electronics

MDPI

Article

Enabling Efficient and Privacy-Preserving Task Allocation with Temporal Access Control for Mobile Crowdsensing

Fuyuan Song [1,2], Yiwei Liu [3,*], Siyao Ma [4], Qin Jiang [1], Xiang Zhang [1] and Zhangjie Fu [1,2]

[1] School of Computer Science, Nanjing University of Information Science and Technology, Nanjing 210044, China; fysong@nuist.edu.cn (F.S.); qin1_jiang@outlook.com (Q.J.); zhangxiang@nuist.edu.cn (X.Z.); fzj@nuist.edu.cn (Z.F.)
[2] State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China
[3] Defence Industry Secrecy Examination and Certification Center, Beijing 100089, China
[4] Beijing Urban Construction Design & Development Group Co., Limited, Beijing 100032, China; masiyao@bjucd.com
* Correspondence: yiweiliu_disecc@163.com

Abstract: The increasing proliferation of GPS-enabled mobile devices, including Unmanned Aerial Vehicles (UAVs), smartphones, and laptops, has resulted in a significant upsurge in the outsourcing of spatial data to cloud servers for storage and computation purposes, such as task allocation and location-based services. However, the reliance on untrusted cloud servers introduces the risk of privacy breaches, as these servers possess the ability to deduce and access users' private information based on task content and query requirements. Existing privacy-preserving task-allocation schemes offer only coarse-grained and non-temporal access control, which restricts their applicability in scenarios involving multiple users and time-series data, such as trajectory and time-related routes. To overcome these challenges, this paper proposes an Efficient and Privacy-Preserving Task Allocation with Temporal Access Control (EPTA-T) scheme for mobile crowdsensing. By leveraging the techniques of Gray code and randomizable matrix multiplication, EPTA-T achieves efficient and privacy-preserving task allocation in mobile crowdsensing. Specifically, EPTA-T supports fine-grained and temporal access control through the utilization of an attribute-based access tree and function integration. The formal security analysis demonstrated that EPTA-T effectively guarantees data privacy and query privacy throughout the task allocation process. Extensive experiments conducted using a real-world dataset indicated that the EPTA-T scheme surpassed the performance of the state-of-the-art scheme.

Keywords: privacy-preserving; task allocation; temporal access control; mobile crowdsensing

check for updates

Citation: Song, F.; Liu, Y.; Ma, S.; Jiang, Q.; Zhang, X.; Fu, Z. Enabling Efficient and Privacy-Preserving Task Allocation with Temporal Access Control for Mobile Crowdsensing. *Electronics* **2023**, *12*, 3016. https://doi.org/10.3390/electronics12143016

Academic Editor: Aryya Gangopadhyay

Received: 19 June 2023
Revised: 4 July 2023
Accepted: 5 July 2023
Published: 10 July 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Recently, with the increasing proliferation of task allocation in mobile crowdsensing, it notably enhances the crowdsensing paradigm capability of helping requesters find workers to complete location-aware tasks. This enables requesters to specify task requirements that can be matched with workers based on their respective locations. However, in the mobile crowdsensing environment, the crowdsensing paradigm is inherently untrusted, creating the potential for the capture or inference of task information and workers' locations [1,2]. A prevailing method is to encrypt the spatial data and task content before outsourcing to the crowdsensing paradigm [3], which alleviates the privacy concerns, but incurs limited query functionality and query performance. In addition, the presence of time-related spatial data necessitates the implementation of Temporal Access Control (TAC) in task allocation. For instance, in a Didi Chuxing-based task allocation scenario, if a driver is situated near a requester between the hours of 9:00 and 10:00, the requester can find the driver at any time within that time range, such as 9:30; otherwise, he/she cannot match the driver. Theoretically speaking, a worker's spatial data can only be accessed at the current time

t_c if and only if t_c falls within the worker's time constraint interval $[t_1, t_2]$. Unfortunately, existing task-allocation schemes do not support temporal access control.

Challenge 1: How to design an Efficient and Privacy-Preserving Task Allocation (EPTA) scheme that supports Temporal Access Control (TAC). In mobile crowdsensing, workers' spatial data are typically associated with time-series objects, while requesters' attributes also contain timestamps. To achieve effective and accurate task allocation, TAC needs to be considered and integrated into the mobile crowdsensing system. However, directly combining existing EPTA schemes [1,2,4] with TAC schemes [5,6] may result in privacy breaches and an excessive burden of key management. EPTA and TAC are performed separately. To this end, the cloud server knows which worker meets the task requirements regardless of whether he/she can be accessed, which reveals the worker's timestamp and the worker's location. Moreover, the crowdsensing paradigm can deduce which workers satisfy the task requirements by observing the accessed spatial data, thereby leaking the access pattern. Unfortunately, access pattern leakage can lead to significant privacy issues, such as the disclosure of home addresses, preferences, and behaviors. Additionally, existing approaches have employed attribute-based encryption to enable fine-grained data sharing with access control [7]. However, this introduces computational overhead in terms of data encryption and task allocation. To improve search performance, some symmetric-encryption-based task-allocation schemes [1,8] have been designed. Nevertheless, in multi-user task allocation models, clients are required to share the same secret key, which may lead to key privacy leakage and key management concerns.

Challenge 2: How to provide an arbitrary geometric range query while achieving a fast search. In task allocation, requesters often submit irregular geometric query ranges to find suitable workers located within specific areas for task assignments. However, existing schemes [9,10] only support single or limited geometric ranges, such as circular, rectangular, or triangular ranges. While homomorphic encryption offers a viable approach for secure geometric range queries [11], the computational overhead associated with it is often unacceptable in practical scenarios. To achieve a fast search, Searchable Symmetric Encryption (SSE) [12–14] is commonly employed to enable secure and efficient retrieval of ciphertexts. Unfortunately, SSE fails to support complex query functions such as geometric range queries and lacks flexible key management for multi-user task allocation scenarios. In particular, some works have explored schemes for k-Nearest Neighbor (kNN) queries [15] and skyline queries [16] with privacy guarantees. However, these schemes cannot be directly applied to geometric range queries.

In this paper, we aimed to address the above challenging issues. Specifically, we propose an Efficient and Privacy-Preserving Task Allocation with Temporal Access Control (EPTA-T) scheme for mobile crowdsensing, which efficiently allocates tasks while preserving privacy and incorporating temporal access control. EPTA-T is specifically designed to address the limitations of existing schemes in supporting both task allocation and temporal access control. The main contributions are summarized as follows:

- We utilized Gray code to encode spatial data into vectors. These encoded vectors were then encrypted using randomizable matrix multiplication, enabling us to achieve efficient and privacy-preserving task allocation. This approach significantly reduced the computational overhead associated with data encryption and task allocation, while eliminating the need for key sharing among users.
- To support temporal access control, we leveraged the techniques of function differentiation and function integration. By incorporating these techniques into ciphertext-policy Attribute-Based Encryption (ABE), we constructed an access-tree-based policy. This policy enables precise control over temporal access to the encrypted data. Additionally, the secret involved in the data encryption is treated as the secret of the access policy, ensuring access pattern privacy protection.
- Formal security analysis proves that EPTA-T ensures the confidentiality of data and queries under an Indistinguishability under Selective Chosen-Plaintext Attacks (IND-SCPA) model. The experimental results over a real-world dataset indicated that the

running time of EPTA-T outperformed the state-of-the-art scheme in terms of index encryption, trapdoor generation, and task allocation.

The remainder of this paper is organized as follows. Section 2 presents an overview of related works on privacy-preserving task-allocation schemes. In Section 3, we outline the system model, threat model, design goals, and problem definition of EPTA-T. In Section 4, we provide the necessary background knowledge. Then, in Section 5, we present the detailed construction and technical aspects of the proposed scheme. In Section 6, we prove the security of the proposed scheme. In Section 7, we evaluate and analyze the performance of EPTA-T via experiments. Finally, we conclude our work in Section 8.

2. Related Work

In this section, we mainly focus on the related works on secure task allocation for mobile crowdsensing, including privacy-preserving task allocation and privacy-preserving range query.

