful computation resource supply  $\chi_i^{j'} \neq \chi_i^j$  is always dominated by  $\chi_i^j$ . If  $\chi_i^{j'} > \chi_i^j$ , then even if the bidder wins, it has to supply  $\chi_i^{j'}$  amount of computation resources. Thereby, the payoff of the vehicle is non-positive. When the vehicle bids  $\chi_i^{j'} < \chi_i^j$ , according to the monotonicity property, if it wins with bid  $\chi_i^{j'}$ , it could also win with  $\chi_i^j$ . Besides, given the critical payment property, the payment in the latter case (i.e.,  $\chi_i^j$ ) is never lower than that in the former case (i.e.,  $\chi_i^{j'}$ ). Therefore, bidding the truthful computation resource supply is the dominant strategy of any vehicle.

Next, we prove that for a strategic bidder that misreports a bidding price  $b_i^{j'} \neq \Theta(\chi_i^j)$  is always dominated by  $\Theta(\chi_i^j)$ . If vehicle  $i$  wins by bidding  $\Theta(\chi_i^j)$  and it is paid the critical value  $p_{i,j}^{\text{CP}} > \Theta(\chi_i^j)$ , then any possible bidding price  $b_i^{j'} \leq p_{i,j}^{\text{CP}}$  still leads to the winning. Moreover, under the critical payment condition, it is still paid the same amount  $p_{i,j}^{\text{CP}}$ , resulting in the same payoff as when bidding  $\Theta(\chi_i^j)$ . Under this circumstance, a bidding price  $b_i^{j'} > p_{i,j}^{\text{CP}}$  will lose the auction, and the vehicle obtains the zero payoff at last. Therefore, for the vehicle that wins by bidding  $\Theta(\chi_i^j)$ , it is not better than truthfully bidding  $\Theta(\chi_i^j)$ . If vehicle  $i$  loses by bidding  $\Theta(\chi_i^j)$ , then the largest winning bidding price is  $p_{i,j}^{\text{CP}} \leq \Theta(\chi_i^j)$ . In this case, bidding at most  $p_{i,j}^{\text{CP}}$  will win the auction but yields a negative payoff of the vehicle, and bidding more than  $p_{i,j}^{\text{CP}}$  will still lose the auction for the vehicle. Therefore, for the vehicle that loses by bidding  $\Theta(\chi_i^j)$ , truthfully bidding  $\Theta(\chi_i^j)$  is still the dominant strategy.

Thereby, our SRC auction mechanism satisfies CIC if both the monotonicity and critical payment properties hold. ■

#### APPENDIX B PROOF OF LEMMA 1

**Lemma 1.** SEAL satisfies combinatorial incentive compatibility (CIC) for both computation resource supply and bidding price.

*Proof:* It is equivalent to prove the CIC of each sub-auction  $\mathcal{A}_j$  for task  $\mathfrak{S}_{j,n}$ ,  $\forall j \in \mathbb{J}'_n$ . In other words, we need to prove the truthfulness of each bidder's reported computing resource and the bidding price in each sub-auction  $\mathcal{A}_j$ . According to Theorem 1, it suffices to prove the monotonicity of worker selection process in  $\mathcal{A}_j$  and payment  $p_{i,j}^{\text{CP}}$  is the critical value for the bidder  $i$  to win the auction  $\mathcal{A}_j$ . Without loss of generality, suppose that bidder  $i$  wins auction  $\mathcal{A}_j$  with the truthful bid  $(\chi_i^j, \Theta(\chi_i^j))$ .

1) **Monotonicity.** The monotonicity of  $\mathcal{A}_j$  is proved in the following two cases.

Case 1: bidder  $i$  decreases its bidding price, i.e.,  $b_i^{j-} < \Theta(\chi_i^j)$ . The reduction in bidding price decreases the bidder's MCF contribution, i.e.,

$$F(\chi_i^j, b_i^{j-}) < F(\chi_i^j, \Theta(\chi_i^j)). \quad (33)$$

As a result, bidder  $i$  is still the winner.

