IEEE Open Journal of the
Computer Society

Received 30 October 2022; accepted 9 November 2022. Date of publication 29 November 2022;
date of current version 12 December 2022. The review of this article was arranged by Associate Editor Quan Chen.

Digital Object Identifier 10.1109/OJCS.2022.3222363

Towards Efficient and Privacy-Preserving Versatile Task Allocation for Internet of Vehicles

ZIHAN LI, MINGYANG ZHAO, GUANYU CHEN, CHUAN ZHANG (Member, IEEE), TONG WU (Member, IEEE), AND LIEHUANG ZHU (Senior Member, IEEE)

School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China

CORRESPONDING AUTHOR: CHUAN ZHANG (e-mail: chuanz@bit.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grants 62202051 and 62102027, in part by the China Postdoctoral Science Foundation under Grants 2021M700435, 2021TQ0042, and 2021TQ0041, in part by the BIT Research and Innovation Promoting Project under Grant 2022YCXZ031, and in part by the Shandong Provincial Key Research and Development Program under Grant 2021CXGC010106.

ABSTRACT Nowadays, task allocation has attracted increasing attention in the Internet of Vehicles. To efficiently allocate tasks to suitable workers, users usually need to publish their task interests to the service provider, which brings a serious threat to users' privacy. Existing task allocation schemes either cannot comprehensively preserve user privacy (i.e., requester privacy and worker privacy) or introduce tremendous resource overhead. In this paper, we propose an efficient and privacy-preserving versatile task allocation scheme (PPVTA) for the Internet of vehicles. Specifically, we utilize the randomizable matrix multiplication technique to preserve requester privacy and worker privacy. Then, the polynomial fitting technique is leveraged to enrich the randomizable matrix multiplication to support versatile task allocation functions, such as threshold-based task allocation (PPVTA-I), conjunctive task allocation (PPVTA-II), and task allocation with bilateral access control (PPVTA-III). We formally analyze the security of our constructions to prove the security under the chosen-plain attack. Based on a prototype, experimental results demonstrate that our constructions have acceptable efficiency in practice.

INDEX TERMS Internet of vehicles, task allocation, privacy preservation.

I. INTRODUCTION

With the spread of computation and communication technologies, the Internet of vehicles attracts increasing attention [1], [2], [3], [4], [5]. Thus far, the Internet of vehicles has brought many new services and applications [6], [7], [8], [9], [10] that reinforce the transformation and enhance the users' experience. To achieve these services and applications, it is the first step to build the connection between tasks (e.g., traffic monitoring tasks and travel recommendation tasks) and workers [11], [12]. Task allocation has become a promising paradigm to connect users [13], [14] since it can allocate tasks from requesters to suitable workers based on their task interests.

Despite the appealing benefits, considering user privacy, resource overhead, and versatile task functions, it is challenging for designing such a task allocation scheme in the Internet of vehicles [15], [16], [17]. The first challenge is how to comprehensively preserve user privacy (i.e., worker privacy and requester privacy) during allocating tasks [18], [19]. In the Internet of vehicles, users' task interests usually imply their sensitive information. For example, a driver participating in traffic monitoring tasks can help other users better understand the current traffic conditions, but his or her location and route can be inferred from the tasks. It is obvious that if their sensitive information is not preserved well, users' enthusiasm for participating in the Internet of vehicles will be significantly reduced. The second challenge is how to efficiently allocate tasks to suitable workers [20]. In practical applications, the number of both tasks and workers is huge [21]. Without an efficient allocation scheme, tremendous resource overhead will be introduced to each entity in the system, especially the service provider. The final challenge is how to satisfy the versatile task allocation functions of users. With the increasing requirements of user experience, users usually have multiple task allocation functions, e.g., conjunctive task allocation [22] and task allocation with bilateral access control [23], [24]. Thus, to further enhance user experience and impel more users to participate in the Internet of vehicles, it is crucial to consider users' versatile task allocation functions.

Thus far, to resolve the above concerns, some task allocation schemes [18], [25], [26], [27], [28], [29], [30], [31], [32] are proposed. For example, Shu et al. [25] utilized the secure k-nearest neighbor computation technique to design a privacy-preserving task allocation scheme. In their scheme, users utilize vectors to represent their task interests, and the service provider achieves task allocation over the encrypted vectors. However, their scheme can only resist the known-plaintext attack, which is vulnerable to practical applications. To improve security, some task allocation schemes are proposed based on traditional cryptographic primitives, e.g., Yao's garble circuit [33], bilinear map [34], and Paillier cryptosystem [35]. For example, Yang et al. [26] utilized the bilinear map to design a privacy-preserving task allocation scheme. Based on the Paillier cryptosystem, Zhao et al. [28] proposed a bilateral privacy-preserving task allocation scheme. In these schemes, user privacy is preserved well under the chosen-plain attack. Unfortunately, due to introducing the time-consuming traditional cryptographic tools, the entities incur tremendous resource overhead in the system. In addition, these schemes usually cannot satisfy versatile task allocation functions, which significantly hinders their implementations. To address these challenges, several works have been proposed by combining the randomizable matrix multiplication technique [18], [36], [37]. For example, Zhang et al. [36] utilized the randomized matrix multiplication technique to design a privacy-preserving task allocation scheme. In their scheme, tasks can be allocated to suitable workers without compromising user privacy. However, there are two limitations. The first is that workers' ciphertexts usually are required to be re-encrypted, which introduces additional overhead. When the number of workers is huge, it is not acceptable in practice. Additionally, these schemes ignore the versatility of task allocation functions.

Challenge: How to design an efficient and versatile task allocation scheme for the Internet of vehicles with comprehensive privacy preservation (i.e., worker privacy and requester privacy)?

A. RELATED WORKS

Nowadays, to allocate tasks to suitable workers without compromising user privacy, some privacy-preserving task allocation schemes [18], [25], [26], [27], [28], [32], [36], [38], [39], [40], [41] have been proposed. Specifically, Gong et al. [38] proposed the first task allocation framework with privacy preservation for crowdsourcing. In their framework, worker privacy is preserved well, while the privacy of requesters is ignored. To preserve worker privacy and requester privacy simultaneously, some works [25], [39], [40], [41] have been proposed. Specifically, based on the secure k-nearest neighbor computation technique, Shu et al. [25] proposed an efficient task allocation scheme with privacy preservation. However, their scheme assumes that all the participating entities are trusted and can only resist the known-plaintext attack. To improve security, some works based on traditional cryptographic tools have been proposed. Specifically, based on bilinear maps, Tang et al. [27] designed a bilateral task allocation scheme. In their scheme, tasks are allocated to eligible workers based on their interests in a win-win manner. Additionally, their scheme can preserve requester privacy and worker privacy, simultaneously. Although these schemes can guarantee security under the chosen-plain attack, applying time-consuming tools also introduces tremendous resource overhead on each entity in the system, especially the service provider. To address these challenges above, inspired by matrix multiplication, some works are proposed to achieve efficient task allocation with security under the chosen-plain attack [18], [32], [36], [37]. Specifically, Ni et al. [18] utilized proxy re-encryption to design an accurate task allocation scheme with privacy preservation. In their scheme, both users' reputation and geographic data are considered. However, in their scheme, the matrix dimension grows with the number of tasks. Thus, when the number of tasks is huge, tremendous resource overhead is still introduced to the entities in the system. Zhang et al. [37] utilized the randomized matrix multiplication technique and data perturbation technique to design a privacy-preserving task allocation scheme. However, in their scheme, workers' ciphertexts are required to be re-encrypted on the service provider side. With the re-encryption mechanism, when the number of workers is huge, it is obvious that the service provider will incur tremendous computational overhead. In addition, the above schemes ignore the versatility of task allocation.

