BlockSC: A Blockchain Empowered Spatial Crowdsourcing Service in Metaverse While Preserving User Location Privacy

Yuan Liu, Yanan Zhang, Shen Su, Lejun Zhang, Xiaojiang Du *Fellow, IEEE*, Mohsen Guizani *Fellow, IEEE*, Zhihong Tian *Senior, IEEE*

***Abstract*—Spatial crowdsourcing (SC) has become a fundamental and emerging technology in Metaverse, facilitating the creation of immersive experiences through location-based services. In these systems, a central SC server leverages SC workers who physically travel to task locations to gather spatiotemporal environment data. However, conventional SC systems face two significant challenges: (1) the SC server, functioning as a centralized authority, can sometimes be unreliable, either due to intentional or unintentional misconduct, and (2) to ensure efficient task assignment and validation, the location privacy of tasks and workers is openly accessible. In this study, we formally define location privacy preserved proof generation and verification problem (LP-PGVP) within an SC task matching scenario, with the aim to the above two challenges. Our proposed solution is a blockchain-based SC system (BlockSC), which provides a decentralized platform for task requesters and workers in the Metaverse context through calling smart contracts. We also introduce a ciphertext-based task matching scheme where task location access is granted only to eligible workers executing a task, benefiting from the design of geographic coordinate transformation and bilinear mapping methodology. To further demonstrate the task matching scheme's operation and impact, we present an easy-to-understand case study. Our evaluation findings confirm that the proposed system effectively maintains location privacy for both SC workers and task requesters, without a considerable sacrifice in task matching efficiency.**

***Index Terms*—Location based service, Metaverse, task assignment, location privacy, blockchain**

I. Introduction

OVER the last decade, and particularly in the COVID-19 post-pandemic, the concept of metaverse has surged in popularity. This spike coincides with the transition of hundreds of millions of people taking their work, education, and leisure activities online. The metaverse is described as a collective virtual environment, which facilitates all activities using augmented reality (AR) and virtual reality (VR) technology [1]. The metaverse gained significant traction with the introduction of Horizon Worlds by Meta (formerly known as Facebook) in 2021. It is now generally recognized as a self-sustaining, extensive spatiotemporal, and 3D immersive virtual shared environment. This space allows avatars and holograms to represent human users, providing them the opportunity to interact, work, and socialize in a smooth and unbroken experience [2].

Location-based crowdsourcing services (LBS), also referred to as spatial crowdsourcing (SC) services, play a critical role in collating VR or AR content [3], [4]. This content serves as a bridge, linking the perceived real world with the digitally generated information, and ultimately connecting the real and virtual worlds in the metaverse. Rapid advancements are observed in many SC platforms catering to the metaverse. These include sharing economy platforms for urban services like Uber, Didi, and TaskRabbit, spatiotemporal data collection services like Waze, OpenStreetMap, and LiveMap, as well as platforms for natural resource conservation like iNaturalist [5]–[9].

In a standard LBS/SC system, three primary entities are involved: task requesters who release tasks, workers who fulfill these tasks, and a centralized server. Task requesters submit their tasks to the central SC server, often specifying a location requirement, such as the need for a photograph from a particular place to enhance the VR environmental content. The server then assigns the task to nearby workers in what is known as the task assignment phase. Subsequently, a worker accepts and completes the task by physically attending the specified location, a step referred to as the task reporting phase. In the above conventional SC system model, there are two challenging issues which are crucial to the success of a SC system [9], [10].

The first issue is how to ensure high reliability of the centralized server. It has been reported that more than half of centralized mobile application platforms, e.g. Evernote and MySpace, ever deliberately reveal users' location information to advertisement servers without the consent of the users [4]. The exposure of user location privacy by a server can cause potentially serious danger since the location data often implies sensitive individual attributes. For example, users'

Y. Liu, S.Su, L. Zhang and Z. Tian are with Cyberspace Institute of Advanced Technology, Guangzhou University, China. Corresponding author: Zhihong Tian (tianzhihong@gzhu.edu.cn)

Y. Zhang is with the Software College, Northeastern University, China.

X. Du is with Department of Electrical and Computer Engineering Stevens Institute of Technology Hoboken, NJ 07030, USA.

G. Mohsen is with Mohamed Bin Zayed University of Artificial Intelligence,Abu Dhabi, United Arab Emirates.

This work is supported by National Key Research and Development Plan Program of China Grant No. 2022YFB3102700, and National Natural Science Foundation of China under Grant No.62172085, No.62172115, No.62172353, No.U20B2046, Guangdong Basic and Applied Basic Research Foundation No.2020A1515010450 and No. 202102020867 Basic Research Program Cofunded by Guangzhou City and Guangzhou University No. 202102010445. Joint Research Fund of Guangzhou and University under Grant No. 202201020380. and Guangdong Higher Education Innovation Group No.2020KCXTD007.

Manuscript received March 31, 2023; revised July 14, 2023; accepted September 10, 2023.

workplaces or home addresses can be identified by temporal and behavior analysis, and the physical identities of users can be accurately tagged with only four spatiotemporal data samples [11]. The core reason behind this phenomenon is that SC tasks are managed by the SC server in a centralized manner without transparency nor traceability. Such location privacy leakage concern is the major barrier of SC workers and task requesters from actively engaging in the SC systems or platforms, resulting in the failure of SC application, e.g., location based metaverse services.

The second issue is how to preserve the location privacy for both tasks and workers during assignment and reporting phases. In a conventional SC system, in order to properly assign SC tasks with potential workers, their locations are unavoidable to be collected by the SC server to calculate their distance, even though the location information are private and sensitive that the workers/requesters are unwilling to be publicly released. To preserve location privacy, three main technologies have been applied, namely differential privacy (DP) [12]–[15], encryption [16], [17] and geo-obfuscation based methods [18]. Since the SC workers are required to execute tasks at specified locations, it is desired that a task's location information is only accessible for the workers who are eligible to fulfill the task meanwhile the workers' location eligibility is verifiable for the SC server. This objective has not been achieved by the existing solutions and we attempt to fill this gap by proposing a blockchain supported location privacy-preservation scheme.

Specifically, to mitigate the above two issues, this paper designs a blockchain based spatial crowdsourcing system framework for LBS in metaverse, which is named by "BlockSC" with the tasks assigned based on the proposed matching scheme protecting the location privacy of tasks and workers. Different from the traditional solutions which protect task location privacy by solving a task assignment problem in a reactive manner, we formalize the blockchain based spatial crowdsourcing service in metaverse as a proof generation and verification problem. In our formalized problem, the task assignment process is replaced by the proof generation process by workers, and the task reporting process is replaced by the proof verification by requesters, and their location privacy cannot be inferred by irrelevant parties in a proactive may. Specifically, in our system model, the blockchain network acts the role of the conventional SC server, and the data exchanged and system process are designed in details. A task requester can submit a task with the encrypted grid coordinates and offsets to the center of the grid at which the task locates, where the grid width is set as the geography range of eligible workers. All the pending tasks are accessible to all the available workers. A worker can locally verify whether he/she is eligible to accept each task by converting its location to a grid after offsetting based on the evaluated task and generating the location proof for the eligible tasks. Among the eligible tasks, the worker can further calculate the distance to the tasks (whose locations are implied by the centers of the converted grids) and accept ones through submitting the corresponding location proof to the blockchain. The worker then goes to the task location and completes the task. Once a task is completed, the worker claims the corresponding rewards together with the task report and its location proof. The blockchain network then verifies the eligibility of the workers through the location proof and record the reward transactions.

The main contributions of this study are summarized as follows.

1) A blockchain based SC system framework for LBS in Metaverse is designed by formalizing a proof generation and verification problem, where the conventional SC server is removed. All the task related operations are exchanged and recorded by the blockchain network in a decentralized manner, including publishing tasks, accepting tasks, location proof verification, and reward distribution.
2) A privacy-preserving task matching scheme is proposed based on geographic coordinate transformation and bilinear mapping methodology, where the location of both tasks and workers are properly protected. A task's location only can be accessed by its eligible workers to measure their distance and generate a valid location proof. A worker's location eligibility can be verified by the blockchain in a public manner based on the proof without knowing their actual locations.
3) The proposed system model is implemented in Hyperledger Fabric framework and a set of smart contracts are designed to realize the proposed task matching scheme. The proposed system is experimentally evaluated and compared with two conventional models, and the results show that our privacy-preservation scheme does not significantly sacrifice the task matching efficiency.

The rest of this article is organized as follows. Section II reviews the existing solutions to the two challenging issues. Section IV introduces the system architecture design and the privacy-preserving task matching scheme. Section V provides a case study to illustrate the main procedures of the proposed task matching scheme demonstrating how the tasks and workers are matched with their location privacy preserved. Section VI presents the analysis against the considered security threats. Section VII shows the experimental setting and results. Finally, Section VIII concludes this study and indicates future research directions.

II. Related Work

Spatial crowdsourcing is also termed by mobile crowdsourcing or crowdsensing, and the literature still lacks a commonly agreed definition [19]. The distinguishable feature of SC from conventional crowdsourcing is the location constraint associating with tasks. The privacy of location is always the research focus of this field and many solutions have been proposed [6], [8], [13], [20], [21]. We summarize the related solutions into three categories: differential privacy (DP), encryption, and geo-obfuscation based solutions.

In the first category, the location information of tasks or workers are mixed with noises so that a single task or worker's location can not be inferred. In [12], differential privacy-based location protection (DPLP) scheme is proposed, where the geography location is spited into noisy three-level grids and

the location privacy is thus protected in grid granularity. In [22], the location of workers is converted to polar coordinates and DP noises are added to the polar radius and the polar angle respectively, improving the utility of the sanitized locations. In [23], a local DP technology is combined with an additive secret sharing scheme so as to iteratively calculate the trie-based statistics of workers' location information. In [24], a double disturbance localized differential privacy (DDLDP) algorithm is proposed to protect the location information of workers. Since noises are involved in the exact location information in the DP based solutions, the travel distance from workers to the tasks are measured with uncertainty, bearing success rate loss in task matching .