Case 2: bidder  $i$  increases its reported amount of computing resource, i.e.,  $\chi_i^{j+} > \chi_i^j$ . This also decreases bidder  $i$ 's MCF, i.e.,

$$F(\chi_i^{j+}, \Theta(\chi_i^j)) < F(\chi_i^j, \Theta(\chi_i^j)). \quad (34)$$

Consequently, bidder  $i$  is still the winner.

According to the above two cases, each sub-auction  $\mathcal{A}_j$  is monotone.

2) *Critical payment.* Next, we prove that  $p_{i,j}^{\text{CP}}$  derived by Algorithm 2 exactly equals to the critical value. The proof is divided into two cases.

Case 1: given the fixed  $\chi_i^j$ , bidder  $i$  bids less than or equal to the obtained payment, i.e.,  $b_i^{j-} \leq p_{i,j}^{\text{CP}}$ . Then, bidder  $i$  still wins and receives the same payment, as well as the same payoff.

Case 2: given the fixed  $\chi_i^j$ , bidder  $i$  bids greater than the obtained payment, i.e.,  $b_i^{j+} > p_{i,j}^{\text{CP}}$ . We assume that bidder  $k$  ( $k \neq i$ ) is the new winner (i.e., critical bidder) when bidder  $i$  does not participate in auction  $\mathcal{A}_j$ . In this case, based on Eqs. (27) and (32), we have

$$\begin{aligned} & F(\chi_i^j, b_i^{j+}) \\ & > \varpi s_{j,n} \left( \frac{P^{\text{hov}} \zeta_{j,n}}{\chi_i^j} + \frac{P^{\text{A2G}} + P^{\text{hov}}}{\gamma_i^j} \right) + (1 - \varpi) \lambda_p p_{i,j}^{\text{CP}} \\ & = \varpi s_{j,n} \left( \frac{P^{\text{hov}} \zeta_{j,n}}{\chi_k^j} + \frac{P^{\text{A2G}} + P^{\text{hov}}}{\gamma_k^j} \right) + (1 - \varpi) \lambda_p b_k^j \\ & = F(\chi_k^j, b_k^j). \end{aligned} \quad (35)$$

Therefore, bidder  $k$  wins the auction while bidder  $i$  loses.

Combing the above two cases, when  $\chi_i^j$  is fixed,  $p_{i,j}^{\text{CP}}$  is exactly the critical payment for bidder  $i$  above which bidder  $i$  losses. It can be concluded that bidding truthfully is the dominant strategy of each bidder to maximize its payoff. Lemma 1 is proved. ■

## APPENDIX C

### PROOF OF LEMMA 2

**Lemma 2.** *SEAL satisfies individual rationality (IR).*

*Proof:* As our SRC auction mechanism satisfies CIC, any bidder will be motivated to submit its truthful bid. Besides, the payoff of any bidder that does not participate in the auction is zero. We consider the following two cases.

Case 1: bidder  $i$  wins the auction with the truthful bid  $(\chi_i^j, \Theta(\chi_i^j))$ . In this case, suppose that bidder  $k \in \mathbb{C}_{j,n} \setminus \{i\}$  is the critical bidder, i.e., virtual winner of task  $\mathfrak{S}_{j,n}$ . Similarly, we have  $F(\chi_i^j, p_{i,j}^{\text{CP}}) = F(\chi_k^j, \Theta(\chi_k^j))$ . Since bidder  $k$  will lose if bidder  $i$  joins in the auction, we have

$$F(\chi_i^j, \Theta(\chi_i^j)) \leq F(\chi_k^j, \Theta(\chi_k^j)) = F(\chi_i^j, p_{i,j}^{\text{CP}}). \quad (36)$$

According to Eq. (27), we have  $p_i^{j*} \geq \Theta(\chi_i^j)$ . Thereby,  $\pi(\chi_i^j, b_i^j) = p_{i,j}^{\text{CP}} - \Theta(\chi_i^j) \geq 0$ .

Case 2: bidder  $i$  loses the auction with the truthful bid  $(\chi_i^j, \Theta(\chi_i^j))$ . In this case,  $\pi(\chi_i^j, b_i^j) = 0$ .