B. CONTRIBUTIONS

To deal with the challenges, an efficient privacy-preserving versatile task allocation scheme (PPVTA) is proposed for the Internet of vehicles. Specifically, we summarize the contributions as follows:

- We identify the challenges in designing task allocation schemes for the Internet of vehicles. Then, to deal with the above challenges, we propose an efficient and privacy-preserving versatile task allocation scheme, named PPVTA.
- Based on the randomizable matrix multiplication and polynomial fitting techniques, PPVTA can support versatile task allocation functions in a privacy-preserving manner. Particularly, PPVTA-I, PPVTA-II, and PPVTA-III support threshold-based task allocation, conjunctive task allocation, and task allocation with bilateral access control, respectively.
- Formal security analysis proves that PPVTA can simultaneously preserve requester privacy and worker privacy

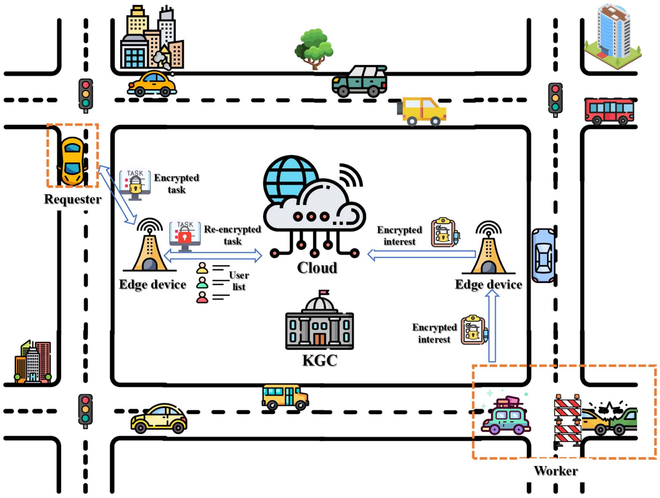


FIGURE 1. System model in PPVTA.

under the chosen-plain attack. Extensive experiments show that our constructions have acceptable efficiency in practice.

Organization: The remainder of our paper is below. The system model and design goals are introduced in Section II. Subsequently, we provide the detailed construction of PPVTA-I in Section III and the discussion of PPVTA-II and PPVTA-III in Section IV. A formal security analysis is provided in Section V, and a performance evaluation is shown in Section VI. Finally, we give a conclusion of this paper in Section VII.

II. MODELS AND DESIGN GOALS

This section first introduces the system model of PPVTA. Subsequently, the threat model of PPVTA is provided. Then, we formulate the design goals of PPVTA.

A. SYSTEM MODEL

As illustrated in Fig. 1, in PPVTA, there are five entities, i.e. workers, requesters, cloud, edge devices, and a key generation center.

- Requesters: The requesters are usually individuals and organizations. They send tasks to edge devices and find workers to complete their tasks.
- Workers: The workers are usually individuals who publish their task interests to edge devices and find suitable tasks.
- Cloud: As a storage service provider, the cloud is responsible for storing encrypted data.
- Edge devices: The edge devices are usually computing service providers in the Internet of vehicles. They are responsible for allocating tasks to suitable workers.
- Key generation center (KGC): As a trusted authority, the KGC generates system parameters and user keys.

At a high level, PPVTA can be described as follows: (1) The KGC generates system parameters. Subsequently, it sends re-encryption keys to edge devices and encryption keys to workers and requesters in the system; (2) Workers send the encrypted interest matrices to edge devices; (3) Edge devices periodically upload the matrices to the cloud for long-term storage; (5) Requesters upload the encrypted task matrices to edge devices; (6) Edge devices re-encrypt the task matrices and allocate these tasks to suitable workers over the ciphertexts.

B. THREAT MODEL

The KGC is considered fully trusted and the interactions with KGC are considered secure. The workers, requesters, cloud, and edge devices are all honest-but-curious. Specifically, they are honest to execute the step of task allocation, however, they try to infer others' private information. Besides, we assume that neither the edge devices nor the cloud would not operate in collusion with other entities. Also, we assume that they do not pretend to be valid workers or requesters. The assumption is consistent with most existing IoV applications. Specifically, the chosen-plain attack (CPA) model is considered in this paper. The adversary can choose some valid plaintexts of interest vectors or task vectors and get the corresponding ciphertexts. Each entity other than the KGC can be an adversary.

C. DESIGN GOALS

The formulated design goals of this paper are below:

- *Utility:* PPVTA should perform efficient and effective task allocation which includes multiple functions, i.e., threshold-based task allocation, conjunctive task allocation, and bilateral task allocation.
- *Privacy:* In each phase of the scheme, PPVTA should well preserve both worker privacy and requester privacy.
- *Efficiency:* PPVTA should achieve task allocation with low computational and communication overhead introduced.

III. PROPOSED PPVTA SCHEMES

PPVTA has three different functions of task allocation. In this section, we only provide the detailed construction of PPVTA-I, which supports threshold-based task allocation. PPVTA-I consists of phases, i.e., setup, worker interest submission, requester task submission, and task allocation. The notations are summarized in Table 1.

A. SETUP

In this phase, the KGC generates secret keys for other entities (e.g., workers, requesters, and edge devices). Assume that M is the maximum number of keywords in the trapdoor of each task. The KGC generates four $(M+2) \times (M+2)$-dimensional invertible random matrices as the master secret keys $\{M_1, M_2\}$ and re-encryption keys $\{B_1, B_2\}$. Next, through secure channels, the KGC sends the re-encryption keys (B_1, B_2) to edge devices.

For worker u_k, the KGC takes $(1, 1, \ldots, 1, 0)$ as the main diagonal and extends the diagonal vector to an $(M+2) \times (M+2)$-dimensional random lower triangular matrix I_k.