2.1. Privacy-Preserving Task Allocation

Currently, privacy-preserving task allocation has received great attention in academia and industry [1,2,4,17,18]. To address the problem of privacy preservation in task allocation, Song et al. [1] proposed an efficient and privacy-preserving task-recommendation scheme by utilizing predicate encryption and randomizable matrix multiplication. Different from randomizable-matrix-based task allocation, Wang et al. [18] utilized a planar Laplace distribution to perturb tasks and workers' locations to protect their location privacy. However, using perturbed locations for task allocation can lead to a decrease in the effectiveness of the task assignments. Thus, Xia et al. [19] presented a probabilistic method to quantify the accessibility between workers and requesters and reduced the effect of the perturbed location on task allocation. To achieve task privacy protection, Zhou et al. [20] designed a tree-based privacy-preserving task-allocation system that addresses the task-allocation problem based on the combinatorial multi-armed slot machine problem. To the balance of user privacy and the usability of task allocation, Wang et al. [18] employed differential privacy to protect location privacy and proposed a multi-task assignment scheme for perturbing the available location information. Xu et al. [21] proposed a privacy-preserving task-allocation scheme using inner-product-based encryption to protect the privacy of task information and workers' locations. However, the schemes in [18–21] do not support temporal access control, i.e., deciding whether the requesters' temporal attributes satisfy a designed time-based access policy.

In summary, how to design an efficient and privacy-preserving task-allocation scheme that supports temporal access control is a crucial issue that needs to be addressed in mobile crowdsensing.

2.2. Privacy-Preserving Range Query

Searchable Symmetric Encryption (SSE) [12–14,22–24] is an effective cryptographic method for secure range query. Based on SSE, Li et al. [25] proposed a tree-aid indistinguishable index structure, which can only support one-dimensional range query. To achieve multi-dimensional range query, several works [1,8] employed an asymmetric scalar-product-preserving encryption approach. For strong security in range query, homomorphic encryption [11,15,26–29] is applied to realize the direct operations over ciphertext, but the computational cost is heavy and unacceptable. Furthermore, Several schemes have explored the limitations and weaknesses of SSE, which significantly balances security and efficiency effectively. Moreover, Wang et al. [9,10] proposed SSE-based schemes for geometric range query, but these schemes have some limitations such as a single geometric query type, inefficient search performance, and a low-dimensional space. To address these issues, Wang et al. [30] proposed a DSSE-based spatial keyword query scheme for dynamic updates. It is worth noting that the above schemes failed to support accurate and efficient geometric range query and temporal access control.

In a word, how to design an efficient and privacy-preserving geometric range query scheme that achieves an arbitrary geometric range query with a fast search is a significant problem that needs to be solved in the spatial data query.

Compared with the existing works, our proposed EPTA-T has three strong aspects: (1) our EPTA-T scheme achieves efficient and privacy-preserving task allocation with temporal access control in mobile crowdsensing; (2) our EPTA-T scheme supports an efficient and privacy-preserving geometric range query with a fast search in a multi-user setting; (3) our EPTA-T scheme guarantees the confidentiality of the data and queries under the IND-SCPA model.

3. Problem Formulation

In this section, we introduce the system model, threat model, design goals, and problem definition of EPTA-T.

3.1. System Model

As shown in Figure 1, the architecture of our mobile crowdsensing system consists of four entities, i.e., a trusted authority ($\mathcal{TA}$), a service provider ($\mathcal{SP}$), multiple workers ($\mathcal{W}$s), and multiple data requesters ($\mathcal{DR}$s).

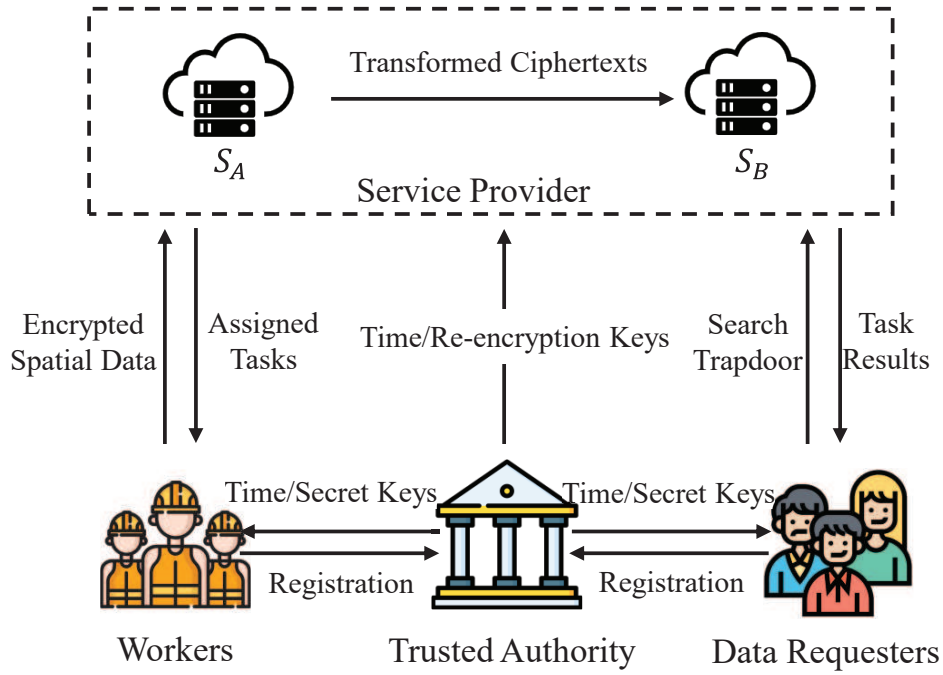


Figure 1. System model.

- Trusted Authority ($\mathcal{TA}$): The $\mathcal{TA}$ is responsible for initializing the system and distributing the secret keys. Initially, the $\mathcal{TA}$ sets up the system to provide registration services for both workers and data requesters and generates encryption keys and re-encryption keys for participating entities.
- Service Provider ($\mathcal{SP}$): The $\mathcal{SP}$ is a crowdsourcing service provider that receives and stores the task content from data requesters. Additionally, the $\mathcal{SP}$ allocates the task requirements to suitable workers. In this model, two non-colluding cloud servers S_A and S_B are introduced to serve as the service provider.
- Workers ($\mathcal{W}s$): Workers encrypt their locations and send the ciphertexts to the service provider. Based on the locations of the workers, the $\mathcal{SP}$ allocates the task requirements to the proper workers. After that, the workers accomplish the task and return the task results to the service provider.

- Data Requesters ($\mathcal{DR}s$): Before publishing the tasks, data requesters encrypt their task requirements. To find suitable workers who satisfy the query constraints, the $\mathcal{DR}s$ generate search trapdoors for their task requirements and submit the search trapdoors along with the encrypted task content to the $\mathcal{SP}$.

In our task allocation model, the $\mathcal{W}$ encrypts his/her spatial data according to the designated access policies, constructs encrypted indexes for task allocation, and transmits both the encrypted spatial data and indexes to the $\mathcal{SP}$. Prior to the $\mathcal{DR}$ releasing a task to the $\mathcal{SP}$, the $\mathcal{DR}$ generates a search trapdoor based on the task requirements and submits it to the $\mathcal{SP}$. Upon receiving the search trapdoor from the $\mathcal{DR}$, the $\mathcal{SP}$ initially converts the $\mathcal{DR}$'s privileged time trapdoor into the trapdoor associated with the current time t_c, but only if the t_c falls within the $\mathcal{DR}$'s privileged time range. Subsequently, the $\mathcal{SP}$ employs EPTA and TAC for task allocation, concurrently verifying whether the $\mathcal{DR}$'s attributes satisfy the access policy and if the search trapdoor corresponds to the appropriate worker index. If these conditions are met, the $\mathcal{SP}$ dispatches the task requirements to the suitable worker. Following the reception of the task results from the selected worker, the $\mathcal{SP}$ returns the results to the $\mathcal{DR}$.

3.2. Threat Model

In our task recommendation, the $\mathcal{TA}$ is fully trusted in the system, and all communications between the $\mathcal{TA}$ and other entities are secure. The $\mathcal{W}$ and $\mathcal{DR}$ are considered to be fully trusted, which means they keep their private keys secret. They would not leak or sell their keys to the $\mathcal{SP}$ for profits. The $\mathcal{SP}$ is considered as semi-honest (i.e., honest-but-curious), which means the $\mathcal{SP}$ will honestly perform the designed protocols to provide the task recommendation services, but the $\mathcal{SP}$ may try to derive private information from encrypted locations and search trapdoors. Based on the information that $\mathcal{SP}$ may derive, we considered the following two attacks:

- *Ciphertext-only attack*: The $\mathcal{SP}$ observes a number of ciphertexts including encrypted locations, encrypted tasks, and search trapdoors, but the $\mathcal{SP}$ cannot obtain their corresponding plaintexts.
- *Chosen-plaintext attack*: Except knowing the encrypted locations and trapdoors, the $\mathcal{SP}$ can oracle access the task recommendation protocols to obtain some plaintext–ciphertext pairs.