Hence, the payoff of any bidder participating in the auction is no less than zero. Lemma 2 is proved. ■

## APPENDIX D PROOF OF THEOREM 3

**Theorem 3.** *SEAL can preserve participants' bidding privacy.*

*Proof:* We prove the bidding privacy protection of SEAL in two successive phases: off-chain auction execution and on-chain fair exchange. In the first phase, note that the private bids involved in the auction execution process are securely processed inside the TEE. Both the UAV and malicious bidders can only observe the encrypted data but no useful bid information from the output of TEE, even if the outside hardware such as I/O or storage is compromised. Therefore, vehicles' private bids  $(\chi_i^j, b_i^j)$  in the auction process can be preserved with the assistance of TEE in the open smart contract systems.

Next, we analyze the bidding privacy in the fair exchange phase (including Commit, On-chain exchange, Claim, and Refund operations). In this phase, only the winners and their critical payments are made public on blockchain ledgers in the auction outcome. On one hand, as the payments to winners are recorded in  $\text{tx}_{\text{commit}}$  in plaintext on transparent blockchain ledgers, adversaries can deduce the private bid information of critical bidders. Nevertheless, since the identities of critical bidders are hidden from the losing bidders in SEAL, their identity privacy can be protected. On the other hand, in our proposed SRC auction, each winner is paid with the critical payment (calculated based on the critical bidder in Eq. (32)) instead of its raw bid, to ensure combinatorial strategy proofness. Given the winners and their critical payments in the public auction outcome, adversaries cannot link winners' true bidding prices with the corresponding critical payments. As such, the true bidding price of each winner can be hidden and preserved in SEAL. Therefore, vehicles' bid privacy can be preserved in the fair exchange phase. ■

## APPENDIX E

### PROOF OF THEOREM 4

**Theorem 4.** *SEAL is a fair auction mechanism, i.e., it guarantees both participation fairness and exchange fairness.*

*Proof:* 1) *Participation fairness.* In the blockchain, as each bidder needs to be registered and make sufficient deposits, any registered bidder who aborts the assigned tasks will be punished by confiscating the deposits. As the deposits are greater than any bidding price, any bidder who aborts can obtain a negative payoff regardless of winning or losing the auction. Besides, for an honest bidder, its expected payoff is always larger than that if it aborts. Therefore, any rational and selfish bidder will honestly follow the auction protocol and has no incentives to abort the auction.

2) *Exchange fairness.* The exchange fairness is guaranteed by the proposed on-chain fair exchange protocol in Algorithm 1. Specifically, after publishing the metadata information  $\{\text{meta}_i\}_{i \in \mathbb{W}}$  on blockchain, the hash values in the hashchain (generated by the UAV) can be utilized as commitments for micropayments to bidders to prevent the UAV from refusing to pay. Besides, after the off-chain delivery of hash values, if the bidder refuses to release task results to the UAV or delivers an incorrect key for decryption, its misbehavior will be immutably recorded in the blockchain and accordingly the smart contract will not conduct the corresponding payment to the bidder from the escrow pool. Under the supervision of smart contracts, the on-chain exchange process of payment and task results can be

executed automatically and atomically between distrustful bidders and the UAV.

Hence, both participation fairness and exchange fairness are ensured in SEAL. ■

#### APPENDIX F PROOF OF THEOREM 5

**Theorem 5.** *The computational complexity and communication complexity of SEAL are  $\mathcal{O}(J_n I \log(I))$  and  $\mathcal{O}(I(M \cdot \text{bit}_\pi + \text{bit}_c))$ , respectively.*

*Proof:* Let  $\text{bit}_p$ ,  $\text{bit}_c$ , and  $\text{bit}_\pi$  be the bit lengths of the plaintext of combinatorial bid, ciphertext and signature (we consider them to be the same by default), and ZKP, respectively. Let  $M$  be the number of consensus nodes in the smart contract system.