TABLE 1. Notations Used in PPVTA

Notation	Description
M	maximum number of keywords
$\mathbb{M}_i$	keyword set of user u_i
$\lvert\mathbb{M}_i\rvert$	number of user u_i's keywords
$w_{i,a}$	a-th keyword of user u_i
$d_{i,a}$	a-th hash of user u_i's keyword
$\{M_1, M_2\}$	master secret keys
$\{B_1, B_2\}$	re-encryption keys
$\{A_{k,1}, A_{k,2}\}$	encryption keys of worker u_k
$\{B_{j,1}, B_{j,2}\}$	encryption keys of requester u_j
$h_s(\cdot)$	keyword-based hash function
$\{\vec{S}_{k,b}\}_{b=1}^{\lvert\mathbb{M}_k\rvert}$	interest vectors of worker u_k
$\{r_{k,b}\}_{b=1}^{\lvert\mathbb{M}_k\rvert}$	random values of worker u_k
$\{\mathcal{S}_{k,b}\}_{b=1}^{\lvert\mathbb{M}_k\rvert}$	interest matrices of worker u_k
$\{\mathsf{E_k}[\mathcal{S}_{k,b}]\}_{b=1}^{\lvert\mathbb{M}_k\rvert}$	encrypted interest matrices of worker u_k
$\{a_{j,i}\}_{i=0}^{M}$	coefficients of requester u_j's keyword function
$\vec{T}_j$	task vector of requester u_j
r_j	random value of requester u_j
$\mathcal{T}_j$	task matrix of requester u_j
$\mathsf{E_j}[\mathcal{T}_j]$	encrypted task matrix of requester u_j
$\mathsf{RE_j}[\mathcal{T}_j]$	re-encrypted task matrix of requester u_j
$\{\Delta_{j,k,b}\}_{b=1}^{\lvert\mathbb{M}_k\rvert}$	results of task allocation between u_j and u_k
t_j	threshold of requester u_j's matching score
W_j	requester u_j's identity set of suitable worker
$\vec{S}_k$	conjunctive interest vector of worker u_k
$\mathbb{A}_i$	attribute set of user u_i
$\mathbb{P}_i$	policy set of user u_i
$e_{i,a}$	a-th hash of user u_i's attribute
$f_{i,a}$	a-th hash of user u_i's policy
$\{g_{i,a}\}_{a=0}^{M}$	coefficients of user u_i's policy function
$\vec{X}_i$	attribute vector of user u_i
$\mathcal{X}_i$	attribute matrix of user u_i
$\mathsf{E_i}[\mathcal{X}_i]$	encrypted attribute matrix of user u_i
$\vec{Y}_i$	policy vector of user u_i
$\mathcal{Y}_i$	policy matrix of user u_i
$\mathsf{E_i}[\mathcal{Y}_i]$	encrypted policy matrix of user u_i

Then, the KGC calculates u_k's encryption keys as

$$\begin{aligned} A_{k,1} &= M_1 \times I_k, \\ A_{k,2} &= I_k \times M_2. \end{aligned} \tag{1}$$

Next, $\{A_{k,1}, A_{k,2}\}$ are sent to the worker u_k.

Similarly, for requester u_j, the KGC extends the same diagonal vector $(1, 1, \ldots, 1, 0)$ to an $(M+2) \times (M+2)$-dimensional random lower triangular matrix I_j. Note that both matrix I_k for worker u_k and matrix I_j for requester u_j are unique. Then, the KGC computes u_j's encryption keys as

$$\begin{aligned} B_{j,1} &= B_1^{-1} \times M_2^{-1} \times I_j, \\ B_{j,2} &= I_j \times M_1^{-1} \times B_2^{-1}. \end{aligned} \tag{2}$$

Next, $\{B_{j,1}, B_{j,2}\}$ are sent to the requester u_j.

B. WORKER INTEREST SUBMISSION

Each worker sends the encrypted interest matrix to the edge devices in this phase. Based on the set of keywords $\mathbb{M}_k$ that the worker u_k is interested in, u_k first calculates the hashes of keywords $\{d_{k,b}\}_{b=1}^{|\mathbb{M}_k|}$ as

$$d_{k,b} = h_s(w_{k,b}). \tag{3}$$

Then, u_k generates $(M+2)$-dimensional interest vectors $\{\vec{S}_{k,b}\}_{b=1}^{|\mathbb{M}_k|}$ as

$$\vec{S}_{k,b} = \left(r_{k,b} \cdot d_{k,b}^0, r_{k,b} \cdot d_{k,b}^1, \ldots, r_{k,b} \cdot d_{k,b}^M, 0\right), \tag{4}$$

where $\{r_{k,b}\}_{b=1}^{|\mathbb{M}_k|}$ are random values.

Subsequently, u_k takes interest vector $vecS_{k,b}$ as the diagonal vector and extends it towards the interest matrix $\mathcal{S}_{k,b}$, an $(M+2) \times (M+2)$-dimensional lower triangular random matrix, where $\vec{S}_{k,b}$ is the main diagonal of $\mathcal{S}_{k,b}$.

Then, with $\{A_{k,1}, A_{k,2}\}$, u_k encrypts the interest matrices as

$$\mathsf{E_k}[\mathcal{S}_{k,b}] = A_{k,1} \times \mathcal{S}_{k,b} \times A_{k,2}. \tag{5}$$

At last, u_k sends $\{\mathsf{E_k}[\mathcal{S}_{k,b}]\}_{b=1}^{|\mathbb{M}_k|}$, the encrypted interest matrices to the edge devices. After a period of storage for the ciphertexts from the worker u_k, the edge devices upload the ciphertexts $\{\mathsf{E_k}[\mathcal{S}_{k,b}]\}_{b=1}^{|\mathbb{M}_k|}$ to the cloud.

C. REQUESTER TASK SUBMISSION

Each requester encrypts the task requirement keywords and publishes the ciphertexts to the edge devices in this phase. Subsequently, the edge devices re-encrypt the ciphertexts.

Step 1: The requester u_j specifies the requirement which consists of $|\mathbb{M}_j|$ keywords that u_j is interested in. To make the number of keywords consistent, $M - |\mathbb{M}_j|$ dummy keywords $\{w_{j,|\mathbb{M}_j|+1}, w_{j,|\mathbb{M}_j|+2}, \ldots, w_{j,M}\}$ are added to $\mathbb{M}_j$. Note that each dummy keyword is different from any real dictionary word and thus it has no impact on the matching result. Based on $\mathbb{M}_j$, u_j first calculates the hashes of keywords $\{d_{j,c}\}_{c=1}^{M}$ as

$$d_{j,c} = h_s(w_{j,c}). \tag{6}$$

Considering the tremendous overhead of m-degree polynomial root tracking which requires 2 m^2 complex floating point operation, we construct a polynomial fitting function of degree M, called the keyword function, as

$$\begin{aligned} f_j(x) &= (x - d_{j,1})(x - d_{j,2}) \cdots (x - d_{j,M}) \\ &= a_{j,0} + a_{j,1}x + \cdots + a_{j,M}x^M, \end{aligned} \tag{7}$$

to hide the keywords. Then, u_j generates the $(M+2)$-dimensional task vector $\vec{T}_j$ as

$$\vec{T}_j = (r_j \cdot a_{j,0}, r_j \cdot a_{j,1}, \ldots, r_j \cdot a_{j,M}, 0), \tag{8}$$

where r_j is a random value.

Subsequently, u_j extends the vector to an $(M+2) \times (M+2)$-dimensional lower triangular random matrix $\mathcal{T}_j$, where the main diagonal is $\vec{T}_j$. Then, u_j encrypted the matrix with $\{B_{j,1}, B_{j,2}\}$ as

$$\mathsf{E_j}[\mathcal{T}_j] = B_{j,1} \times \mathcal{T}_j \times B_{j,2}. \tag{9}$$

At last, u_j sends $\mathsf{E_j}[\mathcal{T}_j]$ to the edge devices.