In the second category, the location information is encrypted and the task assignment and matching is conducted based on the ciphertext. In [25], an encryption algorithm is proposed to encrypt the positions of tasks and staff and an indexing method is designed for the SC server to assign tasks with the nearest workers without knowing their actual locations. In [26], homophobic encryption is applied to protect the locations of workers and tasks and the task matching is proceed following a wait-and-decide mechanism so as to increase the task numbers assigned to workers. In [27], a key derivation method is proposed based on matrix decomposition and tasks are matched with tasks according multi-keywords in a generalized crowdsourcing framework. In [28], a multi location task allocation scheme is designed based on symmetric homomorphic encryption algorithm and the tasks are assigned considering the worker's future trajectory with the smallest Hausdorff half distance to the task location. In [29], the location of workers and tasks are encoded as prefixed attributes based on the prefix encoding method, and a ciphertext policy attribute-based encryption (CP-ABE) scheme is designed to adjust the distance eligibility of workers. A recent work in [30] adopts additive secret sharing based lightweight cryptography to ensure the location privacy in SC platform. These solutions bear sufficiently high computing cost, which even significantly decrease the task matching performance. The proposed task matching scheme also belongs to this category but with much less computational complexity due to the usage of biliner mapping technology.

In the third category, the obfuscation strategy is studied in [14], where the location of vehicles in SC is divided into grids and a location obfuscation. In [15], a Laplacian distribution mechanism based real position blurring model is proposed, where the relative distance ranking is not changed by the blurred location of workers and tasks. In [14], a location obfuscation strategy is studied to ensure geo-indistinguishability property meanwhile minimize the loss caused by location obfuscation.

With the popularity of the blockchain technology, there are many studies of combining crowdsourcing with Blockchain. In [31], for crowdsensing based federated learning, a blockchain based system is proposed to distribute tasks in a decentralized manner without the trusted service, and the differential privacy technique is used to protect training data and location privacy. In [32], an auction based edge computing system is proposed based on blockchain to decrease the network delay caused by the centralized server and the privacy of bid information is protected by the proposed bid confusion strategy. In a distributed vehicular network, a blockchain based reputation system is designed to preserve the privacy of users including location information in [33]. In the crowd sensing based data collection scenario, a blockchain based spatial crowdsourcing is proposed to ensure trusted execution for the allocated tasks, and the workers are motivated to declare their truthful costs [34]. A blockchain based SC management system is designed where deep reinforcement learning is applied to improving the matching performance and tasks with different privacy classifications are managed by multiple sub-blockchains [35]. A blockchain based secure task query algorithm is designed based on the ciphertext of tasks without sharing the encryption keys [36]. A searchable encryption schemes to achieve secure on-chain task matching authorization is studied [37]. A blockchain empowered additive homomorphic encryption with circle based location verification is proposed for vehicle networks to ensure the confidentiality of task location. Different from the existing studies, we aim to propose a blockchain organized SC system where the location of workers and tasks are protected without significantly sacrificing the task matching performance.

III. Problem Formalization

In a conventional SC scenario, there are three types of participants: the SC-server, workers and SC task requesters, and they are organized in a centralized system architecture. The SC server serves as the center of the system to assign tasks for requesters to workers, formalizing the task assignment problem.

Definition 1 (Task Assignment Problem): Given a SC task s and as set of workers $W = \{w_1,\ w_2, \cdots, w_n\}$, the SC task assignment problem with location privacy preservation, $\mathbb{P}_{LP-TAP}(W, s)$, is to assign the task s to a worker w_i^* such that(1) w_i^* can arrive at the task location l_s to fulfill the task and (2) no other workers can arrive at l_s before than w_i^*.

By removing the SC server from the conventional system, we propose a decentralized system architecture where the the blockchain network replace the role of organizing and driving the system workflow, as shown in Fig. 1. In this case, the core issue is not how to assign tasks with workers, but how to generate a proof for workers to self demonstrate their fulfillment a task and how to verify the proof for requesters. Therefore the workers with their location information should be able to generate a verifiable proof which can be validated by requester and the consensus nodes of the blockchain network.

Definition 2 (Proof Generation and Verification Problem): Let $W = \{w_1,\ w_2, \cdots, w_n\}$ be a set of workers and $S = \{s_1, s_2, \cdots, s_m\}$ be a set of tasks with their location $L^s = \{l_1^s, l_2^s, \cdots, l_m^s\}$. Given their location information $L^w = \{l_1^w, l_2^w, \cdots, l_n^w\}$, the proof generation and verification problem (PGVP), $\mathbb{P}_{PGVP}(W, S, L^w, L^s)$=$\{(s_i, w_j)\}$ such that $Verify_s(Proof_w(l_i^s, l_j^w)) == True$.

Here $Proof_w(\cdot)$ is function to output a proof for the worker w_j who accepts the task s_i according to their location information l_i^s and l_j^w, and $Verify_s(\cdot)$ is a function to output a

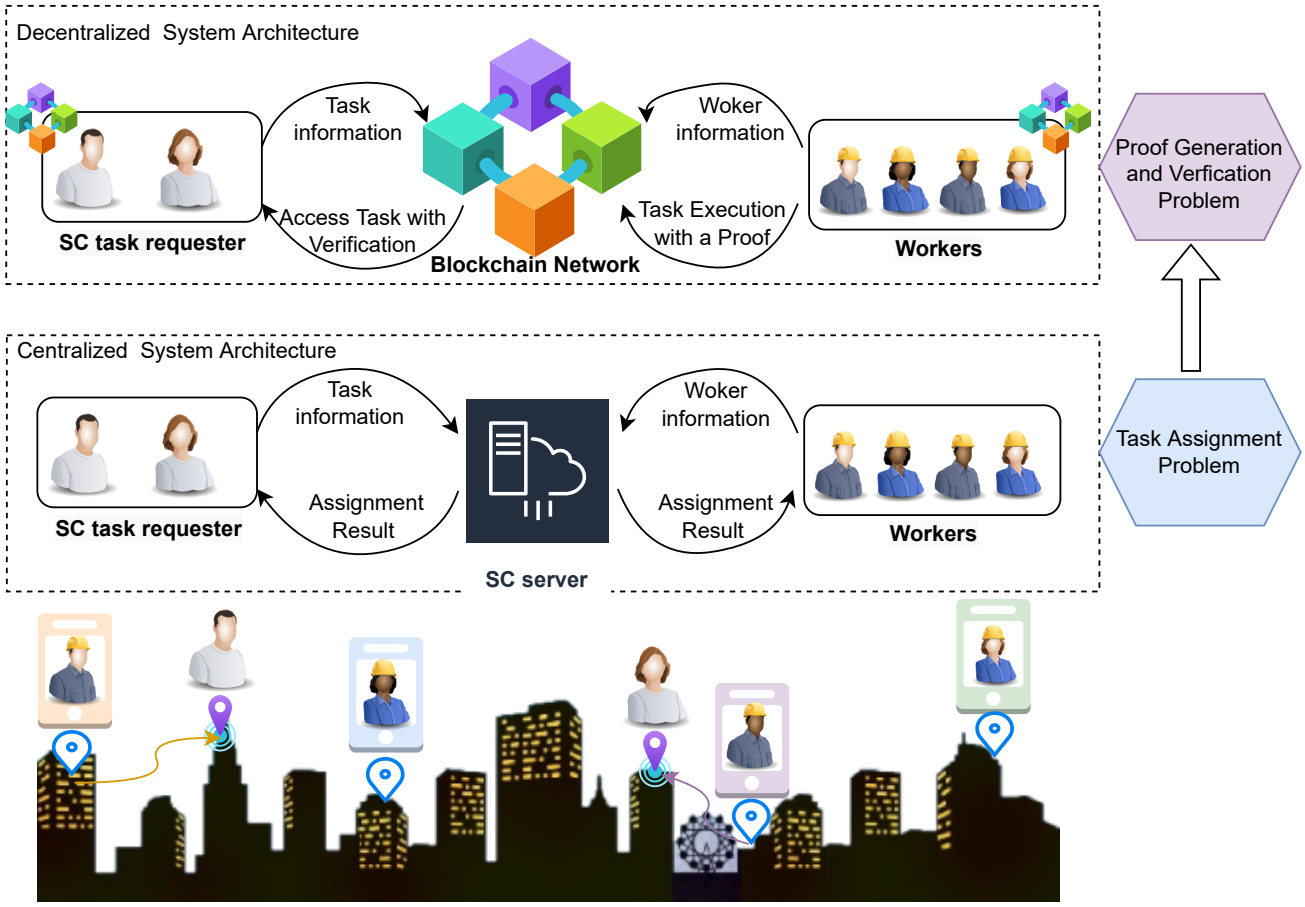

Fig. 1. From task assignment problem to proof generation and verification problem

binary value with $True$ representing for a valid pair of worker and task.

In the above PGVP proble, there are three types of adversary models often happen, which are specified as follows.

1) **Privacy curiosity about task location**: there are malicious entities (requesters/workers/blockchain) who attempt to access the location of all the tasks.
2) **Privacy curiosity about worker location**: there are malicious entities who attempt to access the location of all the workers.
3) **Worker location misreport**: There are dishonest workers who pretend to locate at different locations with their actual positions.

Based on these adversary models, we define location privacy-preserved PGVP.

Definition 3 (Location Privacy-Preserving Proof Generation and Verification Problem, LP-PGVP): A proof generation and verification problem is a LP-PGVP, denoted by $\mathbb{P}_{LP-PGVP}$, if both the task and worker locations cannot be learned by other workers, task requesters, nor the proof validators.