Generally, we assumed that there is no collusion between two cloud servers. This assumption is reasonable, as cloud servers are expected to maintain their reputations and protect their own interests [8]. It is worth noting that the no-collusion assumption has been widely adopted in numerous research works within the security community, particularly in the context of the two-server model [8,15,31].

3.3. Design Goals

The design goals of EPTA-T are summarized as follows:

- Privacy protection: The privacy of workers and data requesters should be protected in our EPTA-T scheme. It is not difficult to see that, in task allocation applications that involve more sensitive user data such as workers' locations, requesters' queries, and task results, revealing these data can easily violate user privacy.
- Temporal access control: The EPTA-T scheme should achieve temporal access control for the time-series data (e.g., trajectory data, validity period, and hours of services) in task allocation.
- Efficiency: The computational overhead and communication overhead on the worker and publisher side should be minimized, since the mobile devices are resource-limited. The proposed EPTA-T scheme should achieve efficient data encryption and task allocation.

3.4. Problem Definition

In this paper, our objective was to achieve Efficient and Privacy-Preserving Task Allocation (EPTA) with Temporal Access Control (TAC) for handling massive spatial data comprising temporal and non-temporal attributes. Each spatial datum L_i is encoded as spatial point p_i with a private identity of the worker id_ϱ. The worker's identity id_ϱ is encrypted using a secret key s under an access policy $\mathcal{T}\varrho$, which encompasses both temporal and non-temporal access attributes. Specifically, the spatial data can only be accessed within the valid time period defined by $at[t_1, t_2] \in \mathcal{T}\varrho$. Let IndexEnc be the index encryption algorithm executed by $\mathcal{W}$. Based on the IndexEnc algorithm, each spatial datum L_i is encrypted as $\widehat{C_{I_i}}$ by using the $\mathcal{W}$'s secret key sk_i, i.e., $\mathsf{IndexEnc}(L_i, sk_i) \rightarrow \widehat{C_{I_i}}$. Let IndexTran be the index transformation algorithm executed by the $\mathcal{SP}$. In the IndexTran algorithm, the $\mathcal{SP}$ utilizes the corresponding re-encryption keys RK_A and RK_B to transform the encrypted ciphertext, i.e., $\mathsf{IndexTran}(\widehat{C_{I_i}}, RK_A, RK_B) \rightarrow C_{I_i}$. Let TrapGen be the trapdoor-generation algorithm executed by $\mathcal{DR}$. When the $\mathcal{DR}$ with an attribute set $S = \{at_1, at_2, \cdots, at_i[t'_1, t'_2], \cdots, at_{|S|}\}$ submits a geometric range R_q of task requirements to the $\mathcal{SP}$ at the current time t_c, the search trapdoor $\widehat{T_Q}$ is generated for the geometric range by using the $\mathcal{DR}$'s secret key sk_q, i.e., $\mathsf{TrapGen}(R_q, sk_q) \rightarrow \widehat{T_Q}$. When the $\mathcal{SP}$ receives the search trapdoor $\widehat{T_Q}$ from the $\mathcal{DR}$, using the TrapTran algorithm, the $\mathcal{SP}$ transforms the $\mathcal{DR}$'s privileged time trapdoor into the search trapdoor linked to the current time t_c when the t_c falls within the privileged time range of $\mathcal{DR}$, i.e., $\mathsf{TrapTran}(\widehat{T_Q}, RK_A, RK_B) \rightarrow T_Q$. Upon receiving the transformed trapdoor T_Q, the $\mathcal{SP}$ performs the task allocation to search the proper workers who satisfy the access policy $\mathcal{T}_\varrho(S, t_c) = 1$ and the geometric range query condition simultaneously. Here, $\mathcal{T}_\varrho(S, t_c) = 1$ means the current time t_c belongs to a valid time period of access policy and the access attribute set, and the $\mathcal{DR}$'s attributes satisfy the designed access policy $\mathcal{T}\varrho$, i.e., $(t_c \in [t_1, t_2] \cap [t'_1, t'_2]) \bigwedge (\mathcal{T}_\varrho(S) = 1)$. Let $\mathsf{TAC}(\cdot)$ be the temporal access control function; we have

$$\mathsf{TAC}(\mathcal{T}_\varrho, S, t_c) = \begin{cases} [\![s]\!], & if\ \mathcal{T}_\varrho(S, t_c) = 1; \\ 0, & otherwise, \end{cases} \tag{1}$$

where $[\![s]\!]$ denotes the encrypted secret. Let $\mathsf{EPTA}(\cdot)$ be the efficient and privacy-preserving task allocation primitive function; we have

$$\mathsf{EPTA}(C_{I_i}, T_Q, t_c, [\![s]\!]) = \begin{cases} 1, & if\ tr(C_{I_i} T_Q) = 0; \\ 0, & otherwise. \end{cases} \tag{2}$$

Based on the above problem statement, we give the definition of EPTA-T as follows.

Definition 1 (EPTA-T). *For a worker with the spatial data L_i encrypted under the access policy $\mathcal{T}\varrho$, the worker is considered as an accessible and suitable one for a $\mathcal{DR}$ who possesses an attribute set S and submits a geometric range R_q of task requirements at the current time t_c, if and only if both conditions $\mathsf{TAC}(\mathcal{T}\varrho, S, t_c) = [\![s]\!]$ and $\mathsf{EPTA}(C_{I_i}, T_Q, t_c, [\![s]\!]) = 1$ are met.*

4. Preliminaries

In this section, we review some building blocks that are used in our proposed scheme, including Gray code [6], the access tree [22], and polynomial functions [4].

4.1. Gray Code

The process of converting a binary number $d_1 d_2 \cdots d_n$ into its corresponding binary reflected Gray code involves iterating from right to left. Starting with the rightmost digit d_n, if the preceding digit d_{n-1} is 1, we replace d_n with $1 - d_{n-1}$. If d_{n-1} is 0, we leave d_n unchanged. We then proceed to the next digit d_{n-1} and repeat the process until we reach the first digit d_1, which remains the same as d_0 was assumed to be 0. The resulting sequence $g_1 g_2 \cdots g_n$ represents the binary reflected Gray code. To convert a binary reflected Gray code $g_1 g_2 \cdots g_n$ back to a binary number, we start with the n-th digit and compute the equation as follows:

$$\Sigma_n = \sum_{i=1}^{n-1} g_i (mod 2). \tag{3}$$

If $\Sigma_n = 1$, replace g_n by $1 - g_n$; otherwise, leave it unchanged. Next, we compute

$$\Sigma_{n-1} = \sum_{i=1}^{n-2} g_i (mod 2). \tag{4}$$

By applying the above calculation, we obtain that the binary number $d_1 d_2 \cdots d_n$ corresponds to the initial binary reflected Gray code.

Therefore, the Gray code for $d+1$ bits can be represented as

$$G_{d+1} = (0||g_1, \cdots, 0||g_{2^d}, 1||g_{2^d}, \cdots, 1||g_1),$$

where $\|$ denotes the concatenation operator. For example, $G_1 = (0, 1)$, $G_2 = (00, 01, 11, 10)$. When using Gray code to encode a $d \times d$ grid, the length of each binary code representing a cell is $2 \cdot 2^{\lceil \log_2 d \rceil}$.

4.2. Access Tree

Consider a tree $\mathcal{T}$ that represents an access policy. In this tree, each non-leaf node represents a threshold gate. The node is defined by its children and a threshold value. The threshold value k_x of a node x is such that $0 < k_x \leq \text{num}_x$, where num_x represents the number of children of node x. On the other hand, each leaf node x in $\mathcal{T}$ represents an attribute at_x, where the threshold value k_x is set to 1.