In the **init** phase, the main computation is to encrypt the bids, resulting in a complexity of  $\mathcal{O}(I)$ .  $I$  is the total number of vehicles. The submission of ciphertext of bids to the TEE costs  $\mathcal{O}(I \cdot \text{bit}_c)$ . In the **off-chain auction execution** phase, the computation and communication complexities are greatly optimized due to the computing over the plaintext inside the TEE. In this phase, there exist two main operations for winner determination and pricing, i.e., bid sorting and comparison, where the total computational complexity yields  $\mathcal{O}(\sum_{j=1}^{J_n} I_j + J_n |\mathbb{C}_{j,n}| \log(|\mathbb{C}_{j,n}|))$ .  $I_j$  is the number of vehicles involved in task  $\mathfrak{S}_{j,n}$ . After the off-chain auction execution, publishing auction result  $(\vec{\beta}, \vec{p})$  incurs a  $\mathcal{O}(M \cdot \text{bit}_p)$  communication complexity.

In the **commit** phase, the computation overhead mainly due to the generation of hashchains for winners, which yields  $\mathcal{O}(\sum_{i=1}^{|\mathbb{W}|} |\mathbb{G}_i|)$ . The communication only occurs in sending the transaction  $\text{tx}_{\text{commit}}$  to the smart contract by the UAV. In the **on-chain exchange** phase, the computation overhead mainly consists of the encryption/decryption of processed results and symmetric key, as well as the generation and verification of ZKP, which yields  $\mathcal{O}(\sum_{i=1}^{|\mathbb{W}|} |\mathbb{G}_i|)$ . For exchange of task results and payments, each winner delivers the task result message  $\text{ResMsg}$  and decryption key  $k_i^l$  to the UAV and receives the hash values as micropayments from the UAV, which yields  $\mathcal{O}(M |\mathbb{W}| \cdot \text{bit}_\pi)$ . In the **claim** phase, each winner computes its due payment, which incurs a  $\mathcal{O}(\sum_{i=1}^{|\mathbb{W}|} |\mathbb{G}_i|)$  computation overhead as  $N \leq |\mathbb{G}_i|$ . Meanwhile, the communication complexity is  $\mathcal{O}(M |\mathbb{W}|)$ . It is worth mentioning that in the on-chain exchange and claim phases, all winners perform task result/payment exchange and redeem the due payment in parallel. Thereby, the system latency can be further alleviated.

In both **deposit** and **refund** phases, the computation and communication overheads yield  $\mathcal{O}(I)$  and  $\mathcal{O}(IM)$  in financial settlement, respectively.

Therefore, the overall computational complexity of SEAL is  $\mathcal{O}(\sum_{j=1}^{J_n} I_j + J_n |\mathbb{C}_{j,n}| \log(|\mathbb{C}_{j,n}|) + \sum_{i=1}^{|\mathbb{W}|} |\mathbb{G}_i|)$ , which can be simplified as  $\mathcal{O}(J_n I \log(I))$ . And it determines winners and payments in polynomial time. Moreover, the overall communication complexity of SEAL is  $\mathcal{O}(IM \text{bit}_\pi + I \text{bit}_c)$ . ■

#### REFERENCES

[1] Y. Wang, Z. Su, Q. Xu, R. Li, T. H. Luan, and P. Wang, "A secure and intelligent data sharing scheme for UAV-assisted disaster rescue," *IEEE/ACM Transactions on Networking*, pp. 1–17, 2023, doi:10.1109/TNET.2022.3226458.

[2] C. Tang, X. Wei, C. Zhu, Y. Wang, and W. Jia, "Mobile vehicles as fog nodes for latency optimization in smart cities," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 9364–9375, 2020.

[3] Y. Wang, Z. Su, N. Zhang, and R. Li, "Mobile wireless rechargeable UAV networks: Challenges and solutions," *IEEE Communications Magazine*, vol. 60, no. 3, pp. 33–39, 2022.

[4] Z. Ning, Y. Yang, X. Wang, L. Guo, X. Gao, S. Guo, and G. Wang, "Dynamic computation offloading and server deployment for UAV-enabled multi-access edge computing," *IEEE Transactions on Mobile Computing*, vol. 22, no. 5, pp. 2628–2644, 2023.