In order to evaluate the performance of a solution to the the LP-PGVP problem, we quantify two metrics: privacy preservation degree (PPD) and task matching efficiency (TME). PPD is the probability of a location being inferred by an attacker based on the public accessable information, and the value range is $[0, 1]$. TME is the probability that the task is able to be completed by a qualified worker, and the value range is also $[0, 1]$. The overall performance evaluation measurement of a solution is denoted by PEM, which is the weighted average of PPD and TME.

$$PEM = \alpha PPD + (1 - \alpha)TME \tag{1}$$

where $\alpha \in [0, 1]$ is the weight of privacy preservation over efficiency, set by system platform or customized by task requesters.

Therefore, the objective of designing a solution to the LP-PGVP problem is to solve the optimal solution of the formalized PEM. It is worthy to note that the truthfulness or quality of a matched task is not considered in the task matching process, which is also out of the scope of the formalized PGVP or LP-PGVP problem. There are several pioneer studies to promote workers in contributing satisfactory performance in [38] and incentive mechanism to recruit high quality workers in [39], and this study is compatible with these studies so as to construct a secure and efficient SC platform.

In the next section, we aim to propose our solution and demonstrate its optimal PEM performance in balencing privacy preservation and efficiency.

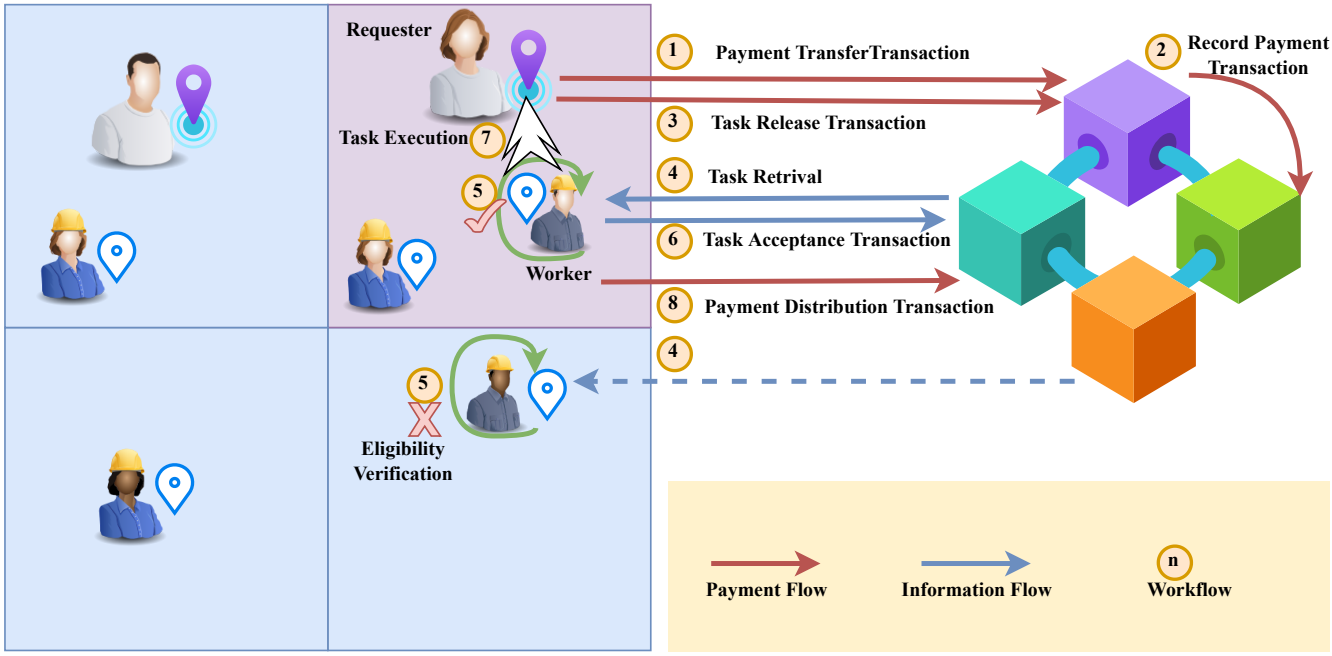

Fig. 2. The Proposed System Framework

IV. The Proposed System Design

A. System Overview

In our system, there are three parties: SC task requester, SC worker and blockchain network, and their relationships are shown in Figure 2. An SC task requester can be performed by an enterprise or an individual, who is willing to recruit a certain number of workers by offering financial rewards to fulfill a task at a specific location (e.g. collecting the images of a landmark building). An SC worker is referred to an individual with a mobile device executing SC tasks, where the workers is willing to complete the task at payment. The blockchain network is composed by consensus nodes who can sustain a distributed data ledger following a consensus and incentive protocol, and smart contracts can been deployed and called so as to execute a pre-determined functional program.

Before a requester publishes a task, the requester should transfer the payment rewards to a public account by submitting a payment transfer transaction to the blockchain (① in Figure 2). The blockchain will record the transaction as long as the account balance of the requester is not less than the transferred value (②). Here the public account can receive task payments from requesters and only the workers who successfully completing the corresponding tasks can redeem the promised rewards. The requester then divide the whole geographic space with grids and the location of the task is denoted by the grid coordinates and the offsets to the grid center. The width of the grid is determined by the task feasibility where only the workers located around the task within the distance of half grid width are capable to complete the task. With the payment transaction being recorded in the blockchain, the requester is eligible to publish a task by submitting a task release transaction consisting of the ciphertext of grid coordinates and plaintext of offsets to the grid center, as well as the task descriptions and the payment transaction pointer (③).

All the unfinished tasks can be retrieved by all the potential workers. Before accepting a task, a worker first verifies whether its location is eligible according to the task matching Algorithm in Section IV-C where the grid coordinates of the workers, after offsetting according to the task, are converted to ciphertext serving as the worker's location proof (④). The eligible workers calculate the distance to the grid center and accept one by submitting a transaction to the blockchain with its location proof (④). The worker then travels to the task location (⑥) and submits a reward distribution transaction from the public account, attached with its completed task report (⑦).

Next, we introduce the detailed design of the proposed system model, including the data structure of the transactions exchanged with the blockchain, the main system procedures.

B. Transaction Types

There are four types of transactions: payment transaction, task release transaction, task acceptance transaction, reward distribution transaction. The main attributes of the four transaction types are summarized in Figure 3 There are three common attributes contained in every transaction:

- transaction_id: a unique index assigned for each transaction, which can be the hash value of the transaction.
- type_id: two bytes, representing for transaction type, with 00 for payment transfer transaction; 01 for task release transaction; 10 for task acceptance transaction; 11 for reward distribution transaction.
- source_id: it indicates the sender of the transaction and the first two types are sent by task requesters and the last two types are sent by workers.

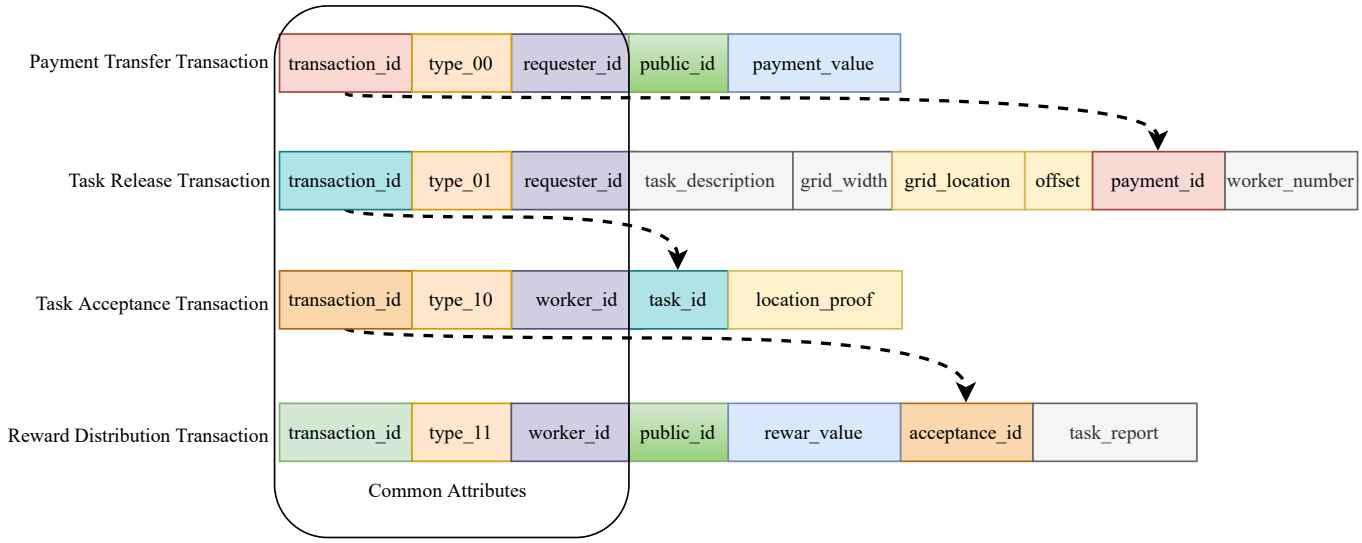

Fig. 3. The Data Structure of Four Types of Transactions

The purpose of a **payment transaction** launched by a task requester is to deposit the task rewards for workers, which is received by *public_id* whose public key and private key are publicly known by all the entities in our system. The last attribute is *payment_value* which is the amount of rewards offered by the task requester. A payment transaction is valid if and only if the requester_id has enough balance to transfer.

A **task release transaction** aims to publish a task, including the task descriptions in *task_description*; the location range in a square with side lenghth of *grid_width*; the encrypted location in *grid_location*; the horizontal and vertical offsets to the center of the located grid in *offset*; the rewards of the task is associated by the transaction_id of the corresponding payment transfer transaction in *payment_id*; the number of workers required is recorded in *worker_number*. A task release transaction is valid if and only if the transaction indexed by payment_id is unspent.

A **task acceptance transaction** is submitted by an eligible worker who decide to accept a task. The accepted task is specified in *task_id* which is the transaction_id of a task release transaction. The *location_proof* is the encrypted grid location of the worker, which is used for the blockchain to verify the location eligibility. A task acceptance transaction is valid if the location_proof can pass the task matching verification for blockchain in Algorithm 1. Once the required number of workers have submitted task acceptance transactions, the blockchain then stop receiving the following task acceptance transactions with the same task_id.