Let $\mathcal{T}_x$ denote the subtree of $\mathcal{T}$ rooted at node x. If x is a non-leaf node, $\mathcal{T}_x(S) = 1$ if and only if it contains at least k_x children x_i $i \in [1, k_x]$ for which $\mathcal{T}_{x_i}(S) = 1$. In other words, $\mathcal{T}_x(S)$ is true if there are enough child nodes satisfying $\mathcal{T}_{x_i}(S) = 1$ according to the threshold value k_x.

If x is a leaf node, $\mathcal{T}_x(S) = 1$ if and only if the attribute at_x is present in the attribute set S. In other words, $\mathcal{T}_x(S)$ is true if the attribute at_x is a member of S.

4.3. Polynomial Function

The polynomial function relates the coefficients of a polynomial to the sums and products of its roots. Let $f(x)$ be a polynomial function of degree n, i.e., $f(x) = a_n x^n + \cdots + a_1 x + a_0$, where the coefficient of x^i is a_i, and $a_n \neq 0$. Based on the property of the polynomial function, we have

$$f(x) = a_n x^n + \cdots + a_1 x + a_0 = a_n (x - x_n) \cdots (x - x_1),$$

where $x_1, \cdots, x_n$ are the roots of $f(x)$. According to the construction of the polynomial function $f(x)$, we can calculate any subproducts of the roots. For any $i \in [0, n]$, the $(n-i)$-th coefficient a_{n-i} is associated with a signed sum of all possible subproducts of the roots, as shown in Equation (5).

$$\sum_{1 \leq k_1 \leq \cdots \leq k_i \leq n} x_{k_1} x_{k_2} \cdots x_{k_i} = (-1)^i \frac{a_{n-i}}{a_n}. \tag{5}$$

5. The Proposed Scheme

In this section, we first introduce the overview of EPTA-T. After that, we give the detailed construction of EPTA-T. In our EPTA-T, it mainly consists of seven algorithms: System Setup (SystSetup), Key Generation (KeyGen), Index Encryption (IndexEnc), Index Transformation (IndexTran), Trapdoor Generation (TrapGen), Trapdoor Transformation (TrapTran), and task allocation (Query). Finally, we analyzed the correctness of EPTA-T.

5.1. Overview

Since EPTA-T supports secure task allocation in multi-worker multi-requester settings without key sharing, the main idea of EPTA-T is to encode and encrypt the index of the

spatial data and the search trapdoor of the task requirement into a matrix-trace-based matching operation, such that TAC can be implemented by the same encryption mechanism without involving additional computational overhead.

Transforming the geometric-range-based task allocation into the inner-product-based matching operation. To enable task allocation based on the geometric ranges of the task requirements, the space is divided into uniform cells with dimensions $L \times L$. The task allocation conditions, specifying the spatial data of workers located within the geometric range defined by the data requester, are encoded using Gray code, where 0 and 1 are represented by two positive integers, X and Y, respectively. Each spatial datum L_i is encoded as spatial point p_i by using Gray code and further expanded into a spatial vector $\overrightarrow{\mathsf{p}_i}$ by padding it with additional entries, including -1. The geometric query range R_q is encoded as the aggregated query code q_i, which represents the combination of the codes for the cells encompassed by the geometric range R_q. Subsequently, the aggregated range query vector $\overrightarrow{\mathsf{q}_i}$ is generated by padding the query code with additional entries, including the sum of the squares of the code values. Specifically, the aggregated query code is generated by replacing the different entries with 0 to reduce the scale of codes in the query range. Consequently, if the inner product between the spatial vector $\overrightarrow{\mathsf{p}_i}$ and the query vector $\overrightarrow{\mathsf{q}_i}$ is 0 (i.e., $\overrightarrow{\mathsf{p}_i} \circ \overrightarrow{\mathsf{q}_i} = 0$), this indicates that the spatial data L_i of the worker are located within the geometric query range R_q. Otherwise, if the inner product is non-zero, this signifies that the spatial data are located outside the query range.

As shown in Figure 2, we give an example of the encoding procedure and geometric range query. In Figure 2, the spatial space is uniformly split into 4×4 cells. The spatial point p_i is encoded as $g_{\mathsf{p}_i} = YYXX$ and further extended as spatial vector $\overrightarrow{\mathsf{p}_i} = (Y, Y, X, X, -1)$. In addition, the geometric query range (green domain) is encoded as $g_{\mathsf{q}_1} = XYXX \vee XYXY \vee YYXX \vee YYXY = 0YX0$ and $g_{\mathsf{q}_2} = YYYY$. After that, the corresponding query vectors are further extended as $\overrightarrow{\mathsf{q}_1} = (0, Y, X, 0, Y^2 + X^2)$ and $\overrightarrow{\mathsf{q}_2} = (Y, Y, Y, Y, 4Y^2)$, respectively. Therefore, the inner product between $\overrightarrow{\mathsf{p}_i}$ and $\overrightarrow{\mathsf{q}_1}$ is 0, which means that the spatial point p_i is located in the geometric query range R_{q_1}.

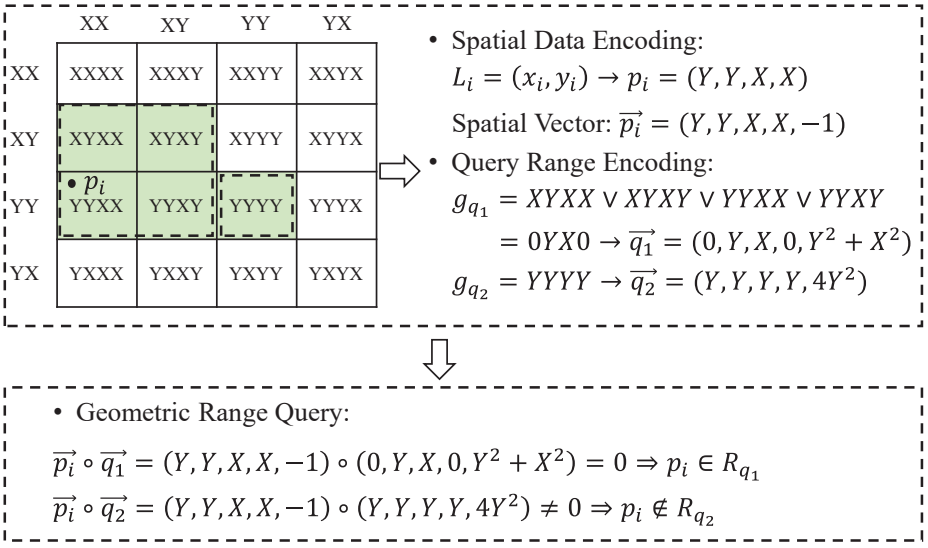


Figure 2. An Encoding Example of Spatial Data and Query Range and Geometric Range Query.

Secure matrix-trace-based task allocation mechanism. In EPTA-T, we encrypt the spatial data L_i as $\widehat{C_{I_i}}$ and transform them as C_{I_i} by calling IndexEnc and IndexTran, respectively. In addition, we encrypt the geometric query range R_q of task allocation as $\widehat{T_Q}$ and transform it as T_Q by calling TrapGen and TrapTran, respectively. After that, we call the Query protocol to calculate the matrix trace (i.e., $tr(C_{I_i} T_Q)$) to retrieve proper workers for task allocation. If the worker $\mathcal{W}_i$ is the proper one, his/her spatial data are located in the geometric range of the task requirement, i.e., $L_i \in R_q$, then $tr(C_{I_i} T_Q) = 0$.

Secure temporal access control. To achieve EPTA with TAC, the key idea is to perturb the search results of EPTA with the access policy by employing a secret used in IndexEnc as the secret of the access policy to encrypt the corresponding spatial data. To this end, privacy-preserving task allocation is correctly performed only when the spatial data are accessible.

To achieve temporal access control, we add time validity to a secret associated with the spatial data and correlate the encrypted secret with an access time period. Only when a data requester has attributes that satisfy the defined access policy and a secret that corresponds to the accessible time period can the data requester access the spatial data.