[5] Y. Wang, W. Chen, T. H. Luan, Z. Su, Q. Xu, R. Li, and N. Chen, "Task offloading for post-disaster rescue in unmanned aerial vehicles networks," *IEEE/ACM Transactions on Networking*, vol. 30, no. 4, pp. 1525–1539, 2022.

[6] S. Sarkar, M. W. Totaro, and K. Elgazzar, "Leveraging the cloud to achieve near real-time processing for drone-generated data," in *IEEE Women in Engineering (WIE) Forum USA East*, 2019, pp. 1–6.

[7] A. Gao, Y. Hu, W. Liang, Y. Lin, L. Li, and X. Li, "A QoE-oriented scheduling scheme for energy-efficient computation offloading in UAV cloud system," *IEEE Access*, vol. 7, pp. 68 656–68 668, 2019.

[8] J. Yao and N. Ansari, "Online task allocation and flying control in fog-aided Internet of drones," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5562–5569, 2020.

[9] M.-A. Messous, S.-M. Senouci, H. Sedjelmaci, and S. Cherkaoui, "A game theory based efficient computation offloading in an UAV network," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4964–4974, 2019.

[10] M. Shin, J. Kim, and M. Levorato, "Auction-based charging scheduling with deep learning framework for multi-drone networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4235–4248, 2019.

[11] H. Yu, K. Meier, M. Argyle, and R. W. Beard, "Cooperative path planning for target tracking in urban environments using unmanned air and ground vehicles," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 2, pp. 541–552, 2015.

[12] J. S. Ng, W. Y. B. Lim, H.-N. Dai, Z. Xiong, J. Huang, D. Niyato, X.-S. Hua, C. Leung, and C. Miao, "Joint auction-coalition formation framework for communication-efficient federated learning in UAV-enabled internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2326–2344, 2021.

[13] F. Wu, T. Zhang, C. Qiao, and G. Chen, "A strategy-proof auction mechanism for adaptive-width channel allocation in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 10, pp. 2678–2689, 2016.

[14] Y. Chen, X. Tian, Q. Wang, J. Jiang, M. Li, and Q. Zhang, "SAFE: A general secure and fair auction framework for wireless markets with privacy preservation," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 3, pp. 2038–2053, 2022.

[15] Q. Huang, Y. Gui, F. Wu, G. Chen, and Q. Zhang, "A general privacy-preserving auction mechanism for secondary spectrum markets," *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1881–1893, 2016.

[16] Y. Chen, X. Tian, Q. Wang, M. Li, M. Du, and Q. Li, "ARMOR: A secure combinatorial auction for heterogeneous spectrum," *IEEE Transactions on Mobile Computing*, vol. 18, no. 10, pp. 2270–2284, 2019.

[17] Q. Wang, J. Huang, Y. Chen, X. Tian, and Q. Zhang, "Privacy-preserving and truthful double auction for heterogeneous spectrum," *IEEE/ACM Transactions on Networking*, vol. 27, no. 2, pp. 848–861, 2019.

[18] Z. Wan, T. Zhang, W. Liu, M. Wang, and L. Zhu, "Decentralized privacy-preserving fair exchange scheme for V2G based on blockchain," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 4, pp. 2442–2456, 2022.

[19] M. Liwang, S. Dai, Z. Gao, Y. Tang, and H. Dai, "A truthful reverse-auction mechanism for computation offloading in cloud-enabled vehicular network," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4214–4227, 2019.

[20] J. Lin, D. Yang, M. Li, J. Xu, and G. Xue, "Frameworks for privacy-preserving mobile crowdsensing incentive mechanisms," *IEEE Transactions on Mobile Computing*, vol. 17, no. 8, pp. 1851–1864, 2018.

[21] Z. Wang, J. Li, J. Hu, J. Ren, Z. Li, and Y. Li, "Towards privacy-preserving incentive for mobile crowdsensing under an untrusted platform," in *IEEE INFOCOM*, 2019, pp. 2053–2061.