A **reward distribution transaciton** is submitted by a worker who has fulfilled the accepted task. The reward is paid from the *public_id* by the worker with the private key of public_id. The *rewarded_value* is calculated by the task's payment_value dividing worker_number. The *acceptance_id* refers to the task acceptance transaction's transaction_id. The last attribute *task_report* records the results of task executing. A reward distribution transaction is valid if its acceptance_id is valid and the reward_value and task_report are feasible according to the task description.

C. System Procedures

In our system, there are six main procedures: initialization, registration, location transformation, grid location encryption, task justification and verification.

Step 1: Initialization. When the system is initialized, we generate a symmetric bilinear mapping function e and $g \in G$ where G is a $p-$order multiplicative cyclic group. The function e has the bilinear property, i.e., for $\forall g_1 \in G$, $\forall g_2 \in G$, $\forall a \in Z_{p+}$, $\forall b \in Z_{p+}$, we have $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$.

Step 2: Registration. In order to behave in our system, each user should register to a key management server to obtain its id and secret keys. For each user i, the key management server generates a random and unique integer $k_i \in Z_{p+}$ and calculates $s_i = g^{k_i \text{MSK}_s}$ and $r_i = \frac{\text{MSK}_r}{k_i}$ keys based on a pair of master secret keys MSK_s and MSK_r. The id of user i is the hash value of the conjunction of s_i and r_i, which is public. Therefore, after registration, the user i is assigned with triple values [id_i,s_i,r_i].

Step 3: Location Transformation. For a geography range $[X_{min}, X_{max}] \times [Y_{min}, Y_{max}]$, and a grid width d, the whole area can be divided into $\frac{X_{max}-X_{min}}{d} \times \frac{Y_{max}-Y_{min}}{d}$ grids. The position (x_i, y_i) of a user i should be transformed to a grid coordinates $(\hat{x}_i, \hat{y}_i)$ and offsets (δ_i^x, δ_j^y), where

$$\begin{aligned} \hat{x}_i &= \lceil \tfrac{x_i - X_{min}}{d} \rceil \\ \hat{y}_i &= \lceil \tfrac{y_i - Y_{min}}{d} \rceil \\ \delta_i^x &= x_i - (\hat{x}_i d - \tfrac{d}{2}) \\ \delta_j^y &= y_i - (\hat{y}_i d - \tfrac{d}{2}) \end{aligned} \tag{2}$$

where $\lceil \cdot \rceil$ rounds the input up to the nearest integer.

Step 4: Grid Location Encryption. The grid location $\hat{x}_i$ and $\hat{y}_i$ are confidential information for both task requesters and workers. To release a task, the grid location should be

encrypted based on a pair of personally generated secret integers $T_i^x \in Z_{p+}$ and $T_i^y \in Z_{p+}$ as follows.

$$\begin{aligned} Enc(\hat{x}_i) = &\quad (s_i^{T_i^x r_i \hat{x}_i}, g^{T_i^x}) \\ Enc(\hat{y}_i) = &\quad (s_i^{T_i^y r_i \hat{y}_i}, g^{T_i^y}) \end{aligned} \tag{3}$$

where r_i and s_i are publicly known information associated with user i when the user registered in the system; T_i^x and T_i^y are private information generated by id_i locally for encrypting the current location information and the values of T_i^x and T_i^y should be frequently changed by user i according to their security policies. Based on the encrypted grid location, any other task requester or worker cannot access the physical position, thus the location privacy of a user can be preserved.

Step 5: Location Justification and Proof Generation A worker whose location is (x_j, y_j) should justify whether it is an eligible worker in the same grid with the task. The worker then offsets its location according to the evaluated task's δ_i^x and δ_i^y, by adjusting as follows.

$$\begin{aligned} \hat{x}_j = &\quad x_j + \delta_i^x \\ \hat{y}_j = &\quad y_j + \delta_i^y \end{aligned} \tag{4}$$

The adjusted location is transferred and encrypted according to Step 4 and Step 5, respectively, and the worker can obtain $Enc(\hat{x}_j)$ and $Enc(\hat{y}_j)$. Then the worker can execute the location proof verification algorithm in Algorithm 1. When the Algorithm returns true, the worker is eligible for the evaluated task and the distance to the adjusted grid center is the accurate distance to the task, otherwise the worker is not qualified to fulfill the task. The location proof is composed by $Enc(\hat{x}_j)$ and $Enc(\hat{y}_j)$, which is submitted to the blockchain through a task acceptance transaction as a signal to accept the task.

Step 6: Task Execution and Rewards Distribution. After the worker fulfilling the task, the worker submits a reward distribution transaction. The blockchain then evaluates the referred location proof in the associated task acceptance transaction by executing Algorithm 1 in the same manner with that in Step 5.

Algorithm 1 Location Proof and Verification Algorithm

Input: $Enc_i^x(\hat{x}_i)$,$Enc_i^y(\hat{y}_i)$,$Enc_j^x(\hat{x}_j)$, $Enc_j^y(\hat{y}_j)$
Output: MatchingResult;//True or False
1: Calculate $temp_a^x$=$e(Enc_i^x(\hat{x}_i)[1], Enc_j^x(\hat{x}_j)[2])$;
2: Calculate $temp_b^x$=$e(Enc_i^x(\hat{x}_i)[2], Enc_j^x(\hat{x}_j)[1])$;
3: Calculate $temp_a^y$=$e(Enc_i^y(\hat{y}_i)[1], Enc_j^y(\hat{y}_j)[2])$;
4: Calculate $temp_b^y$=$e(Enc_i^y(\hat{y}_i)[2], Enc_j^y(\hat{y}_j)[1])$;
5: **if** $temp_a^x$==$temp_b^x$ && $temp_a^y$==$temp_b^y$ **then**
6: **return** MatchingResult=True;
7: **else**
8: **return** MatchingResult=False;
9: **end if**

Algorithm 1 takes the encrypted task location and the encrypted worker location proof as the inputs and outputs the binary matching result. Two symmetric bilinear mapping functions are conducted for comparing the location with respect to x-axis (Line 1 and 2) and y-axis (Line 3 and 4), respectively. If the conditions in Line 5 are satisfied, then the task and the worker are matched eligible, otherwise they are mismatched.

Proposition 1: Given a task encrypted location and the location proof of a worker, if and only if Algorithm 1 return true, then the work locates in the exact grid centralized by the task.

Proof 1: By adjusting a worker j's location with the offsets of the task i as in Eq.4, the geography grid layout for the worker is the one with the task locating at the center of the task exact grid. We only need to proof whether the worker is in the same grid with the task. In other words, we need to show whether the condition in Line 5 of Algorithm 1 can ensure $\hat{x}_i == \hat{x}_j$ and $\hat{y}_i == \hat{y}_j$.

Based on Eq.(3) and Line 1-2 in Algorithm 1, we can obtain

$$\begin{aligned} temp_a^x = &\quad e(s_i^{T_i^x r_i \hat{x}_i}, g^{T_j^x}) \\ = &\quad e(g^{\text{MSK}_s k_i T_i^x \frac{\text{MSK}_r}{k_i} \hat{x}_i}, g^{T_j^x}) \\ = &\quad e(g,g)^{\text{MSK}_s k_i T_i^x \frac{\text{MSK}_r}{k_i} \hat{x}_j T_i^x} \\ = &\quad e(g,g)^{\text{MSK}_s \text{MSK}_r T_i^x T_j^x \hat{x}_i} \end{aligned} \tag{5}$$

and

$$\begin{aligned} temp_b^x = &\quad e(s_j^{T_j^x r_j \hat{x}_j}, g^{T_i^x}) \\ = &\quad e(g^{\text{MSK}_s k_j T_j^x \frac{\text{MSK}_r}{k_j} \hat{x}_j}, g^{T_i^x}) \\ = &\quad e(g,g)^{\text{MSK}_s k_j T_j^x \frac{\text{MSK}_r}{k_j} \hat{x}_i T_j^x} \\ = &\quad e(g,g)^{\text{MSK}_s \text{MSK}_r T_i^x T_j^x \hat{x}_j} \end{aligned} \tag{6}$$

Comparing Eq.(5) and Eq.(6), we find that if and only if $temp_a^x = temp_b^x$, then $\hat{x}_i = \hat{x}_j$.

Similarly, based on Eq3 and Line 3-4, we can obtain

$$temp_a^y = e(g,g)^{\text{MSK}_s \text{MSK}_r T_i^y T_j^y \hat{y}_i} \tag{7}$$

and

$$temp_b^y = e(g,g)^{\text{MSK}_s \text{MSK}_r T_i^y T_j^y \hat{y}_j} \tag{8}$$

Comparing Eq.(7) and Eq.(8), we can conclude that if and only if $temp_a^y = temp_b^y$, then $\hat{y}_i = \hat{y}_j$.

Therefore, if and only if $temp_a^x = temp_b^x$ and $temp_a^y = temp_b^y$, the worker's adjusted grid (with the task at grid center) is the same with the task.

D. Grid Length Determination

The grid length d is an important parameter in our system, which closely influents the privacy preservation degree and task matching efficiency metrics, thus influents the overal performance evaluation measurement. The setting of d should be able to maximize PEM.

Privacy Preservation Degree, PPD: Let r is the radius of the earth, $m = 2 \times r \times \arcsin\sqrt{\sin^2 \frac{X_{max} - X_{min}}{2}}$ and $n = 2 \times r \times \arcsin\sqrt{\sin^2 \frac{Y_{max} - Y_{min}}{2}}$, the totoal number of grids can be denoted by N where $N = \lceil \frac{m}{d} \rceil \times \lceil \frac{n}{d} \rceil$ where $\lceil \cdot \rceil$ is a function to round up the result to an integer. For reason of simplicity, we directly express the relation between N and d as $N = \frac{m \times n}{d^2}$. Therefore the PPD is calculated as

$$PPD(d) = 1 - \frac{1}{N} = 1 - \frac{d^2}{m \times n} \tag{9}$$

. Let the total number of workers is W, and the density of workers in each grid is assumed to be consistent. The expected number of workers in a grid is $\frac{W}{N}$.