For TAC, we constructed a hierarchical temporal access tree $\mathcal{T}$ and utilized its leaf nodes to denote different time attributes $at[t_1, t_2]$. In addition, when the $\mathcal{DR}$ can access the spatial data, the $\mathcal{DR}$'s privileged time period $at[t_1', t_2'] \in S$ should be satisfied. If the current time t_c satisfies $t_c \in [t_1, t_2] \cap [t_1', t_2']$, we utilize the polynomial function property to transform the time-related derivation function $f(t)$ associated with $[t_1, t_2]$ and $[t_1', t_2']$ into the original function $F(t)$ related to t_c. If $at_x \in S$, we calculate the original function $F_x(t)$ as shown in Equation (6).

$$F_x(t)|_{t=t_c} = a \int_{t_1}^{t_2} \int_{t_1'}^{t_2'} f_x(t) dt dt' = e(g_p, H)^{a y_x(0)}, \tag{6}$$

where a represents a unique entry used to differentiate between different data requesters ($\mathcal{DR}$) selected by the $\mathcal{TA}$, x is a leaf node of the hierarchical temporal access tree, and $y_x(0)$ is the secret share of s for the leaf node x. The entry s is regarded as the secret of the hierarchical temporal access tree $\mathcal{T}$'s root node, which is used to encrypt the private identity of the worker id_i. When the $\mathcal{DR}$'s attributes satisfy the defined access policy at the current time t_c, we have $F_{root} = e(g_p, H)^{as}$.

In our EPTA-T, $\mathcal{SP}$ should first execute TAC by checking whether the $\mathcal{DR}$'s attributes satisfy the defined access policy tree $\mathcal{T}$ at the current time t_c. If the $\mathcal{SP}$ calculates that $F_{root} = e(g_p, H)^{as}$ holds, this indicates that the $\mathcal{DR}$ can access the spatial data. After that, the $\mathcal{SP}$ performs EPTA to calculate $tr(C_{I_i} T_Q)$. If $tr(C_{I_i} T_Q) = 0$ holds, the accessible spatial data satisfy the geometric query range, and the $\mathcal{SP}$ selects the worker identity id_ϱ as the proper worker and sends the task to this target one.

Remark 1. *To the best of our knowledge, EPTA-T is the first work to support fine-grained and temporal access control in privacy-preserving task allocation, which achieves efficient geometric range query and IND-SCPA security in the multi-worker multi-requester setting without key sharing.*

5.2. The Detailed Construction of EPTA-T

Based on the above overview, we describe the detailed construction of EPTA-T:

(1) **SystSetup** $(1^\lambda) \rightarrow (pp, msk)$. Given a security parameter λ, the $\mathcal{TA}$ generates the master key msk, which consists of two $(n+1) \times (n+1)$ random invertible matrices $\{M_1, M_2\}$, a hash function $h : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$, which hashes any spatial vector and query vector to positive values in $\mathbb{Z}_p^*$ and converts the attribute string to a group element, respectively, a random symmetric key K, and a random permutation $\pi : \mathbb{Z}_p^{n+1} \rightarrow \mathbb{Z}_p^{n+1}$. In addition, $\mathcal{TA}$ generates a composite order bilinear system $\mathbb{S} = (\mathbb{G}, \mathbb{G}_T, e)$ of order $n = p \times r$ and generators $g_p \in \mathbb{G}_p$, $g_r \in \mathbb{G}_r$. Additionally, the $\mathcal{TA}$ randomly selects an entry $\theta \in \mathbb{Z}_p$ and calculates $H = g_p^\theta g_r^\theta$. Namely, the masker secret key can be denoted as $msk = \{M_1, M_2, \pi, h\}$, and the public parameters are generated as $pp = \{\mathbb{S}, g_p, g_r, h, H\}$.

(2) **KeyGen** $(msk, \mathcal{W}_i, \mathcal{DR}_q) \rightarrow (sk_i, sk_q, RK_A, RK_B)$. For a worker $\mathcal{W}_i$, the $\mathcal{TA}$ first chooses two random $(n+1) \times (n+1)$ matrices $\{M_{i,1}, M_{i,2}\}$. Then, $\mathcal{TA}$ computes $M_{i,1}' = M_1 M_{i,1}^{-1}$ and $M_{i,2}' = M_1 M_{i,2}^{-1}$. For a data requester $\mathcal{DR}_q$, the $\mathcal{TA}$ randomly selects two $(n+1) \times (n+1)$ $\{M_{q,1}, M_{q,2}\}$, and then, the $\mathcal{TA}$ computes $M_{q,1}' = M_{q,1}^{-1} M_1^{-1}$ and $M_{q,2}' = M_2^{-1} M_{q,2}^{-1}$. Finally, the $\mathcal{TA}$ sends the secret key $sk_i = \{M_{i,1}, M_{i,2}, \pi, h\}$ to the worker $\mathcal{W}_i$ and the secret key $sk_q = \{M_{q,1}, M_{q,2}, \pi, h, K\}$ to the data requester

$\mathcal{DR}_q$, and the $\mathcal{TA}$ distributes the re-encryption key $RK_A = \{M'_{i,1}, M'_{q,2}\}$ to S_A and the re-encryption key $RK_B = \{M'_{i,2}, M'_{q,1}\}$ to S_B. To distinguish different $\mathcal{DR}$s, the $\mathcal{TA}$ chooses a unique element $a \in \mathbb{Z}_p$ for a certain $\mathcal{DR}$'s attribute set S.

(3) **IndexEnc** $(L_i, sk_i) \rightarrow \widehat{C_{I_i}}$. Given a worker $\mathcal{W}_i$'s spatial data $L_i = (x_i, y_i)$, $\mathcal{W}_i$ encrypts his/her spatial data and outsources it to $\mathcal{SP}$ as follows:

Step 1: Firstly, the $\mathcal{W}_i$ encodes the spatial data L_i as a spatial vector $\overrightarrow{\mathsf{p}_i} = (b_1, b_2, \cdots, b_d) \leftarrow Gray(L_i)$, where b_i $(i \in [1, d-1])$ is a value of X or Y, and $b_d = -1$.

Step 2: For the spatial vector $\overrightarrow{\mathsf{p}_i} = (b_1, b_2, \cdots, b_d)$, the $\mathcal{W}_i$ transforms $\overrightarrow{\mathsf{p}_i}$ to a positive integer $v_i = \sum_{l=1}^{d} b_l \cdot 2^{d-l}$. After that, the workers have a collection of integers, i.e., $\{v_1, v_2, \cdots, v_m\}$, where m denotes the number of workers in the mobile crowdsensing system.

Step 3: To protect the privacy of the transformative integer v_i, the $\mathcal{W}_i$ maps the integer with the one-way hash function h and obtains the hash value $h(v_i)$. Then, the $\mathcal{W}_i$ calculates different powers for the hash value $h(v_i)$ to generate an $(n+1)$-dimensional vector $I_i = (h(v_i)^0, h(v_i)^1, \cdots, h(v_i)^n)$.

Step 4: Next, the $\mathcal{W}_i$ generates a one-time random positive number α and embeds it into the vector I_i. That is, the $\mathcal{W}_i$ obtains the random vector $\overline{I_i} = (\alpha h(v_i)^0, \alpha h(v_i)^1, \cdots, \alpha h(v_i)^n)$.

Step 5: After that, the $\mathcal{W}_i$ permutes the random vector $\overline{I_i}$ to another vector $\widehat{I_i}$ by using the permutation π, i.e., $\widehat{I_i} = \pi(\overline{I_i})$. Then, the $\mathcal{W}_i$ transforms the permuted vector $\widehat{I_i}$ to a corresponding diagonal matrix $\widehat{D_{I_i}}$ with the diagonal being $\widehat{I_i}$.