Step 2: After receiving $\mathsf{E_j}[\mathcal{T}_j]$ from u_j, the edge devices re-encrypt the data utilizing (B_1, B_2) as

$$\mathsf{RE_j}[\mathcal{T}_j] = B_1 \times \mathsf{E_j}[\mathcal{T}_j] \times B_2. \tag{10}$$

D. TASK ALLOCATION

Finally, the edge devices execute task allocation in the phase. After finding the ciphertexts of worker u_k's interest matrices in the temporary storage of the edge devices or obtaining the ciphertexts from the long-term storage of the cloud, for worker u_k and requester u_j, the edge devices compute $\{\Delta_{j,k,b}\}_{b=1}^{|\mathbb{M}_k|}$ as

$$\begin{aligned}\Delta_{j,k,b} &= tr(\mathsf{RE_j}[\mathcal{T}_j] \times \mathsf{E_k}[\mathcal{S}_{k,b}])\\ &= tr(B_1 \times B_1^{-1} \times M_2^{-1} \times I_j \times \mathcal{T}_j\\ &\quad \times I_j \times M_1^{-1} \times B_2^{-1} \times B_2\\ &\quad \times M_1 \times I_k \times \mathcal{S}_{k,b} \times I_k \times M_2)\\ &= tr(M_2^{-1} \times I_j \times \mathcal{T}_j \times I_j \times I_k \times \mathcal{S}_{k,b} \times I_k \times M_2)\\ &= \vec{T}_j \times \vec{S}_{k,b}. \end{aligned} \tag{11}$$

In (11), $tr(\cdot)$ represents the matrix trace function. Obviously, $\Delta_{j,k,b} = 0$ only if $\mathcal{S}_{k,b} \in \mathbb{M}_j$. The number of zero values in $\{\Delta_{j,k,b}\}_{b=1}^{|\mathbb{M}_k|}$ is considered as $score_{j,k}$, the matching score between u_j and u_k. If $score_{j,k} \geq t_j$, the edge devices will put the worker u_k's identity id_k into W_j, the requester u_j's identity set of capable workers. Finally, the edge devices get $W_j = \{id_k\}_{score_{j,k} \geq t_j}$ after computing the matching scores.

IV. DISCUSSION

In addition to threshold-based task allocation, PPVTA can support some other functions, such as conjunctive task allocation and task allocation with bilateral access control. In this section, we focus on the other two task allocation functions of PPVTA.

A. CONJUNCTIVE TASK ALLOCATION

In practical applications, the workers are allowed to find tasks that satisfy all of their interests. To meet the requirement, PPVTA-II supports conjunctive task allocation. Specifically, unlike threshold-based task allocation, edge devices do not calculate the matching score. In the phase of worker interest submission, after the worker u_k generates the interest vectors $\{\vec{S}_{k,b}\}_{b=1}^{|\mathbb{M}_k|}$, u_k sums these vectors as the conjunctive interest vector

$$\vec{S}_k = \sum_{b=1}^{|\mathbb{M}_k|} \vec{S}_{k,b}. \tag{12}$$

Then, u_k generates the conjunctive interest matrix and encrypted matrix in the same way as PPVTA-I so that u_k hides all his or her keywords of interests in one matrix. Then, u_k sends the matrix to the edge devices. In the phase of task allocation, for worker u_k and requester u_j, the edge devices only compute one task allocation result $\Delta_{j,k}$ as

$$\begin{aligned}\Delta_{j,k} &= tr(\mathsf{RE_j}[\mathcal{T}_j] \times \mathsf{E_k}[\mathcal{S}_k])\\ &= \vec{T}_j \times \vec{S}_k. \end{aligned} \tag{13}$$

$\Delta_{j,k} = 0$ only if $\mathbb{M}_k \subseteq \mathbb{M}_j$, and if $\Delta_{j,k} = 0$, the edge devices will get id_k into W_j, u_j's set of suitable worker identities.

B. TASK ALLOCATION WITH BILATERAL ACCESS CONTROL

To improve the accuracy of sensing data, both workers and requesters may have the requirements of access control. PPVTA-III supports task allocation with bilateral access control. Specifically, each user generates two matrices, i.e., policy matrix and attribute matrix, and the principle of task allocation in PPVTA-III is similar to PPVTA-II. In PPVTA-III, the matrices are extended to $M+4$ dimensions, where M is the maximum number of both requester u_j and worker u_k's policies.

In the phase of setup, all the secret keys are extended to $(M+4)$-dimensional matrices and the main diagonals of both I_k and I_j are $(M+4)$-dimensional vectors $(1, 1, \ldots, 1, 0)$. Then, the KGC generates the secret keys and sends them to the edge devices, workers, and requesters.

In the phase of worker interest submission, the worker u_k first adds $M - |\mathbb{P}_k|$ dummy keywords to $\mathbb{P}_k$ and then calculates the hashes of attributes $\{e_{k,b}\}_{b=1}^{|\mathbb{A}_k|}$ and hashes of policies $\{f_{k,b}\}_{b=1}^{M}$. Similarly to the generation of the task vector in PPVTA-I, u_k generates coefficients of the policy function $\{g_{k,b}\}_{b=0}^{M}$. Then, u_k generates an $(M+4)$-dimensional attribute vector $\vec{X}_k$ as

$$\vec{X}_k = \left(\sum_{b=1}^{|\mathbb{A}_k|} r_{k,b} \cdot e_{k,b}^0, \sum_{b=1}^{|\mathbb{A}_k|} r_{k,b} \cdot e_{k,b}^1, \ldots, \sum_{b=1}^{|\mathbb{A}_k|} r_{k,b} \cdot e_{k,b}^M, s_{k,1}, s_{k,2}, 0 \right) \tag{14}$$

and an $(M+4)$-dimensional policy vector $\vec{Y}_k$ as

$$\vec{Y}_k = (s_{k,3} \cdot g_{k,0}, s_{k,3} \cdot g_{k,1}, \ldots, s_{k,3} \cdot g_{k,M}, -s_{k,1}, s_{k,2}, 0), \tag{15}$$

where $\{r_{k,b}\}_{b=1}^{|\mathbb{A}_k|}$, $s_{k,1}$, $s_{k,2}$, and $s_{k,3}$ are random values. Next, u_k generates and encrypts the attribute matrix $\mathcal{X}_k$ and policy matrix $\mathcal{Y}_k$, and sends $\mathsf{E_k}[\mathcal{X}_k]$ and $\mathsf{E_k}[\mathcal{Y}_k]$ to the edge devices.