Task Matching Efficiency, TME: We then quantify the TME metric. We do not consider the capacity differences between workers and assume that each worker can successfully complete a task with probability $\theta \in [0,1]$. Thus the probability of task to be successfully executed is calculated as

$$TME(d) = 1-(1-\theta)^{\frac{W}{N}} = 1-(1-\theta)^{\frac{W\times d^2}{m\times n}} \tag{10}$$

Therefore, we can obtain

$$PEM(d) = \alpha(1-\frac{d^2}{m\times n}) + (1-\alpha)\left(1-(1-\theta)^{\frac{W\times d^2}{m\times n}}\right). \tag{11}$$

The optimal value of d is determined by solving the following optimization problem:

$$\begin{aligned} & \max_d PEM(d) \\ \textbf{s.t.} \quad & d \in (0, \max\{m,n\}] \\ & PPD(d) \geq PPD_{min} \\ & TME(d) \geq TME_{min} \end{aligned} \tag{12}$$

where PPD_{min} is the minimal privacy preservation degree and TME_{min} is the minimal task matching efficiency set according to the platform designer or customer requirements.

In an ideal case where $PPD_{min}=0$ and $TME_{min}=0$, we can calculate the first order derivation of $PEM(d)$:

$$\begin{aligned} PEM'(d) \quad &= -\frac{2\alpha\times d}{m\times n} \\ &-\frac{2(1-\alpha)\times W\times d}{m\times n}(1-\theta)^{\frac{W\times d^2}{m\times n}} \times ln(1-\theta) \end{aligned} \tag{13}$$

Let$d^* = \underset{d}{arg}\{PEM'(d)=0\}$, we can obtain that

$$d^* = \sqrt{\frac{m\times n\times ln(-\frac{\alpha}{(1-\alpha\times W\times ln(1-\theta)})}{W\times ln(1-\theta)}} \tag{14}$$

When $d\in(0,d^*]$, $PEM'(d)>0$ where $PEM(d)$ monotonically increases with d, and when $d\in[d^*,\max\{m,n\}]$, $PEM'(d)<0$ where $PEM(d)$ monotonically decreases with d. Therefore, when $d=d^*$, $PEM(d)$ reaches its maximum value.

V. A Numerate Case of the Proposed Privacy Preservation

To explain the proposed task matching algorithm, we provide a numerical case in this section as shown in Figure 4 The task locates at A(2.1,2.1) with grid width 1 and the whole

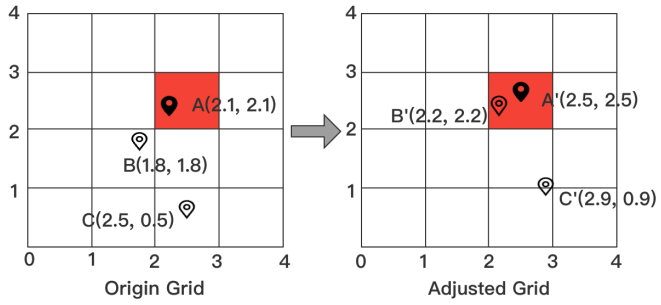

Fig. 4. The Demonstration Case Study

range is $[0,4]\times[0,4]$. A worker is at B(1.8,1.8) and another worker is at C(2.5,0.5). After location transformation, the grid of the task is (3,3) with offset (0.4,0.4).

For the two workers, they are going to evaluate their eligibility for the task. The worker at B adjusts its coordinates from B(1.8,1.8) to B'(2.2,2.2) based on the task's offset. The grid of the worker adjusted at B' is (3,3). Similarly, the grid of the second worker adjusted is (3,1). Next, we demonstrate how the task is matched with the two workers following the proposed system without exposing their location privacy.

We implement the proposed task matching algorithm based on Java's JPBC 2.0.0 library. The symmetric bilinear mapping function e is $y^2=x^3+x$ and g is set as a point in the eclipse curve [6216...1758,6575...5461]. For the purpose of friendly presentation, we only list the first and the last four numbers when the length is greater than 8 numbers. The full version of the example is available at https://github.com/Winter1997/BlockSC.git.

The key generation parameters are set as $\text{MSK}_s = 2$ MSK_r=4, and for the users at A B C the randomly generated key are k_A=2, k_B=4, k_C=6, which are unknown to workers nor task requesters.

The task requester at A encrypts its grid coordinates (3,3) with privately generated T_A^x=3441...8725 and T_A^y=5330...6937, and the encrypted grid location is calculated as

$$\begin{aligned} Enc_A^x = \quad & ([7280\ldots0855, 7658\ldots2310] \\ & ,[7478\ldots6533, 6151\ldots8019]) \\ Enc_A^y = \quad & ([7193\ldots9142, 2613\ldots6401] \\ & ,[9751\ldots4057, 4710\ldots7701]) \end{aligned}$$

The worker at B encrypts its adjusted grid coordinates (3,3) with privately generated T_B^x= 2612...9326 and W_B^y= 8768...7123, and the generated location proof is

$$\begin{aligned} Enc_B^x = \quad & ([1512\ldots7725, 7293\ldots4889] \\ & ,[6403\ldots8698, 9758\ldots4816]) \\ Enc_B^y = \quad & ([2470\ldots0621, 9125\ldots0913] \\ & ,[2301\ldots9971, 9359\ldots4238]) \end{aligned}$$

The worker at position C encrypts its adjusted grid coordinates (3,1) with privately generated T_C^x= 4215...8603 and W_C^y= 1901...6114, and the generated location proof is

$$\begin{aligned} Enc_C^x = \quad & ([4948\ldots4822, 2818\ldots5381] \\ & ,[4243\ldots1030, 2230\ldots7468]) \\ Enc_C^y = \quad & ([6249\ldots9588, 6650\ldots1436] \\ & ,[1003\ldots9786, 5040\ldots5204]) \end{aligned}$$

First, we match the task at A with the worker at B by calculating

$$\begin{aligned} e(Enc_A^x[1], Enc_B^x[2]) &= [2251\ldots4530, 9152\ldots7500] \\ e(Enc_A^x[2], Enc_B^x[1]) &= [2251\ldots4530, 9152\ldots7500] \\ e(Enc_A^y[1], Enc_B^y[2]) &= [8162\ldots3753, 6283\ldots7111] \\ e(Enc_A^y[2], Enc_B^y[1]) &= [8162\ldots3753, 6283\ldots7111] \end{aligned}$$

Therefore, the task at A and the worker at B are matched successfully, by generating a verifiable location proof.

Next, we match the task at A with the worker at C by calculating

$$e(Enc_A^x[1], Enc_C^x[2]) = [2274 \ldots 2961, 4651 \ldots 4429]$$
$$e(Enc_A^x[2], Enc_C^x[1]) = [2274 \ldots 2961, 4651 \ldots 4429]$$
$$e(Enc_A^y[1], Enc_C^y[2]) = [5838 \ldots 3122, 3600 \ldots 9048]$$
$$e(Enc_A^y[2], Enc_C^y[1]) = [9053 \ldots 2289, 4203 \ldots 9627]$$

Therefore, the task at A and the worker at C are not matched.

VI. Security Analysis

In this section, the proposed BlockSC is analyzed against the formalized threat models, i.e., the adversaries are privacy curious about task location, privacy curious about worker location, and misreporting worker locations.

A. Privacy Protection of Task Location

The task's position is determined by an encoded grid location and its deviations from the center of the grid. An interested adversary may only hypothesize a grid. Inquisitive parties wanting to know the locations of all tasks have access to the encoded grid position and unencrypted offsets. Despite the fact that the grid location is made up of whole numbers, it's impossible to enumerate the encrypted position because the privately created variables T^x and T^y occupy a significantly large space. An entity, without knowledge of T^x and T^y, would find it computationally challenging to deduce the grid data based solely on the encrypted grid location. Without the grid location, the plain text offsets do not reveal the task's actual location. Consequently, the location privacy of the task is safeguarded.

B. Privacy Protection of Worker Location

In this scheme, every worker sends their location proof to the blockchain, where the location proof is the encrypted grid coordinates of a worker adjusted relative to a specific task. Much like the task location, an entity is unable to determine a worker's grid data based on the location proof recorded by the blockchain. Even the requester of an accepted task can only confirm the compatibility result, inferring that the worker is qualified without knowing the worker's exact location. Simultaneously, a worker doesn't disclose any location-related data when assessing eligibility during task matching.

C. Against Worker Location Misreport

For dishonest workers who falsely claim to be at a different location, there are two possible scenarios. In the first scenario, the falsely claimed location successfully matches with the task. Since the real location isn't in the same grid, the worker would need to travel a greater distance to complete the task, making it more expensive than choosing a task in their actual grid. As a result, a rational worker has no incentive to falsely report their grid location to achieve a successful match. In the second scenario, where the misreported location doesn't match with the task, the worker cannot gain any profit from this misreporting behavior. Hence, the proposed system is well-equipped to handle false reporting of worker locations.

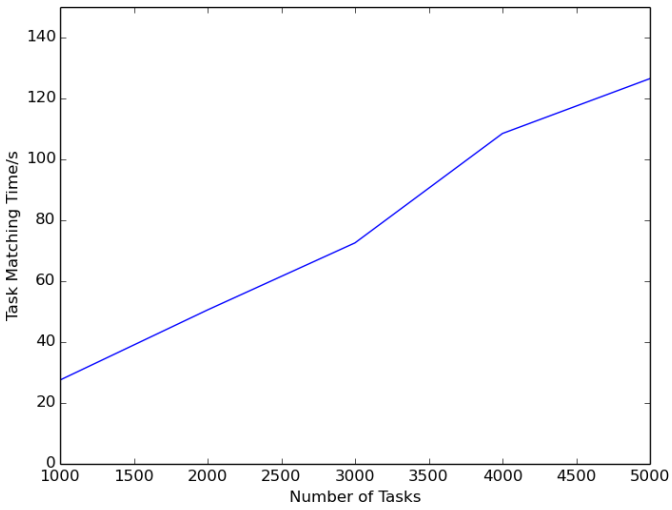

Fig. 5. Task Matching based Verification Time with Different Number of Workers

VII. Experimental Evaluations

In this section, we first evaluate the impact of different parameter settings over the privacy preservation and task matching efficiency by conducting a set of real data based experiments. After that we compare the proposed task matching algorithm with two typical encryption based solutions in [40] and [41] from two perspectives: computation complexity analysis and real data based task matching performance.