Step 6: Finally, the $\mathcal{W}_i$ generates a random $(n+1) \times (n+1)$ upper triangular matrix $\mathcal{U}_i$, where the main diagonal entries are 1, and the remainder of the non-zero entries are one-time random values. Then, the $\mathcal{W}_i$ uses his/her secret key sk_i to encrypt $\widehat{D_{I_i}}$ as shown in Equation (7). After this operation, the $\mathcal{W}_i$ sends the encrypted location $\widehat{C_{I_i}}$ to S_A.

$$\widehat{C_{I_i}} = M_{i,1} \mathcal{U}_i \widehat{D_{I_i}} M_{i,2} \tag{7}$$

Specifically, to protect identity privacy, the worker $\mathcal{W}_i$ encrypts the identity id_ϱ as c_ϱ via the standard encryption algorithm (e.g., AES) and encrypts the hierarchical temporal access tree as follows:

- Firstly, the worker $\mathcal{W}_i$ selects the entry s as the secret of the root node of the hierarchical temporal access tree $\mathcal{T}_\varrho$.
- Then, the worker $\mathcal{W}_i$ calculates the secret shares of s for each leaf node x as $y_x(0)$.
- Finally, for each leaf node x, if x is represented as a temporal attribute $at_x[t_1, t_2]$, the $\mathcal{W}_i$ splits $y_x(0)$ into $y_x(0)', y_x(0)''$, which should guarantee that $y_x(0) = y_x(0)' + y_x(0)''$ holds. Thus, $\mathcal{W}_i$ encrypts the leaf node x as $C_x = (H^{y_x(0)'}, h(at_x)^{y_x(0)'}, H^{y_x(0)''}, h(at_x)^{y_x(0)''})$; otherwise, $\mathcal{W}_i$ encrypts it as $C_x = (g_p^{y_x(0)}, h(at_x)^{y_x(0)})$.

To achieve secure task allocation, $\mathcal{W}_i$ outsources the ciphertexts $(c_\varrho, \widehat{C_{I_i}}, C_x)$ to S_A.

(4) **IndexTran** $(c_\varrho, \widehat{C_{I_i}}, C_x, RK_A, RK_B) \rightarrow C_{I_i}$. Upon receiving the encrypted index $\widehat{C_{I_i}}$, S_A uses the corresponding re-encryption key $M'_{i,1}$ to re-encrypt $\widehat{C_{I_i}}$ as shown in Equation (8).

$$\begin{aligned} \widetilde{C_{I_i}} &= M'_{i,1} \widehat{C_{I_i}} \\ &= M'_{i,1} M_{i,1} \mathcal{U}_i \widehat{D_{I_i}} M_{i,2} \\ &= M_1 \mathcal{U}_i \widehat{D_{I_i}} M_{i,2}. \end{aligned} \tag{8}$$

Then, S_A sends $\widetilde{C_{I_i}}$ to S_B. After receiving converted ciphertext $\widetilde{C_{I_i}}$, S_B re-encrypts it with the re-encryption key $M'_{i,2}$ as shown in Equation (9).

$$\begin{aligned} C_{I_i} &= \widetilde{C_{I_i}} M'_{I,2} \\ &= M_1 \mathcal{U}_i \widehat{D_{I_i}} M_{i,2} M'_{i,2} \\ &= M_1 \mathcal{U}_i \widehat{D_{I_i}} M_2. \end{aligned} \tag{9}$$

Finally, S_B stores the converted ciphertext C_{I_i}, the encrypted identity c_ϱ, and the encrypted access tree C_x:

(5) **TrapGen** $(R_q, sk_q) \rightarrow \widehat{T_Q}$. The data requester $\mathcal{DR}_q$ designated the geometric range of the task requirement as $R_q = (R_{q,ll}, R_{q,lu}, R_{q,rl}, R_{q,ru})$, where ll, lu, rl, ru denote left-lower, left-upper, right-lower, and right-upper of the geometric range bound, respectively. Then, the $\mathcal{DR}_q$ encodes geometric query range R_q into the query vector $\overrightarrow{q}$ based on the Gray code. After that, the data requester $\mathcal{DR}_q$ performs the following operations to generate the search trapdoor:

Step 1: Firstly, $\mathcal{DR}_q$ encodes the geometric range R_q of the task requirement to query vector $\overrightarrow{q}$ as shown in Equation (10) by using the Gray code method.

$$\overrightarrow{q} = (q_1, q_2, \cdots, q_d), \tag{10}$$

where q_i $(i \in [1, d-1])$ is one of X, Y, or 0 and $q_d = \sum_{i=1}^{d-1} q_i^2$.

Step 2: Secondly, to protect the privacy of the query vector, the $\mathcal{DR}_q$ transforms the query vector $\overrightarrow{q}$ into an n-dimensional vector $v_q = (v_{q,1}, v_{q,2}, \cdots, v_{q,n})$ by using the same transformation $v_{q,i} = \sum_{i=1}^{d} q_i \cdot 2^{d-i}$ as mentioned in IndexEnc, where each entry $v_{q,i}$ $(i \in [1, n])$ is a positive integer value.

Step 3: After that, the $\mathcal{DR}_q$ hashes the integer vector v_q as shown in Equation (11) by using the hash function h.

$$h(v_q) = (h(v_{q,1}), h(v_{q,2}), \cdots, h(v_{q,n})). \tag{11}$$

Step 4: To further protect the privacy of the mapping-based task requirement, the $\mathcal{DR}_q$ constructs a polynomial function $f(q)$ of the task requirement with the degree being n, as shown in Equation (12).

$$\begin{aligned} f(q) &= \prod_{i=1}^{n} (q - h(v_{q,i})) \\ &= a_0 + a_1 q + \cdots + a_n q^n. \end{aligned} \tag{12}$$

Step 5: Subsequently, the $\mathcal{DR}_q$ extracts the coefficients of the task function to construct the coefficient vector as $Q = (a_o, a_1, \cdots, a_n)$. Then, the $\mathcal{DR}_q$ embeds a one-time random positive number β into the coefficient vector Q and generates the random vector $\overline{Q} = (\beta a_0, \beta a_1, \cdots, \beta a_n)$.

Step 6: Next, the $\mathcal{DR}_q$ permutes the random vector $\overline{Q}$ as $\widehat{Q} = \pi(\overline{Q})$, and then, the $\mathcal{DR}_q$ transforms $\widehat{Q}$ into the corresponding diagonal matrix $\widehat{D_Q}$ with the diagonal being $\widehat{Q}$.

Step 7: Finally, the $\mathcal{DR}_q$ generates a random $(n+1) \times (n+1)$ upper triangular matrix $\mathcal{P}_q$ with the main diagonal being $(1, 1, \cdots, 1)$ and encrypts the diagonal matrix $\widehat{D_Q}$ by using the secret key sk_q and the $\mathcal{DR}_q$ as shown in Equation (13). After that, the $\mathcal{DR}_q$ sends the encrypted task $Enc_K(\mathcal{T})$ and search trapdoor $\widehat{T_Q}$ to S_A, where Enc denotes symmetric encryption (e.g., AES), and $\mathcal{T}$ is the plaintext of $\mathcal{DR}_q$'s task.

$$\widehat{T_Q} = M_{q,2} \widehat{D_Q} \mathcal{P}_q M_{q,1}. \tag{13}$$

(6) **TrapTran** $(\widehat{T_Q}, RK_A, RK_B) \rightarrow T_Q$. Upon receiving the search trapdoor $\widehat{T_Q}$, S_A utilizes the corresponding re-encryption key $M'_{q,2}$ to re-encrypt $\widehat{T_Q}$, as shown in Equation (14).

$$\begin{aligned} \widetilde{T_Q} &= M'_{q,2} \widehat{T_Q} \\ &= M'_{q,2} M_{q,2} \widehat{D_Q} \mathcal{P}_q M_{q,1} \\ &= M_2^{-1} \widehat{D_Q} \mathcal{P}_q M_{q,1}. \end{aligned} \tag{14}$$

After that, S_A sends $\widetilde{T_Q}$ to S_B. Subsequently, S_B finds the requester $\mathcal{DR}_q$'s re-encryption key $M'_{q,1}$ and re-encrypts $\widetilde{T_Q}$ to T_Q as shown in Equation (15).

$$\begin{aligned} T_Q &= \widetilde{T_Q} M'_{q,1} \\ &= M_2^{-1} \widehat{D_Q} \mathcal{P}_q M_{q,1} M'_{q,1} \\ &= M_2^{-1} \widehat{D_Q} \mathcal{P}_q M_1^{-1}. \end{aligned} \tag{15}$$

(7) **Query** $(c_\varrho, C_x, t_c, C_{I_i}, T_Q) \rightarrow \mathcal{R}_i$. Upon receiving the search trapdoor T_Q, for each encrypted object $\mathsf{EO} = (c_\varrho, C_x, C_{I_i})$, S_B first checks whether the $\mathcal{DR}_q$'s attribute set S satisfies the defined access policy $\mathcal{T}_\varrho$ based on Definition 1. If not, S_B outputs $\perp$; otherwise, S_B executes the following Query procedure for task allocation.