Similarly, in the phase of requester task submission, the requester u_j generates an $(M+4)$-dimensional attribute vector $\vec{X}_j$ as

$$\vec{X}_j = \left(\sum_{c=1}^{|\mathbb{A}_j|} r_{j,c}\cdot e^0_{j,c}, \sum_{c=1}^{|\mathbb{A}_j|} r_{j,c}\cdot e^1_{j,c}, \dots, \sum_{c=1}^{|\mathbb{A}_j|} r_{j,c}\cdot e^M_{j,c},\right.$$
$$\left. s_{j,1}, s_{j,2}, 0\right) \tag{16}$$

and an $(M+4)$-dimensional policy vector $\vec{Y}_j$ as

$$\vec{Y}_j = (s_{j,3}\cdot g_{j,0}, s_{j,3}\cdot g_{j,1}, \dots, s_{j,3}\cdot g_{j,M}, s_{j,1}, -s_{j,2}, 0), \tag{17}$$

where $\{r_{j,c}\}_{c=1}^{|\mathbb{A}_j|}$, $s_{j,1}$, $s_{j,2}$, and $s_{j,3}$ are random values. Then, u_j also generates and encrypts the two matrices and sends the encrypted attribute and policy matrices to the edge devices. Next, the edge devices re-encrypt the two matrices.

In the phase of task allocation, the edge devices compute the task allocation result $\Delta_{j,k}$ as

$$\begin{aligned}\Delta_{j,k} &= tr(\mathsf{RE_j}[\mathcal{Y}_j]\times\mathsf{E_k}[\mathcal{X}_k] + \mathsf{RE_j}[\mathcal{X}_j]\times\mathsf{E_k}[\mathcal{Y}_k])\\ &= \vec{Y}_j\times\vec{X}_k + \vec{X}_j\times\vec{Y}_k.\end{aligned} \tag{18}$$

$\Delta_{j,k}=0$ only if $\mathbb{A}_k\subseteq\mathbb{P}_j$ and $\mathbb{A}_j\subseteq\mathbb{P}_k$. Note that because of the random values $s_{k,1}$, $s_{k,2}$, $s_{j,1}$, and $s_{j,2}$, the edge devices cannot get any privacy by only calculating $tr(\mathsf{RE_j}[\mathcal{Y}_j]\times\mathsf{E_k}[\mathcal{X}_k])$ or $tr(\mathsf{RE_j}[\mathcal{X}_j]\times\mathsf{E_k}[\mathcal{Y}_k])$. If $\Delta_{j,k}=0$, the edge devices will add the worker u_k's identity id_k to u_j's worker identity set W_j.

V. SECURITY ANALYSIS

We first provide a formal security analysis of the security of our proposed encryption and re-encryption methods. Then, we prove that both requester privacy and worker privacy are preserved well in PPVTA. Since the encryption process of requesters is similar to that of workers, we focus on the security of the workers' encryption process due to space limitations.

Theorem 1: Our encryption method is secure against the chosen-plain attack.

Proof: As shown in Fig. 2, we provide an experiment between challenger $\mathcal{C}$ and adversary $\mathcal{A}$. Based on the experiment, we define security under the CPA model as

$$\left|Adv_{CPA,\mathcal{A}}[b'=b]-\frac{1}{2}\right|\le\epsilon,$$

where ϵ is negligible.

1: ***Setup:*** The challenger $\mathcal{C}$ initializes two $(M+2)\times(M+2)$-dimensional matrices $\{M_1, M_2\}$. $\{M_1, M_2\}$ denote the master secret keys.
2: ***Phase 1:*** The adversary $\mathcal{A}$ applies for the encryption keys (A_1, A_2) from $\mathcal{C}$. Then, $\mathcal{C}$ chooses an $(M+2)\times(M+2)$-dimensional irreversible lower triangular matrix I at random and its main diagonal elements are $(1, 1, \cdots, 0)$. Based on I, $\mathcal{C}$ generates and then publishes the encryption keys (A_1, A_2).
3: ***Challenge:*** $\mathcal{A}$ chooses the encryption keys (A_1, A_2) and two databases $DB_0 = (X_{0,1}, X_{0,2}, \cdots, X_{0,n})$ and $DB_1 = (X_{1,1}, X_{1,2}, \cdots, X_{1,n})$, where $\{X_{0,i}, X_{1,i}\}_{i=0}^{n}\in|\mathbb{Z}|_{M+2}$. $\mathcal{C}$ flips a random binary coin $b\in\{0,1\}$ outside the view of $\mathcal{A}$. Then, $\mathcal{C}$ extends each vector $X_{b,i}$ in DB_b to an $(M+2)\times(M+2)$-dimensional random lower triangular matrix $\mathcal{X}_{b,i}$. Subsequently, $\mathcal{C}$ encrypts $\{\mathcal{X}_{b,i}\}_{i=0}^{n}$ to generate $EDB_b=\{E[\mathcal{X}_{b,i}]\}_{i=0}^{n}$ and publishes EDB_b.
4: ***Phase 2:*** As challenger $\mathcal{C}$ did in ***Phase 1***, $\mathcal{C}$ requests the encryption keys.
5: ***Guess:*** Adversary $\mathcal{A}$ will submit the guess b'.

FIGURE 2. An Experiment between the adversary and the challenger.

Assume that $X_0=(x_{0,0}, x_{0,1}, \dots, x_{0,M+2})$ is the vector to be encrypted. In the ***Challenge*** phase, the challenger $\mathcal{C}$ sets X_0 as the main diagonal and then generates a random lower triangular matrix $\mathcal{X}_0$. Next, $\mathcal{C}$ utilizes the secret keys (A_1, A_2) to generate the ciphertext $A_1\times\mathcal{X}_0\times A_2$. Assume that the elements in A_1, $\mathcal{X}_0$, and the production of $A_1\times\mathcal{X}_0$ are $a_{1,i,j}$, $x_{0,i,j}$, and $z_{1,i,j}$, respectively. Then, $\mathcal{C}$ computes

$$\begin{aligned}z_{1,i,j} &= a_{1,i,1}x_{0,1,j}+a_{1,i,2}x_{0,2,j}+\cdots+a_{1,i,M+2}x_{0,M+2,j}\\ &= \sum_{k=1}^{M+2} a_{1,k,j}x_{0,k,j},\end{aligned} \tag{19}$$

where

$$\begin{cases}\forall M+2\ge j>i\ge 1, x_{0,i,j}=0\\ \forall M+2\ge i=j\ge 1, x_{0,i,j}=x_{0,i}\\ \text{otherwise}, x_{0,i,j}=*.\end{cases} \tag{20}$$

For simplicity, $*$ denotes the random values in $\mathcal{X}_0$. Next, we assume that the elements in A_2 and the production of $A_1\times\mathcal{X}_0\times A_2$ are $a_{2,i,j}$ and $z_{2,i,j}$, respectively. Then, $\mathcal{C}$ computes

$$\begin{aligned}z_{2,i,j} &= z_{1,i,1}z_{2,1,j}+z_{1,i,2}z_{2,2,j}+\cdots+z_{1,i,M+2}z_{2,M+2,j}\\ &= \sum_{k=1}^{M+2} z_{1,k,j}z_{2,k,j} = \sum_{k=1}^{M+2}\left(\sum_{l=1}^{M+2} a_{1,l,j}x_{0,l,j}\right)z_{2,k,j}.\end{aligned} \tag{21}$$

It can be seen that $\forall M+2\ge j>i\ge 1$, $x_{0,i,j}$ is a one-time random value. Also, $\{a_{1,i,j}\}_{i,j}^{M+2}$ and $\{a_{2,i,j}\}_{i,j}^{M+2}$ are fixed values. Therefore, it is obvious that $z_{2,i,j}$ is a random value associated with $x_{0,i,j}$. That is, although adversary $\mathcal{A}$ can request the corresponding ciphertexts from $\mathcal{C}$ continuously, the ciphertexts look random from the view of $\mathcal{A}$. Thus, $\mathcal{A}$ has a negligible advantage to distinguish the corresponding plaintexts. Subsequently, $\mathcal{A}$ can only guess $b'=0$ or $b'=1$ at random. It can be seen that

$$\left|Adv_{CPA,\mathcal{A}}[b'=b]-\frac{1}{2}\right|\le\epsilon,$$

where ϵ is negligible. Therefore, Theorem 1 is proven.