To evaluate our system in a practical scenario, we use a real spatial dataset Gowalla (https://github.com/Winter1997/BlockSC.git) in our experiments. We randomly select 5100 records located in the range 73° ∼135° and 4°∼54°, among which 5000 are treated as worker positions, and 100 are treated as task positions. The experimental environment is set in Ubuntu 16.04 operating system, with hardware setting as CPU i7-9700 3.00GHz, RAM 16G. The blockchain network is implemented based on Hyperledger Fabric2.0—a consortium blockchain framework(https://github.com/hyperledger/fabric). The smart contracts and the whole system are developed using Java programming language.

A. Parameter Evaluations

The number of tasks and the width of grid division are two important parameters in the proposed system, we conduct a set of experiments by varying the values of the two parameters.

We set different task numbers ranging from 1000 to 5000 matched with a single worker, and the task matching based justification time (at Step 5) and task matching based verification time (at Step 6) are presented in Figure 5 and Figure 6 respectively. Figure 5 shows that the task matching based justification time linearly increases with the number of tasks. As the task number increase from 50 to 300, the justification time increases from 100ms to 550ms, with each task to be evaluated with 1.8ms on average. It indicates that our matching based justification at Step 5 has a linear scalability with respect to the task number. Figure 6 shows the time of task

TABLE I
COMPUTATION COMPLEXITY OF OUR MODEL COMPARED WITH TWO TYPICAL SOLUTIONS

Operation	Shu [40]	Zhou [41]	BlockSC
E	$4(h_x+h_y)+5(m_x+m_y)$	$2(h_x+h_y)+2(m_x h_x+m_y h_y)$	8
e	$2(m_x h_x+m_y h_y)$	$2(m_x h_x+m_y h_y)$	4
f_s	$h_x+h_y+m_x+m_y$	0	0
H	$h_x+h_y+m_x h_x+m_y h_y$	$h_x+h_y+m_x h_x+m_y h_y$	0

*E: exponentiation computation on group G; e: bilinear mapping operation on group G; f_s: key-based hash operation; H: hash operation; h_x: h -eight of task abscissa index tree; h_y: height of task ordinate index tree;m_x: height of worker abscissa index tree;m_y: height of worker abscissa index tree.

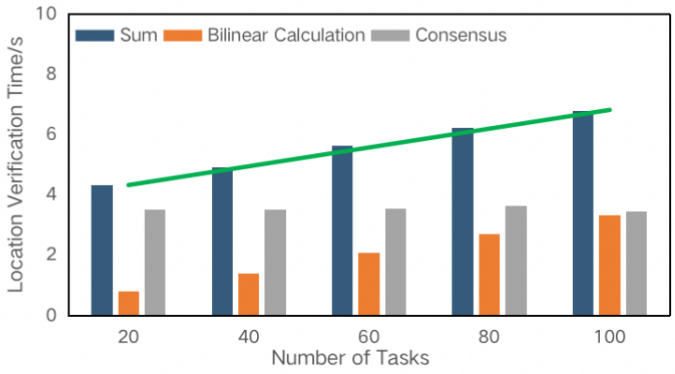

Fig. 6. Task Matching based Justification Time with Different Number of Tasks

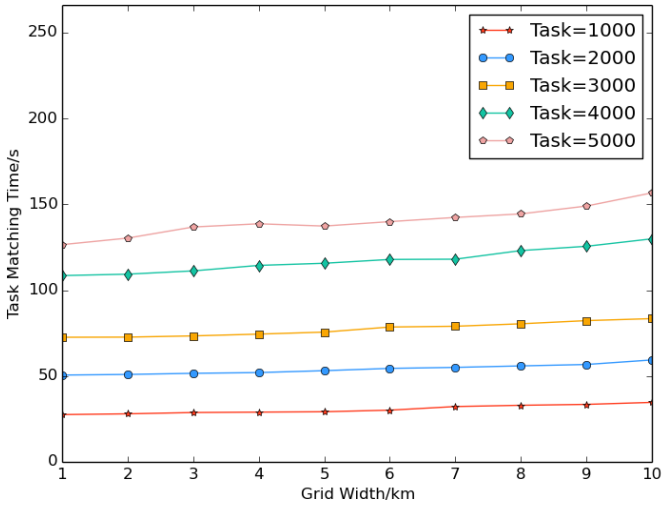

Fig. 7. Task Matching based Justification Time with Different Grid Width Settings

verification. Blue represents the total running time, yellow represents the time occupied by bilinear calculation, and gray represents the total block time. It can be seen from the figure that the time of verification is positively correlated with the number of tasks. When the number of tasks is 100, the total verification time is close to 7 seconds. However, in the real scenario, the number of tasks accepted by a worker is far less than 100, and the real-time requirement of task verification is not high. Therefore, this time is within the acceptable range.

Since the task matching based verification time is nearly independent with task numbers, we only evaluate the task matching based justification time performance when studying the impacts of grid width parameter. We set task numbers being 1000, 2000, 3000, 4000, 5000, and present the justification time in Figure 7 with grid width increasing from 1km to 10km. We can observe that for the same task number, the matching time increases with task numbers which is consistent with the results in Figure 5. When the grid width becomes larger, the matching time also increases. The reason behind this phenomenon is that the number of tasks in a single task become greater when the grid width is larger, and the number of tasks being evaluated also increases.

B. Computation Complexity Comparison

Given a pair of task and worker, we statistically analyze the number of common time-consuming operations of the proposed task matching model, Shu [40] and Zhou [41]. The comparison results are shown in Table I.

In one time task matching process, our model includes task location encryption with four exponential operations, worker location proof generation with four exponential operations (Eq.3), and four bilinear mapping operations for calculating the matching result(Algorithm 1 Line 1 to 4). Compared with [41], our model consumes less number of operations for every operation type with $8 \leq 2(h_x+h_y)+2(m_x h_x+m_y h_y)$ and $4 \leq 2(m_x h_x+m_y h_y)$. Considering the fact that [41] is more computational efficient than [40] as claimed in [41], our model bears the least computation cost among the three solutions.

C. Matching Performance Comparison

In this section, we numerically compare the task matching performance of the proposed model with Shu [40] and Zhou [41]. The experimental settings are the same with those in Sec VII-A, we match a worker with tasks in different numbers and present the task matching time in Figure 8. In Figure 8, we can observe that our model achieves the shortest task matching time, while Zhou's and Shu's almost take the same time. This is because our system only needs to determine whether the longitude and latitude grid coordinates of worker and task are matched, while in the system of Zhou or Shu, each task is compared with every value in the routine of the tree arriving a task, proportional to the height of the tree structure and each worker is compared in the same manner. Thus, the number of matching calculation of Zhou and Shu is the sum number of the tree heights of the positions of the evaluated task and worker.