Step 1: For each leaf node x in hierarchical temporal access tree $\mathcal{T}_\varrho$, if $at_x \notin S$, S_B outputs $\perp$. If $at_x \in S$ and $t_c \in [t_1, t_2] \cap [t'_1, t'_2]) \bigwedge (\mathcal{T}_\varrho(S) = 1)$, S_B utilizes the ciphertext C_x and time-related derivation function to calculate $F_x = a \int_{t_1}^{t_2} \int_{t'_1}^{t'_2} f_x(t) dt dt' = e(g_p, H)^{a y_x(0)}$; otherwise, S_B outputs $\mathsf{TAC}(\mathcal{T}_\varrho, S, t_c) = 0$. By doing this, we have $F_{root} = e(g_p, H)^{as}$ in the root node of the temporal access tree.

Step 2: Then, S_B checks whether the worker $\mathcal{W}_i$ is a suitable one via the corresponding matrix trace between the $\mathcal{W}_i$'s index C_{I_i} and $\mathcal{DR}_q$'s search trapdoor T_Q, which is equivalent to checking whether the inner product of spatial vector $\overrightarrow{\mathsf{p}_i}$ and query vector $\overrightarrow{\mathsf{q}}$ is equal to 0. By doing this, S_B can determine whether the spatial data L_i of the $\mathcal{W}_i$ are located in the geometric query range R_q of the task requirement by calculating the trace $tr(C_{I_i} T_Q)$ as shown in Equation (16).

$$\begin{aligned} tr(C_{I_i} T_Q) &= tr(M_1 \mathcal{U}_i \widehat{D_{I_i}} M_2 M_2^{-1} \widehat{D_Q} \mathcal{P}_q M_1^{-1}) \\ &= tr(M_1 \mathcal{U}_i \widehat{D_{I_i}} \widehat{D_Q} \mathcal{P}_q M_1^{-1}) \\ &= \overline{I_i} \circ \overline{Q} \\ &= \alpha\beta \Big(a_0 h(v_i)^0 + a_1 h(v_i)^1 + \cdots + a_n h(v_i)^n \Big). \end{aligned} \tag{16}$$

If $tr(C_{I_i} T_Q) = 0$ holds, the worker $\mathcal{W}_i$'s spatial data L_i are located in the geometric query range R_q of the task requirement. Thus, $\mathsf{EPTA}(C_{I_i}, T_Q, t_c, C_x, c_\varrho) = \mathcal{R}_i = 1$ holds, which means the $\mathcal{W}_i$ is an accessible and proper worker at the current time t_c and adds $\{c_\varrho, F_{root}\}$ to the search result $\mathcal{R}$. Otherwise, S_B chooses another re-encrypted spatial datum and performs the above secure task allocation procedure until all encrypted spatial data are searched.

5.3. Correctness Analysis

From linear algebra, we have the following theorem about the correctness of EPTA-T.

Theorem 1 (Correctness of EPTA-T)**.** In EPTA-T, $\mathsf{EPTA\text{-}T.Query}(C_{I_i}, T_Q, t_c, C_x, c_\varrho) \rightarrow \mathcal{R}_i = 1$ is correct if and only if both $F_{root} = e(g_p, H)^{as}$ and $tr(C_{I_i} T_Q) = 0$ are satisfied.

Proof. To demonstrate the correctness of EPTA-T, we should prove that both the data requester $\mathcal{DR}_q$'s attribute set S satisfies the temporal access policy $\mathcal{T}_\varrho$ at the current time t_c and the worker $\mathcal{W}_i$'s location L_i is located in the geometric range R_q of the task requirement when $tr(C_{I_i} T_Q) = 0$ are satisfied.

For the temporal access control constraint, since $y_x(0) = y_x(0)' + y_x(0)''$ and $C_x = (H^{y_x(0)'}, h(at_x)^{y_x(0)'}, H^{y_x(0)''}, h(at_x)^{y_x(0)''})$, we have Equations (17) and (18).

$$F'_x = \frac{e(g_p^a h(at_x)^a, H^{y_x(0)'})}{e(H^a, h(at_x)^{y_x(0)'})} = e(g_p, H)^{a y_x(0)'}. \tag{17}$$

$$F''_x = \frac{e(g_p^a h(at_x)^a, H^{y_x(0)''})}{e(H^a, h(at_x)^{y_x(0)''})} = e(g_p, H)^{a y_x(0)''}. \tag{18}$$

Based on Equations (17) and (18), we can calculate that $F_x = F'_x \cdot F''_x = e(g_p, H)^{ay_x(0)}$. Similarly, for the non-leaf node x in $\mathcal{T}_\varrho$, we also can calculate F_x in a recursive way. To this end, we have $F_root = e(g_p, H)^{as}$, which means the $\mathcal{DR}_q$'s attribute set S satisfies the defined access policy $\mathcal{T}_\varrho$ at the current time t_c. Therefore, the correctness of TAC is demonstrated.

For the EPTA constraint, note that $C_{I_i} T_Q = M_1 \mathcal{U}_i \widehat{D_{I_i}} \widehat{D_Q} \mathcal{P}_q M_1^{-1}$. According to the similarity transformation in linear algebra, we have $tr(C_{I_i} T_Q) = tr(\mathcal{U}_i \widehat{D_{I_i}} \widehat{D_Q} \mathcal{P}_q)$. Since $\mathcal{U}_i$ and $\mathcal{P}_q$ are upper triangular matrices with the main diagonals being $(1, 1, \cdots, 1)$, both $\mathcal{U}_i \widehat{D_{I_i}}$ and $\widehat{D_Q} \mathcal{P}_q$ are upper triangular matrices with the main diagonals being $\widehat{I}_i$ and $\widehat{Q}$. Additionally, by using the same permutation π, $\overline{I_i}$ and $\overline{Q}$ are permuted to $\widehat{I}_i$ and $\widehat{Q}$, respectively. Therefore, we have

$$\begin{aligned} tr(C_{I_i} T_Q) &= \widehat{I}_i \circ \widehat{Q} \\ &= \overline{I_i} \circ \overline{Q} \\ &= \alpha\beta \Big(a_0 h(v_i)^0 + a_1 h(v_i)^1 + \cdots + a_n h(v_i)^n \Big), \end{aligned} \tag{19}$$

where $\widehat{I}_i \circ \widehat{Q}$ denotes the inner product between $\widehat{I}_i$ and $\widehat{Q}$. In Equation (19), since both α and β are two one-time positive values, if $tr(C_{I_i} T_Q) = 0$ holds, the equation $a_0 h(v_i)^0 + a_1 h(v_i)^1 + \cdots + a_n h(v_i)^n = 0$ is satisfied, which means the hash value $h(v_i)$ is the root of the polynomial function $f(q)$ constructed by the requester $\mathcal{DR}_q$. Since $h(v_i)$ and $h(v_q)$ are derived from the spatial vector $\overrightarrow{\mathsf{p}_i}$ and $\overrightarrow{\mathsf{q}}$, we have $\overrightarrow{\mathsf{p}_i} \circ \overrightarrow{\mathsf{q}} = 0$. That is, the $\mathcal{W}_i$'s spatial data L_i are located in the geometric query range R_q of the task requirement, i.e., $\mathcal{R}_i = 1$. Therefore, the correctness of EPTA-T is demonstrated. □

Discussion: Our EPTA-T achieves secure task allocation and temporal access control simultaneously. By using randomizable matrix multiplication, EPTA-T can resist the IND-SCPA in multi-user setting. In addition, EPTA-T supports temporal access control by leveraging function differentiation and integration, along with CP-ABE, to regulate access to tasks over time. Our EPTA-T considers the locations of the workers, which is not sufficient for accurate task allocation. In practical task allocation scenarios, workers' keywords are also significant for task matching. To provide a detailed discussion on the advantages and challenges of EPTA-T, we discuss both the functionality and efficiency of EPTA-T as follows:

- Functionality: Our EPTA-T scheme can support task allocation in multi-user settings without key sharing. This is made possible by leveraging proxy re-encryption, which allows the transformation of ciphertexts from an asymmetric key environment to re-encrypted ciphertexts in a symmetric key setting. As a result, even when workers and requesters employ different keys, secure task allocation can still be achieved by utilizing different types of ciphertexts.
- Efficiency: Our EPTA-T scheme provides significant reductions in computational overhead for both the data encryption and task allocation processes. By utilizing randomizable matrix multiplication and matrix decomposition techniques, our EPTA-T scheme avoids heavy cryptographic operations such as fully homomorphic encryption, Paillier encryption, exponential operations, and secure circuits. This design choice leads to a notable improvement in the performance of data encryption and task allocation, as the computational burden is reduced.