Theorem 2: Our re-encryption method for requesters is secure against the chosen-plain attack.

Proof: Similar to the security analysis of the encryption method, we can prove that the proposed re-encryption method is secure against the chosen-plain attack. Thus, we omit the detailed proof.

Theorem 3: If both Theorems 1 and 2 are proven, the privacy of interest and task will be not leaked to any adversary.

Proof: In our proposed construction, workers and requesters do not receive data from other entities. Since the encryption and re-encryption methods have been proven to be secure against the chosen-plain attack, workers and requesters cannot infer others' private data based on their secret keys. For the cloud, it knows $\{\mathsf{E}_{\mathsf{k}}[\mathcal{S}_{k,b}]\}_{b=1}^{|\mathbb{M}_k|}$ and $\mathsf{RE}_{\mathsf{j}}[\mathcal{T}_j]$. Since Theorem 1 is proven, the cloud cannot break the privacy of interest and task from the ciphertexts $\{\mathsf{E}_{\mathsf{k}}[\mathcal{S}_{k,b}]\}_{b=1}^{|\mathbb{M}_k|}$ and the re-encrypted ciphertexts $\mathsf{RE}_{\mathsf{j}}[\mathcal{T}_j]$. For the edge devices, it knows $\{\mathsf{E}_{\mathsf{k}}[\mathcal{S}_{k,b}]\}_{b=1}^{|\mathbb{M}_k|}$, B_1, B_2, $\mathsf{E}_{\mathsf{j}}[\mathcal{T}_j]$, $\mathsf{RE}_{\mathsf{j}}[\mathcal{T}_j]$, $score_{j,k}$, $\delta_{j,k,b}$, and t_j. Similar to the cloud, the edge devices cannot obtain any private data from $\{\mathsf{E}_{\mathsf{k}}[\mathcal{S}_{k,b}]\}_{b=1}^{|\mathbb{M}_k|}$, $\mathsf{E}_{\mathsf{j}}[\mathcal{T}_j]$ and $\mathsf{RE}_{\mathsf{j}}[\mathcal{T}_j]$. Since $score_{j,k}$, $\delta_{j,k,b}$, and t_j do not contain private data of interests and tasks, the edge device cannot break privacy.

VI. PERFORMANCE ANALYSIS

In this section, based on a prototype, we analyze the performance of our constructions, i.e., PPVTA-I, PPVTA-II, and PPVTA-III. We make comparisons between our constructions and the two most recent multi-keyword task allocation schemes FRUIT [36] and SETM [26].

A. EXPERIMENTAL CONFIGURATION

The cloud server is deployed on the aliyun platform,[1] which is 64-bit, 4 CPUs with 16 GB of RAM. We utilize four laptops with 16 GB of RAM to act as edge devices. Users communicate with the laptops on their mobile phones. In our constructions and FRUIT, the security parameter λ equals 80, and the programs are coded in Java by using the Jama library. In SETM, programs are coded in Java by using the JPBC library. The Type A curve with 80-bit security is selected. Each experiment is executed ten times, and we record the average running time as the final experimental results. The number of keywords ranges from 10 to 100.

B. EXPERIMENTAL EVALUATION

1) ON THE WORKER SIDE

We evaluate the computational costs on the worker side and illustrate the experimental results in Fig. 3. Particularly, in Fig. 3, we range the number of keywords from 5 to 100. It can be seen that PPVTA-II and PPVTA-III are more efficient than other schemes. The computational costs of PPVTA-I are similar to that of FRUIT, and with the increase in the number of keywords, their computational costs are higher than PPVTA-II, PPVTA-III, and SETM. Specifically, PPVTA-II and PPVTA-III require only $2T_{M+2}$ and $4T_{M+4}$, where T_{M+2} represents the time for an $(M+2) \times (M+2)$-dimensional matrix multiplication operation and T_{M+4} represents the time for an $(M+4) \times (M+4)$-dimensional matrix multiplication operation. In SETM, the worker requires $4T_{exp} + |\mathbb{M}_k|T_{mul}$ to encrypt his or her interest, where T_{exp} denotes the time for executing an exponent operation and T_{mul} denotes the time for executing a multiple operation in the bilinear maps. In addition, it can be seen that the computational costs of PPVTA-I and FRUIT grow linearly with the number of keywords, and they require $2|\mathbb{M}_k|T_{M+2}$ and $2|\mathbb{M}_k|T_{M+4}$, respectively.

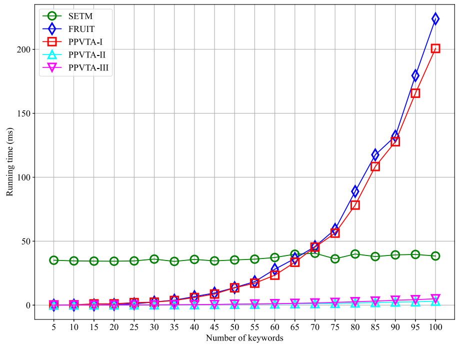


FIGURE 3. Computational costs w.r.t. number of keywords on the worker side.

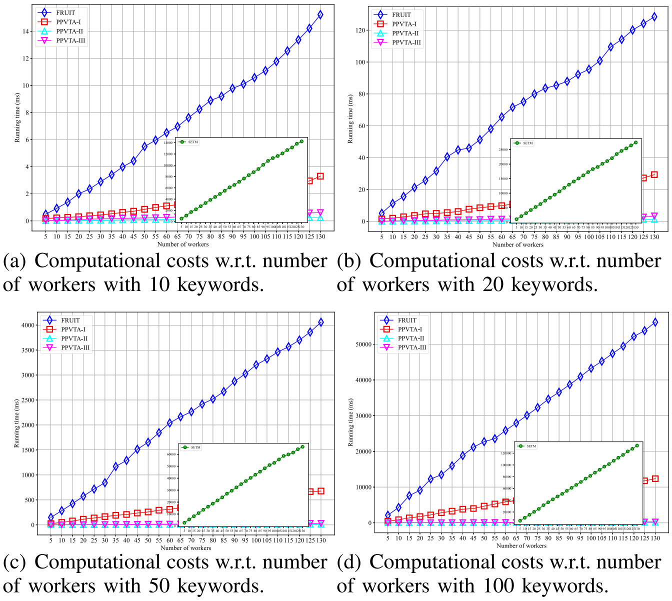
2) ON THE EDGE DEVICES

Next, we evaluate the computational costs of the edge devices. Specifically, the number of keywords is set as 10, 20, 50, and 100. The number of workers ranges from 5 to 130. The experimental results are shown in Fig. 4. Since SETM relies on time-consuming pairing operations to perform the matching operations, the computational costs of SETM are much higher than other schemes, which are designed based on the randomizable matrix multiplication technique. Also, it can be seen that FRUIT has higher computational costs than PPVTA-I, PPVTA-II, and PPVTA-III. This is because, in FRUIT, additional matrix operations are required to re-encrypt the workers' ciphertexts. In addition, PPVTA-I has higher computational costs than PPVTA-II and PPVTA-III. This is because PPVTA-I requires $(|\mathbb{M}_k| + 2)T_{M+2}$ to perform the matching operations, while in PPVTA-II and PPVTA-III, only T_{M+2} and $2T_{M+4}$ are required, respectively.