[22] H. S. Galal and A. M. Youssef, "Trustee: Full privacy preserving vickrey auction on top of Ethereum," in *Financial Cryptography and Data Security*, 2020, pp. 190–207.

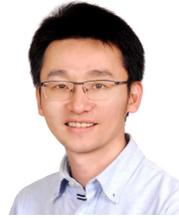
[23] Z. Xiong, S. Feng, W. Wang, D. Niyato, P. Wang, and Z. Han, "Cloud/fog computing resource management and pricing for blockchain networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4585–4600, 2019.

[24] J. S. Ng, W. Y. B. Lim, Z. Xiong, D. Niyato, C. Leung, and C. Miao, "A double auction mechanism for resource allocation in coded vehicular edge computing," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 2, pp. 1832–1845, 2022.

[25] Z. Gao, M. Liwang, S. Hosseinalipour, H. Dai, and X. Wang, "A truthful auction for graph job allocation in vehicular cloud-assisted networks," *IEEE Transactions on Mobile Computing*, vol. 21, no. 10, pp. 3455–3469, 2022.

- [26] D. Callegaro and M. Levorato, "Optimal computation offloading in edge-assisted UAV systems," in *IEEE GLOBECOM*, 2018, pp. 1–6.
- [27] T. Bai, J. Wang, Y. Ren, and L. Hanzo, "Energy-efficient computation offloading for secure UAV-edge-computing systems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 6074–6087, 2019.
- [28] X. Hou, Z. Ren, W. Cheng, C. Chen, and H. Zhang, "Fog based computation offloading for swarm of drones," in *IEEE ICC*, 2019, pp. 1–7.
- [29] M. Liwang, Z. Gao, and X. Wang, "Let's trade in the future! A future-enabled fast resource trading mechanism in edge computing-assisted UAV networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 11, pp. 3252–3270, 2021.
- [30] A. Sacco, F. Esposito, G. Marchetto, and P. Montuschi, "A self-learning strategy for task offloading in UAV networks," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 4, pp. 4301–4311, 2022.
- [31] M. Brandenburger, C. Cachin, R. Kapitza, and A. Sorniotti, "Trusted computing meets blockchain: Rollback attacks and a solution for Hyperledger Fabric," in *IEEE SRDS*, 2019, pp. 324–32409.
- [32] J. Wang, N. Lu, Q. Cheng, L. Zhou, and W. Shi, "A secure spectrum auction scheme without the trusted party based on the smart contract," *Digital Communications and Networks*, vol. 7, no. 2, pp. 223–234, 2021.
- [33] Z. Shi, C. D. Laat, P. Grosso, and Z. Zhao, "When blockchain meets auction models: A survey, some applications, and challenges," *arXiv preprint arXiv:2110.12534*, 2021.
- [34] W. L. Tan, W. C. Lau, O. Yue, and T. H. Hui, "Analytical models and performance evaluation of drive-thru internet systems," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 1, pp. 207–222, 2011.
- [35] J. D. Fricker and R. K. Whitford, "Chapter 2 - traffic flow: Theory and analysis," in *Fundamentals of Transportation Engineering: A Multimodal Systems Approach*. Prentice Hall, 2004.
- [36] M. H. Cheung, F. Hou, V. W. Wong, and J. Huang, "DORA: Dynamic optimal random access for vehicle-to-roadside communications," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 4, pp. 792–803, 2012.
- [37] N. Nisan, "Chapter 9 - algorithmic mechanism design: Through the lens of multi-unit auctions," in *Handbook of Game Theory with Economic Applications*. Elsevier, 2015, vol. 4, pp. 477–515.
- [38] I. Anati, S. Gueron, S. P. Johnson, and V. R. Scarlata, "Innovative technology for CPU based attestation and sealing," in *ACM HASP*, vol. 13, 2013, pp. 1–9.
- [39] Y. Wang, Z. Su, N. Zhang, J. Chen, X. Sun, Z. Ye, and Z. Zhou, "SPDS: A secure and auditable private data sharing scheme for smart grid based on blockchain," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7688–7699, 2021.
- [40] Y. Wang, H. Peng, Z. Su, T. H. Luan, A. Benslimane, and Y. Wu, "A platform-free proof of federated learning consensus mechanism for sustainable blockchains," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 12, pp. 3305–3324, 2022.
- [41] R. L. Rivest and A. Shamir, "Payword and micromint: Two simple micro-payment schemes," in *Security Protocols*, 1997, pp. 69–87.
- [42] H. Cui, Z. Wan, X. Wei, S. Nepal, and X. Yi, "Pay as you decrypt: Decryption outsourcing for functional encryption using blockchain," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3227–3238, 2020.
- [43] Traceset of mobility data of taxi cabs. San Francisco, CA, USA. [Online]. Available: <https://crawdad.org/epfl/mobility/20090224/cab>
- [44] N. Cheng, F. Lyu, W. Quan, C. Zhou, H. He, W. Shi, and X. Shen, "Space/aerial-assisted computing offloading for IoT applications: A learning-based approach," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 5, pp. 1117–1129, 2019.