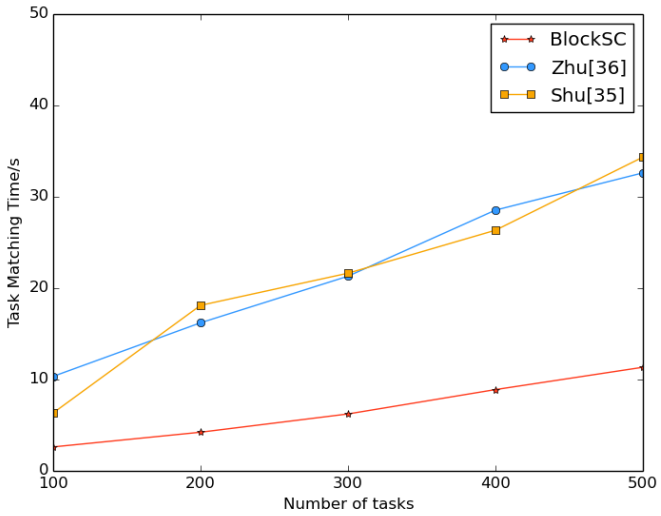

Fig. 8. Matching Time of Three Solutions with Different Number of Tasks

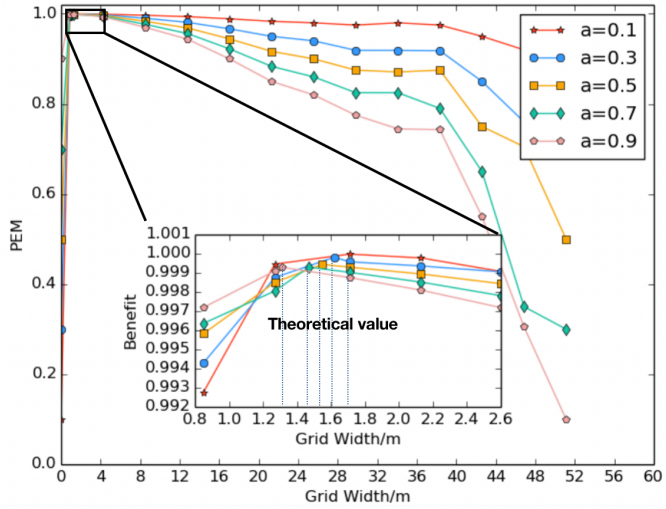

Fig. 9. Performance Measurement with α Adjustment when $\theta = 0.8$

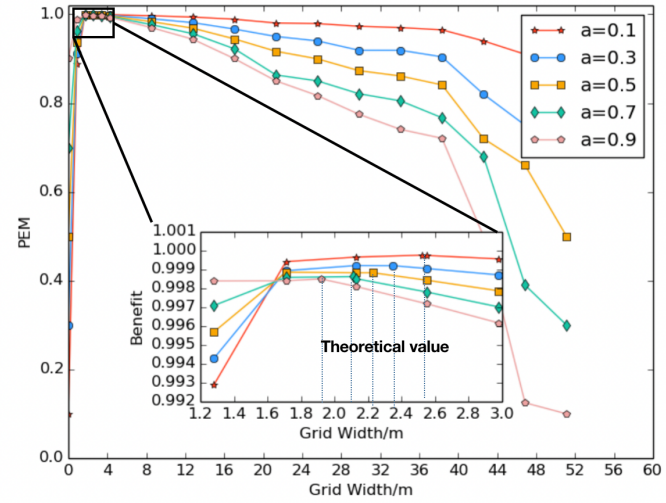

Fig. 10. Performance Measurement with α Adjustment when $\theta = 0.5$

D. Performance Measurement

To validate the efficiency of the grid length determination design, We choose a subset dataset in the latitude range [30.9175,31.4126] and the longitude range [121.2750 121.8661]. We set the task success execution probability θ to be 0.8. By adjusting the weight value $\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, we calculate the PPD, TME, PEM in different settings of grid width and present the results in Figure 9 and Figure 10 while $\theta = 0.8$ and $\theta = 0.5$. As grid width increases, the PEM value increases fastly and then decreases slowly, demonstrating a maximal value around between 1.2 and 3.0. We observe that the optimal grid width is exactly consistent with the theoretically values based on Eq. (14) (2.13,1.62,1.55,1.47,1.31, respectively when $\theta = 0.8$, and 2.52,2.35,2.23,2.11,1.92 when $\theta = 0.5$).

VIII. Conclusion

In this study, we have proposed a blockchain based spatial crowdsourcing system, BlockSC to mitigate two challenging issues: server reliability and privacy preservation. Different from most of the existing studies, we formalize a proof generation and verification problem (PGVP), and location privacy-preserving LP-PGVP based on the conventional task matching problem, where the blockchain network acts the role of the conventional centralized server. We then propose a solution to the formalized LP-PGVP by designing a new SC system framework BlockSC where the location privacy of both workers and tasks are properly protected without significantly scarifying task matching efficiency. More specifically, in the proposed system, we have designed four types of transactions (payment transfer transaction, task release transaction, task acceptance transaction, and reward distribution transaction) and six main procedures (initialization, registration, location transformation, grid location encryption, task matching based justification, and task matching based verification). The proposed task matching algorithm can justify or verify whether a worker with the designed location proof and a task with encrypted grid location are eligible based on the bilinear mapping technology. The privacy preservation degree and task matching efficiency is balanced by setting the grid width to an analytically calculated optimal value. Real data based experiments examine the nearly linear scalability of the proposed solution with task number, and the comparison results with two traditional solutions show the significantly decreased computation complexity and time efficiency.

In future work, we plan to conduct more experimental evaluations based real date sets in various metaverse application scenarios so as to find the practical vulnerabilities and investigate potential improvements. After that, we will test the applicability of our model in a real application, such as Uber or DiDi, with the expectation of contributing our best to the crowd sensing field and constructing a sustainable metaverse ecosystem.

References

[1] M. Damar, "Metaverse shape of your life for future: A bibliometric snapshot," *Journal of Metaverse*, vol. 1, no. 1, pp. 1–8, 2021.

[2] Y. K. Dwivedi, L. Hughes, A. M. Baabdullah, S. Ribeiro-Navarrete, M. Giannakis, M. M. Al-Debei, D. Dennehy, B. Metri, D. Buhalis, C. M. Cheung, K. Conboy, R. Doyle, R. Dubey, V. Dutot, R. Felix, D. Goyal, A. Gustafsson, C. Hinsch, I. Jebabli, M. Janssen, Y.-G. Kim, J. Kim, S. Koos, D. Kreps, N. Kshetri, V. Kumar, K.-B. Ooi, S. Papagiannidis, I. O. Pappas, A. Polyviou, S.-M. Park, N. Pandey, M. M. Queiroz, R. Raman, P. A. Rauschnabel, A. Shirish, M. Sigala, K. Spanaki, G. Wei-Han Tan, M. K. Tiwari, G. Viglia, and S. F. Wamba, "Metaverse beyond the hype: Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy," *International Journal of Information Management*, vol. 66, pp. 102 542:1–55, 2022.

[3] Y. Wang, Z. Su, N. Zhang, R. Xing, D. Liu, T. H. Luan, and X. Shen, "A survey on metaverse: Fundamentals, security, and privacy," *IEEE Communications Surveys & Tutorials*, 2022.

[4] H. Jiang, J. Li, P. Zhao, F. Zeng, Z. Xiao, and A. Iyengar, "Location privacy-preserving mechanisms in location-based services: A comprehensive survey," *ACM Computing Surveys*, vol. 54, no. 1, 2022.

[5] L. Kazemi and C. Shahabi, "Geocrowd: enabling query answering with spatial crowdsourcing," in *Proceedings of the 20th international conference on advances in geographic information systems*, 2012, pp. 189–198.

[6] W. Feng, Z. Yan, H. Zhang, K. Zeng, Y. Xiao, and Y. T. Hou, "A survey on security, privacy, and trust in mobile crowdsourcing," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2971–2992, 2017.

[7] A. Sarı, A. Tosun, and G. I. Alptekin, "A systematic literature review on crowdsourcing in software engineering," *Journal of Systems and Software*, vol. 153, pp. 200–219, 2019.

[8] S. R. B. Gummidi, X. Xie, and T. B. Pedersen, "A survey of spatial crowdsourcing," *ACM Transactions on Database Systems (TODS)*, vol. 44, no. 2, pp. 1–46, 2019.

[9] Y. Tong, Z. Zhou, Y. Zeng, L. Chen, and C. Shahabi, "Spatial crowdsourcing: a survey," *the VLDB Journal*, vol. 29, no. 1, pp. 217–250, 2020.

[10] J. Phuttharak and S. W. Loke, "A review of mobile crowdsourcing architectures and challenges: Toward crowd-empowered internet-of-things," *Ieee access*, vol. 7, pp. 304–324, 2018.

[11] H. Wang, C. Gao, Y. Li, G. Wang, D. Jin, and J. Sun, "De-anonymization of mobility trajectories: Dissecting the gaps between theory and practice," in *25th Annual Network and Distributed System Security Symposium NDSS*, 2018.

[12] J. Wei, Y. Lin, X. Yao, and J. Zhang, "Differential privacy-based location protection in spatial crowdsourcing," *IEEE Transactions of Service Computing*, vol. 15, no. 1, pp. 45–58, 2022.

[13] L. Zheng, L. Chen, and P. Cheng, "Privacy-preserving worker allocation in crowdsourcing," *VLDB Journal*, vol. 31, no. 4, pp. 733–751, 2022.

[14] C. Qiu, A. C. Squicciarini, C. Pang, N. Wang, and B. Wu, "Location privacy protection in vehicle-based spatial crowdsourcing via geo-indistinguishability," *IEEE Trans. Mob. Comput.*, vol. 21, no. 7, pp. 2436–2450, 2022.

[15] H. Wang, E. Wang, Y. Yang, J. Wu, and F. Dressler, "Privacy-preserving online task assignment in spatial crowdsourcing: A graph-based approach," in *IEEE Conference on Computer Communications (INFOCOM)*, 2022, pp. 570–579.

[16] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K. Tan, "Private queries in location based services: anonymizers are not necessary," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, J. T. Wang, Ed., 2008, pp. 121–132.

[17] H. To and C. Shahabi, "Location privacy in spatial crowdsourcing," in *Handbook of Mobile Data Privacy*. Springer, 2018, pp. 167–194.

[18] C. Qiu, A. C. Squicciarini, Z. Li, C. Pang, and L. Yan, "Time-efficient geo-obfuscation to protect worker location privacy over road networks in spatial crowdsourcing," in *The 29th ACM International Conference on Information and Knowledge Management (CIKM)*, 2020, pp. 1275–1284.

[19] E. Estellés-Arolas and F. González-Ladrón-de Guevara, "Towards an integrated crowdsourcing definition," *Journal of Information science*, vol. 38, no. 2, pp. 189–200, 2012.

[20] Y. Ma, Y. Sun, Y. Lei, N. Qin, and J. Lu, "A survey of blockchain technology on security, privacy, and trust in crowdsourcing services," *World Wide Web*, vol. 23, no. 1, pp. 393–419, 2020.

[21] Z. Tian, Y. Wang, Y. Sun, and J. Qiu, "Location privacy challenges in mobile edge computing: Classification and exploration," *IEEE Network*, vol. 34, no. 2, pp. 52–56, 2020. [Online]. Available: https://doi.org/10.1109/MNET.001.1900139

[22] F. Song and T. Ma, "A location privacy protection method in spatial crowdsourcing," *J. Inf. Secur. Appl.*, vol. 65, p. 103095, 2022.

[23] M. Yang, I. Tjuawinata, K. Lam, J. Zhao, and L. Sun, "Secure hot path crowdsourcing with local differential privacy under fog computing architecture," *IEEE Trans. Serv. Comput.*, vol. 15, no. 4, pp. 2188–2201, 2022.

[24] Z. Sun, Y. Wang, Z. Cai, T. Liu, X. Tong, and N. Jiang, "A two-stage privacy protection mechanism based on blockchain in mobile crowdsourcing," *Int. J. Intell. Syst.*, vol. 36, no. 5, pp. 2058–2080, 2021.

[25] H. Li, Q. Song, G. Li, Q. Li, and R. Wang, "GPSC: A grid-based privacy-reserving framework for online spatial crowdsourcing," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 11, pp. 5378–5390, 2022.