3) ON THE REQUESTER SIDE

Then, we evaluate the computational cost of the requester side and range the number of keywords from 5 to 100. We illustrate the experimental results in Fig. 5. Particularly, to generate task ciphertext, the computational cost in SETM is $(3|\mathbb{M}_k| + 3)T_{exp} + 3|\mathbb{M}_k|T_{mul}$, which is much higher than other schemes. Additionally, since

[1] https://www.aliyun.com/?utm_content=se_1012440662



(a) Computational costs w.r.t. number of workers with 10 keywords.

(b) Computational costs w.r.t. number of workers with 20 keywords.

(c) Computational costs w.r.t. number of workers with 50 keywords.

(d) Computational costs w.r.t. number of workers with 100 keywords.

FIGURE 4. Computational costs w.r.t. number of workers on the edge devices.

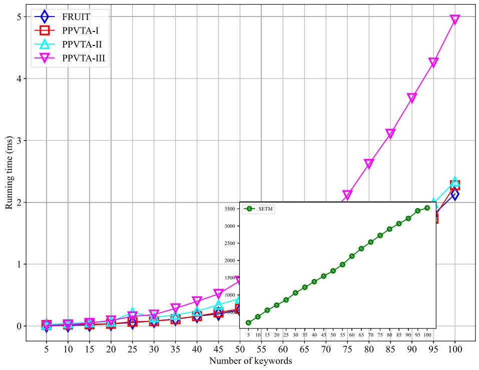


FIGURE 5. Computational costs w.r.t. number of keywords on the requester side.

PPVTA-III needs to generate attribute and policy matrices, while other schemes only need to generate an attribute matrix, PPVTA-III has higher computational costs than other schemes. In summary, our proposed constructions have practically acceptable efficiency.

VII. CONCLUSION

In this paper, based on the techniques of polynomial fitting and randomizable matrix multiplication, an efficient and privacy-preserving versatile task allocation scheme (PPVTA) is proposed for the Internet of vehicles. PPVTA can support versatile task allocation functions with privacy preservation in practical applications. Particularly, PPVTA-I, PPVTA-II, and PPVTA-III support threshold-based task allocation, conjunctive task allocation, and task allocation with bilateral access control, respectively. Security analysis proves that in our constructions, both worker privacy and requester privacy are preserved well under the chosen-plain attack. Experimental results on a prototype demonstrate that our constructions have acceptable efficiency in practice. For future work, to provide more versatile task allocation functions, we will further extend our constructions to support location-based and reputation-based task allocation in the Internet of vehicles.

REFERENCES

[1] A. Hbaieb, S. Ayed, and L. Chaari, "A survey of trust management in the internet of vehicles," *Comput. Netw.*, vol. 203, 2022, Art. no. 108558.

[2] R. Alireza, F. Erfanian, C. T. Timmerer, and H. Hellwagner, "QoCoVi: QoE- and cost-aware adaptive video streaming for the internet of vehicles," *Comput. Commun.*, vol. 190, pp. 1–9, 2022.

[3] H Yi, "A secure blockchain system for internet of vehicles based on 6G-enabled network in box," *Comput. Commun.*, vol. 186, pp. 45–50, 2022.

[4] C. Ksouri, I. Jemili, M. Mosbah, and A. Belghith, "Towards general internet of vehicles networking: Routing protocols survey," *Concurrency Comput. Pract. Experience*, vol. 34 no. 7, 2022, Art. no. e5994.

[5] W. Zhang et al., "Optimizing federated learning in distributed industrial IoT: A multi-agent approach," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3688–3703, Dec. 2021.

[6] A. Saad, A. Shalaby, and A. A. Mohamed, "Research on the internet of vehicles assisted traffic management systems for observing traffic density," *Comput. Elect. Eng.*, vol. 101, 2022, Art. no. 108100.

[7] M. R. Anwar, S. Wang, M. F. Akram, S. Raza, and S. Mahmood, "5G-enabled MEC: A distributed traffic steering for seamless service migration of internet of vehicles," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 648–661, Jan. 2022.

[8] N. Zhao et al., "Deep-reinforcement-learning-based latency minimization in edge intelligence over vehicular networks," *IEEE Internet Things J.*, vol. 9, no. 2, pp. 1300–1312, Jan. 2022.

[9] X. Bai, S. Chen, Y. Shi, C. Liang, and X. Lv, "Blockchain-based authentication and proof-of-reputation mechanism for trust data sharing in internet of vehicles," *Ad Hoc Sens. Wirel. Netw.*, vol. 53, no. 1-2, pp. 85–113, Mar. 2022.

[10] C. Zhu, G. Pastor, Y. Xiao, and A. Ylä-Jääski, "Vehicular fog computing for video crowdsourcing: Applications, feasibility, and challenges," *IEEE Commun. Mag.*, vol. 56, no. 10, pp. 58–63, Oct. 2018.

[11] D. T. Tri, V. Nguyen, V. -N. Pham, L. N. T. Huynh, M. D. Hossain, and E. -N. Huh, "Modeling data redundancy and cost-aware task allocation in mec-enabled internet-of-vehicles applications," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1687–1701, Jan. 2021.

[12] T. Wang, X. Luo, and W. Zhao, "Improving the performance of tasks offloading for internet of vehicles via deep reinforcement learning methods," *IET Commun.*, vol. 16, no. 10, pp. 1230–1240, 2022.

[13] D. Lee, S. H. Lee, N. Masoud, M. S. Krishnan, and V. C. Li, "Digital twin-driven deep reinforcement learning for adaptive task allocation in robotic construction," *Adv. Eng. Informat.*, vol. 53, 2022, Art. no. 101710.

[14] Z. Wang et al., "Joint flight scheduling and task allocation for secure data collection in UAV-aided IoTs," *Comput. Netw.*, vol. 207, 2022, Art. no. 108849.

[15] Y. Jiang, K. Zhang, Y. Qian, and R. Q. Hu, "Preserving location privacy and accurate task allocation in edge-assisted mobile crowdsensing," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2022, pp. 704–709.

[16] Y. Qian, Y. Ma, J. Chen, D. Wu, D. Tian, and K. Hwang, "Optimal location privacy preserving and service quality guaranteed task allocation in vehicle-based crowdsensing networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4367–4375, Jul. 2021.