**Zhou Su** has published technical papers, including top journals and top conferences, such as *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY*, *IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING*, *IEEE TRANSACTIONS ON MOBILE COMPUTING*, *IEEE/ACM TRANSACTIONS ON NETWORKING*, and *INFOCOM*. Dr. Su received the Best Paper Award of International Conference *IEEE ICC2020*, *IEEE BigdataSE2019*, and *IEEE CyberSciTech2017*. He is an Associate Editor of *IEEE INTERNET OF THINGS JOURNAL*, *IEEE OPEN JOURNAL OF THE COMPUTER SOCIETY*, and *IET COMMUNICATIONS*.



**Tom H. Luan** received the Ph.D. degree from the University of Waterloo, Canada, in 2012. He is currently a Professor with Xi'an Jiaotong University, China. He has authored/coauthored more than 97 journal articles and 58 technical articles in conference proceedings. His research mainly focuses on content distribution and media streaming in vehicular ad hoc networks and peer-to-peer networking and the protocol design and performance evaluation of wireless cloud computing and edge computing.



**Jiliang Li** received the Dr. rer. nat. degree in computer science from the University of Göttingen, Göttingen, Germany, in 2019. He is currently a Researcher Professor and PhD Supervisor with the School of Cyber Science and Engineering, Xi'an Jiaotong University, Xi'an, China. His research interests include information security, cryptography, blockchain and IoT security.



**Qichao Xu** received the Ph.D degree from the school of Mechatronic Engineering and Automation, Shanghai University, Shanghai, China, in 2019. He is currently an Associate Professor with Shanghai university. His research interests are in trust and security, the general area of wireless network architecture, Internet of things, vehicular networks, and resource allocation. He has published more than 50 papers in some respected journals, e.g., *IEEE TIFS*, *IEEE TDSC*, *IEEE TWC*, *IEEE TII*, *IEEE TVT*, etc. He was receipt of the best paper awards from several international conferences including *IEEE IWCMC2022*, *IEEE MSN2020*, *EAI MONAMI2020*, *IEEE Comsoc GCCTC2018*, *IEEE CyberSciTech 2017*, and *WiCon2016*.



**Ruidong Li** received the D.Eng. degree from the University of Tsukuba in 2008. He is currently an Associate Professor with the College of Science and Engineering, Kanazawa University, Japan. His research interests include future networks, big data networking, blockchain, and network security. He is the Secretary of *IEEE Com-SoC Internet Technical Committee* and the Founder and Chair of the *IEEE SIG on big data intelligent networking* and *IEEE SIG on intelligent Internet edge*. He is a guest editor of prestigious journals, such as *IEEE COMMUNICATIONS MAGAZINE*, *IEEE NETWORK MAGAZINE*, and *IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING*.



**Yuntao Wang** received the Ph.D degree in Cyberspace Security from Xi'an Jiaotong University, Xi'an, China, in 2022, where he is currently an Assistant Professor with the School of Cyber Science and Engineering. His research interests include security and privacy in intelligent IoT, network games, and blockchain.

and *IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING*.