[26] M. Li, J. Wu, W. Wang, and J. Zhang, "Toward privacy-preserving task assignment for fully distributed spatial crowdsourcing," *IEEE Internet Things J.*, vol. 8, no. 18, pp. 13 991–14 002, 2021.

[27] J. Shu, X. Jia, K. Yang, and H. Wang, "Privacy-preserving task recommendation services for crowdsourcing," *IEEE Trans. Serv. Comput.*, vol. 14, no. 1, pp. 235–247, 2021.

[28] Y. Guan, P. Xiong, and R. Lu, "Privacy-preserving fog-based multi-location task allocation in mobile crowdsourcing," in *IEEE Global Communications Conference, GLOBECOM*, 2021, pp. 1–6.

[29] W. Huang, X. Lei, and H. Huang, "PTA-SC: privacy-preserving task allocation for spatial crowdsourcing," in *IEEE Wireless Communications and Networking Conference, WCNC*. IEEE, 2021, pp. 1–7.

[30] M. Zhou, Y. Zheng, S. Wang, Z. Hua, H. Huang, Y. Gao, and X. Jia, "PPTA: A location privacy-preserving and flexible task assignment service for spatial crowdsourcing," *Comput. Networks*, vol. 224, p. 109600, 2023. [Online]. Available: https://doi.org/10.1016/j.comnet.2023.109600

[31] W. Wang, Y. Wang, Y. Huang, C. Mu, Z. Sun, X. Tong, and Z. Cai, "Privacy protection federated learning system based on blockchain and edge computing in mobile crowdsourcing," *Comput. Networks*, vol. 215, p. 109206, 2022.

[32] Y. Xu, M. Xiao, A. Liu, and J. Wu, "Edge resource prediction and auction for distributed spatial crowdsourcing with differential privacy," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 15 554–15 569, 2022.

[33] Y. Liu, Z. Xiong, Q. Hu, D. Niyato, J. Zhang, C. Miao, C. Leung, and Z. Tian, "Vrepchain: A decentralized and privacy-preserving reputation system for social internet of vehicles based on blockchain," *IEEE Trans. Veh. Technol.*, vol. 71, no. 12, pp. 13 242–13 253, 2022. [Online]. Available: https://doi.org/10.1109/TVT.2022.3198004

[34] M. Kadadha, R. Mizouni, S. Singh, H. Otrok, and A. Ouali, "*ABCrowd*: An *A*uction mechanism on *B*lockchain for spatial *Crowd*sourcing," *IEEE Access*, vol. 8, pp. 12 745–12 757, 2020.

[35] H. Lin, S. Garg, J. Hu, G. Kaddoum, M. Peng, and M. S. Hossain, "Blockchain and deep reinforcement learning empowered spatial crowdsourcing in software-defined internet of vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3755–3764, 2021.

[36] Y. Guo, H. Xie, Y. Miao, C. Wang, and X. Jia, "Fedcrowd: A federated and privacy-preserving crowdsourcing platform on blockchain," *IEEE Trans. Serv. Comput.*, vol. 15, no. 4, pp. 2060–2073, 2022.

[37] C. Zhang, Y. Guo, X. Jia, C. Wang, and H. Du, "Enabling proxy-free privacy-preserving and federated crowdsourcing by using blockchain," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6624–6636, 2021.

[38] Y. Xie, Y. Wang, K. Li, X. Zhou, Z. Liu, and K. Li, "Satisfaction-aware task assignment in spatial crowdsourcing," *Inf. Sci.*, vol. 622, pp. 512–535, 2023. [Online]. Available: https://doi.org/10.1016/j.ins.2022.11.081

[39] Y. Xu, M. Xiao, J. Wu, S. Zhang, and G. Gao, "Incentive mechanism for spatial crowdsourcing with unknown social-aware workers: A three-stage stackelberg game approach," *IEEE Trans. Mob. Comput.*, vol. 22, no. 8, pp. 4698–4713, 2023. [Online]. Available: https://doi.org/10.1109/TMC.2022.3157687

[40] J. Shu, X. Liu, Y. Zhang, X. Jia, and R. H. Deng, "Dual-side privacy-preserving task matching for spatial crowdsourcing," *Journal of Network and Computer Applications*, vol. 123, pp. 101–111, 2018.

[41] F. Zhou, J. Li, Y. Lin, J. Wei, and V. K. A. Sandor, "A secure and efficient task matching scheme for spatial crowdsourcing," *IEEE Access*, vol. 8, pp. 155 819–155 831, 2020.

Yuan Liu is a professor at Cyberspace Institute of Advanced Technology of Guangzhou University in Guangdong, China. She achieved her Ph.D degree in School of Computer Engineering from Nanyang Technological University (NTU), Singapore, in 2014. She obtained her B.Sc degree in the honor school, Harbin Institute of Technology, China, in 2010. She was an associate professor at northeastern university, China from 2015 to 2022. From 2014 to 2015, and she ever worked as Research Fellow at Joint NTU-UBC Research Center of Excellence in Active Living for the Elderly (LILY), NTU, Singapore. Her research interests include incentive mechanism design, federated learning, trust management, blockchain consensus protocols, blockchain powered artificial intelligence, and threat intelligence system.

Yanan Zhang received the BS degree in Software Engineering from Changsha University of Technology, in Hunan China. Now he is a master student of Software Engineering, Northeastern University in Shenyang China. His research interests include location privacy mechanism and blockchain applications.

Shen Su is an Associate Professor at Cyberspace Institute of Advanced Technology of Guangzhou University in Guangdong, China. He received his B.E, M.E., and PH.D. degree from Harbin Institute of Technology. His research interests include blockchain security, DNS, route modelling, route security, cyber range, vehicular networks, wireless sensor networks. He has published more than 60 journal and conference papers in such areas, with more than 1200 citations and 7 ESI high-indexed papers. He has served as a guest editor of CMES, and reviewers for Transaction on Industrial Information, IEEE Network magazine, IEEE Internet Computing, Journal of the Franklin Institute, etc.

Lejun Zhang is a Professor at Cyberspace Institute of Advanced Technology of Guangzhou University in Guangdong, China. He received his M.S. degree in computer science and technology in Harbin Institute of Technology and the Ph.D. degrees in computer science and technology at Harbin Engineering University. Now he is currently a professor and Ph.D. Supervisor of the Cyberspace Institute of Advanced Technology, Guangzhou University. He was a Visiting Scholar with Carnegie Mellon University. His research interests include Cyberspace Security, blockchain and information security.

Xiaojiang Du is the Anson Wood Burchard Endowed-Chair Professor in the Department of Electrical and Computer Engineering at Stevens Institute of Technology. He was a tenured professor at Temple University between August 2009 and August 2021. Dr. Du received his B.S. from Tsinghua University, Beijing, China in 1996. He received his M.S. and Ph.D. degree in Electrical Engineering from the University of Maryland, College Park in 2002 and 2003, respectively. His research interests are security, wireless networks, and systems. He has authored over 500 journal and conference papers in these areas, including the top security conferences IEEE S&P, USENIX Security, and NDSS. Dr. Du has been awarded more than 8 million US Dollars research grants from the US National Science Foundation (NSF), Army Research Office, Air Force Research Lab, the State of Pennsylvania, and Amazon. He won the best paper award at several conferences, such as IEEE ICC 2020, IEEE GLOBECOM 2014 and the best poster runner-up award at the ACM MobiHoc 2014. He serves on the editorial boards of three IEEE journals. He is the General Co-Chair of IEEE/ACM IWQoS 2023, the TPC Co-Chair of IEEE CloudNet 2023, and the Lead Chair of the Security (CISS) Symposium of IEEE Globecom 2023. Dr. Du is an IEEE Fellow, an ACM Distinguished Member, and an ACM Life Member

Mohsen Guizani (Fellow, IEEE) received the B.S. (with Distinction), M.S., and Ph.D. degrees in electrical and computer engineering from Syracuse University, Syracuse, NY, USA, in 1985, 1987, and 1990, respectively. He is currently a Professor of Machine Learning and the Associate Provost with the Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE. Previously, he worked in different institutions in the USA. He has authored ten books and more than 800 publications. His research interests include applied machine learning and artificial intelligence, Internet of Things, intelligent autonomous systems, smart city, and cybersecurity. Dr. Guizani has won several research awards, including the 2015 IEEE Communications Society Best Survey Paper Award, the Best ComSoc Journal Paper Award in 2021 as well five Best Paper Awards from ICC and Globecom Conferences. He is also the recipient of the 2017 IEEE Communications Society Wireless Technical Committee Recognition Award, the 2018 AdHoc Technical Committee Recognition Award, and the 2019 IEEE Communications and Information Security Technical Recognition (CISTC) Award. He was listed as a Clarivate Analytics Highly Cited Researcher in Computer Science in 2019, 2020, and 2021. He served as the Editor-in Chief of IEEE NETWORK and is currently serving on the Editorial Boards of many IEEE Transactions and Magazines. He was the Chair of the IEEE Communications Society Wireless Technical Committee and the Chair of the TAOS Technical Committee. He served as the IEEE Computer Society Distinguished Speaker and is currently the IEEE ComSoc Distinguished Lecturer. He was elevated to the IEEE Fellow in 2009.

Zhihong Tian is currently a Professor, and Dean, with the Cyberspace Institute of Advanced Technology, Guangzhou University, Guangdong Province, China. Guangdong Province Universities and Colleges Pearl River Scholar (Distinguished Professor). He is also a part-time Professor at Carlton University, Ottawa, Canada. Previously, he served in different academic and administrative positions at the Harbin Institute of Technology. He has authored over 200 journal and conference papers in these areas. His research interests include computer networks and cyberspace security. His research has been supported in part by the National Natural Science Foundation of China, National Key research and Development Plan of China, National High-tech R&D Program of China (863 Program), and National Basic Research Program of China (973 Program). He also served as a member, Chair, and General Chair of a number of international conferences. He is a distinguished Member of the China Computer Federation, and a senior member of IEEE.