[17] M. Huang, C. M. Victor, A. L. Liu, and N. N. Xiong, "TMA-DPSO: Towards efficient multi-task allocation with time constraints for next generation multiple access," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 5, pp. 1652–1666, May 2022.

[18] J. Ni, K. Zhang, Q. Xia, X. Lin, and X. S. Shen, "Enabling strong privacy preservation and accurate task allocation for mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 19, no. 6, pp. 1317–1331, Jun. 2020.

[19] Z. Wang et al., "Personalized privacy-preserving task allocation for mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 18, no. 6, pp. 1330–1341, Jun. 2019.

[20] Q. Li, M. Li, B. Q. Vo, and R. Kowalczyk, "An efficient algorithm for task allocation with the budget constraint," *Expert Syst. Appl.*, vol. 210, 2022, Art. no. 118279.

[21] 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. Mobile Comput.*, vol. 21, no. 7, pp. 2436–2450, Jul. 2022.

[22] Q. Gan et al., "Verifiable searchable symmetric encryption for conjunctive keyword queries in cloud storage," *Front. Comput. Sci.*, vol. 16, no. 6, 2022, Art. no. 166820.

[23] C. Zhang et al., "Achieving fuzzy matching data sharing for secure cloud-edge communication," *China Commun.*, vol. 19, no. 7, pp. 257–276, 2022.

[24] S. Xu et al., "Match in my way: Fine-grained bilateral access control for secure cloud-fog computing," *IEEE Trans. Dependable Secur. Comput.*, vol. 19, no. 2, pp. 1064–1077, Mar./Apr. 2022.

[25] 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, Jan. 2021.

[26] K. Yang and S. Dutta, "Secure and efficient task matching with multi-keyword in multi-requester and multi-worker crowdsourcing," in *Proc. 29th IEEE/ACM Int. Symp. Qual. Serv.*, 2021, pp. 1–6.

[27] W. Tang, K. Zhang, J. Ren, Y. zhang, and X. Shen, "Privacy-preserving task recommendation with win-win incentives for mobile crowdsourcing," *Inf. Sci.*, vol. 527, pp. 477–492, 2020.

[28] B. Zhao, S. Tang, X. Liu, X. Zhang, and W. -N. Chen, "iTAM: Bilateral privacy-preserving task assignment for mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 20, no. 12, pp. 3351–3366, Dec. 2021.

[29] X. Wang et al., "Online spatial crowdsensing with expertise-aware truth inference and task allocation," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 412–427, Jan. 2022.

[30] W. Chen, W. Wang, Z. Li, Q. Ye, and Q. Wu, "Joint pricing and task allocation for blockchain empowered crowd spectrum sensing," *Peer-to-Peer Netw. Appl.*, vol. 15, no. 1, pp. 783–792, 2022.

[31] F. Donglai and L. Yanhua, "Trust-aware task allocation in collaborative crowdsourcing model," *Comput. J.*, vol. 64, no. 6, pp. 929–940, 2021.

[32] J. Ni, K. Zhang, X. Lin, Q. Xia, and X. S. Shen, "Privacy-preserving mobile crowdsensing for located-based applications," in *Proc. IEEE Int. Conf. Commun.*, 2017, pp. 1–6.

[33] C. Cai, Y. Zheng, and C. Wang, "Leveraging crowdsensed data streams to discover and sell knowledge: A secure and efficient realization," in *Proc. 38th IEEE Int. Conf. Distrib. Comput. Syst.*, 2018, pp. 589–599.

[34] G. Ateniese, D. Francati, D. Nuñez, and D. Venturi, "Match me if you can: Matchmaking encryption and its applications," *J. Cryptology*, vol. 34, no. 3, 2021, Art. no. 16.

[35] H. Wu, L. Wang, K. Cheng, D. Yang, J. Tang, and G. Xue, "Privacy-enhanced and practical truth discovery in two-server mobile crowdsensing," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 3, pp. 1740–1755, May/Jun. 2022.

[36] C. Zhang, M. Zhao, L. Zhu, W. Zhang, T. Wu, and J. Ni, "FRUIT: A blockchain-based efficient and privacy-preserving quality-aware incentive scheme," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 12, pp. 3343–3357, Oct. 2022.

[37] C. Zhang, M. Zhao, L. Zhu, T. Wu, and X. Liu, "Enabling efficient and strong privacy-preserving truth discovery in mobile crowdsensing," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 3569–3581, Sep. 2022.

[38] Y. Gong, Y. Guo, and Y. Fang, "A privacy-preserving task recommendation framework for mobile crowdsourcing," in *Proc. IEEE Glob. Commun. Conf.*, 2014, pp. 588–593.

[39] J. Shu and X. Jia, "Secure task recommendation in crowdsourcing," in *Proc. IEEE Glob. Commun. Conf.*, 2016, pp. 1–6.

[40] J. Shu, X. Liu, X. Jia, K. Yang, and R. H. Deng, "Anonymous privacy-preserving task matching in crowdsourcing," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 3068–3078, Apr. 2018.

[41] J. Shu, K. Yang, X. Jia, X. Liu, C. Wang, and R. H. Deng, "Proxy-free privacy-preserving task matching with efficient revocation in crowdsourcing," *IEEE Trans. Dependable Secur. Comput.*, vol. 18, no. 1, pp. 117–130, Jan. 2021.

ZIHAN LI is currently working toward the undergraduate degree with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing, China. He is currently working with the Research Laboratory of Advanced Network and Data Security, School of Cyberspace Science and Technology, Beijing Institute of Technology. His research interests include applied cryptography and cloud security.

MINGYANG ZHAO received the B.S. degree from the Beijing Institute of Technology, Beijing, China, in 2021. He is currently working toward the master's degree with the School of Cyberspace Science and Technology, Beijing Institute of Technology. His research interests include applied cryptography, cloud security, and blockchain.

GUANYU CHEN is currently working toward the undergraduate degree with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing, China. He is currently with the Research Laboratory of Advanced Network and Data Security, School of Cyberspace Science and Technology, Beijing Institute of Technology. His research interests include applied cryptography and blockchain.

CHUAN ZHANG (Member, IEEE) received the Ph.D. degree in computer science from the Beijing Institute of Technology, Beijing, China, in 2021. From September 2019 to September 2020, he was a Visiting Ph.D. degree with the BBCR Group, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. He is currently an Assistant Professor with the School of Cyberspace Science and Technology, Beijing Institute of Technology. His research interests include secure data services in cloud computing, applied cryptography, machine learning, and blockchain.

TONG WU (Member, IEEE) received the Ph.D. degree in computer science from the University of Wollongong, Wollongong, NSW, Australia, in 2020. She is currently a Postdoctoral Research Fellow with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing, China. Her research interests include applied cryptography, cloud security, and blockchain security.

LIEHUANG ZHU (Senior Member, IEEE) received the Ph.D. degree in computer science from the Beijing Institute of Technology, Beijing, China, in 2004. He is currently a Professor with the School of Cyberspace Science and Technology, Beijing Institute of Technology. His research interests include security protocol analysis and design, group key exchange protocols, wireless sensor networks, and cloud computing.