6. Security Analysis

Since CPA is stronger than COA, if EPTA-T can resist CPA, it logically follows that it can also provide resistance against COA. Therefore, in this section, we mainly analyzed how EPTA-T can protect the confidentiality of spatial data, workers' identity, the geometric query of the task requirements, and the task results under an Indistinguishability under Selective Chosen-Plaintext Attacks (IND-SCPA) model. Before demonstrating the security of EPTA-T under the IND-SCPA model, we give the definitions of the secure simulation-

based experiment between a computationally bounded adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ as follows.

Definition 2 (IND-SCPA data privacy of EPTA-T). *Let $\prod$ be a privacy-preserving task-allocation scheme with temporal access control over a security parameter λ. A security game of data privacy between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ is defined as follows:*

- *Initialization: Given a security parameter λ, $\mathcal{A}$ generates two spatial databases $DB_0 = \{L_{0,1}, L_{0,2}, \cdots, L_{0,m}\}$, and $DB_1 = \{L_{1,1}, L_{1,2}, \cdots, L_{1,m}\}$, and sends them to $\mathcal{C}$.*
- *Setup: $\mathcal{C}$ performs the Setup algorithm and theKeyGen algorithm to generate a master key, public parameter, secret key, and re-encryption key and sends the public parameter to $\mathcal{A}$*
- *Phase 1: $\mathcal{A}$ adaptively submits several encrypted data and trapdoor requests, where each request is one of the following types:*
  1. ***Encrypted data request**: Upon the j-th encrypted data request, $\mathcal{A}$ outputs a spatial dataset $L'_j = \{L'_{j,1}, L'_{j,2}, \cdots, L'_{j,m}\}$. $\mathcal{C}$ responds with encrypted spatial data $\widehat{C'_{I_i}} \leftarrow \mathsf{IndexEnc}(L'_j, sk_i)$.*
  2. ***Search trapdoor request**: Upon the j-th search trapdoor request, $\mathcal{A}$ generates a geometric range query R_q and sends it to $\mathcal{C}$. After that, $\mathcal{C}$ responds with a search trapdoor $T_{Q_j} = \mathsf{TrapGen}(R_{q_j}, sk_q)$. Here, R_{q_j} is subject to $\mathcal{L}(DB_0, R_{q_j}) = \mathcal{L}(DB_1, R_{q_j})$, where $\mathcal{L}$ is a leakage function obtaining all the information leaked during the task allocation process:*
- *Challenge: With DB_0 and DB_1 selected from the initialization, $\mathcal{C}$ flips a coin $b \in \{0,1\}$, executes $\mathsf{IndexEnc}$ to calculate the encrypted spatial data $\widehat{C'_{I_i}}$, and sends them to $\mathcal{A}$.*
- *Phase 2: $\mathcal{A}$ runs Phase 1 again and adaptively selects several spatial data, then issues them to $\mathcal{C}$.*
- *Guess: $\mathcal{A}$ outputs his/her guess b' of b. If $b' = b$ holds, $\mathcal{A}$ wins the security game; otherwise, $\mathcal{A}$ fails.*

$\prod$ is secure against IND-SCPA on data privacy if, for any polynomial-time adversary $\mathcal{A}$ in the above security game, $\mathcal{A}$ has at most a negligible advantage as follows:

$$Adv_{\prod,\mathcal{A}}^{IND-SCPA,Data} = |Pr(b' = b) - \frac{1}{2}| \leq negl(\lambda),$$

where $negl(\lambda)$ is a negligible function.

Definition 3 (IND-SCPA query privacy of EPTA-T). *Let $\prod$ be a privacy-preserving task-allocation scheme with temporal access control over a security parameter λ. A security game of query privacy between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ is defined as follows:*

- *Initialization: $\mathcal{A}$ submits two geometric range queries R_{q_0} and R_{q_1} with the same dimensional space to $\mathcal{C}$.*
- *Setup: $\mathcal{C}$ runs the KeyGen algorithm to generate a secret key for trapdoor encryption.*
- *Phase 1: $\mathcal{A}$ adaptively submits a series of requests to $\mathcal{C}$. After that, $\mathcal{C}$ responds with encrypted spatial data and trapdoor. Each request is one of the following two types:*
  1. ***Encrypted data request**: Upon the j-th encrypted data request, $\mathcal{A}$ outputs a spatial dataset $L'_j = \{L'_{j,1}, L'_{j,2}, \cdots, L'_{j,m}\}$. Then, $\mathcal{C}$ responds with encrypted spatial data $\widehat{C'_{I_i}} = \mathsf{IndexEnc}(L'_j, sk_i)$, where $\widehat{C'_{I_i}}$ is subject to $\mathcal{L}(\widehat{C'_{I_i}}, R_{q_0}) = \mathcal{L}(\widehat{C'_{I_i}}, R_{q_1})$.*
  2. ***Search trapdoor request**: Upon the j-th trapdoor request, $\mathcal{A}$ outputs a geometric query request R_{q_j}. Then, $\mathcal{C}$ responds with a search trapdoor $\widehat{T_{Q_j}} \leftarrow \mathsf{TrapGen}(R_{q_j}, sk_q)$:*
- *Challenge: With R_{q_0} and R_{q_1} selected in the initialization, $\mathcal{C}$ flips a coin $b \in \{0,1\}$ and calculates $\widehat{T_{Q_{b,j}}}$. After that, $\mathcal{C}$ sends it back to $\mathcal{A}$.*
- *Phase 2: $\mathcal{A}$ adaptively selects a series of trapdoor requests and sends them to $\mathcal{C}$, which are still subject to the same restrictions in Phase 1.*

- *Guess: $\mathcal{A}$ outputs his/her guess b' of b. If $b' = b$ holds, $\mathcal{A}$ wins the security game; otherwise, $\mathcal{A}$ fails.*

$\prod$ is secure against IND-SCPA regarding query privacy if, for any polynomial-time adversary $\mathcal{A}$ in the above security game, $\mathcal{A}$ has at most a negligible advantage:

$$Adv_{\prod,\mathcal{A}}^{IND-SCPA,Query} = |Pr(b' = b) - \frac{1}{2}| \leq negl(\lambda).$$

6.1. Data Privacy and Identity Privacy

Theorem 2. *Our EPTA-T scheme guarantees IND-SCPA data privacy and identity privacy.*

Proof. To prove the IND-SCPA data privacy of EPTA-T, we should demonstrate that $\mathcal{A}$ cannot distinguish the ciphertext $\widehat{C_{I_{0,j}}}$ and $\widehat{C_{I_{1,j}}}$ based on the security game defined in Definition 2, even though $\mathcal{A}$ has oracle access to IndexEnc.

Considering that the $\mathcal{W}_i$'s spatial data L_i are firstly encoded as spatial vector $\overrightarrow{\mathsf{p}_i}$ by using Gray code, then $\mathcal{W}_i$ transforms $\overrightarrow{\mathsf{p}_i}$ into a positive integer v_i and obtains a hash value by using the hash function h. After that, the $\mathcal{W}_i$ calculates an $(n+1)$-dimensional vector I_i based on the hash value $h(v_i)$. In addition, the $\mathcal{W}_i$ permutes the random vector $\overline{I_i}$ to $\widehat{I_i}$ with the random permutation π. Following this, the permuted vector $\widehat{I_i}$ is transformed into a corresponding diagonal matrix $\widehat{D_{I_i}}$. Finally, the spatial data L_i are encrypted as $\widehat{C_{I_i}} = M_{i,1}\mathcal{U}_i\widehat{D_{I_i}}M_{i,2}$. Since $\mathcal{A}$ has no idea about the one-time random value α, the random permutation π, and the random triangular matrix $\mathcal{U}_i$, it is hard to recover the secret key sk_i. Additionally, $\mathcal{A}$ can obtain several plaintext–ciphertext pairs, but it is difficult for $\mathcal{A}$ to launch a linear analysis attack.

In Phase 1 and Phase 2 of Definition 2, $\mathcal{A}$ can adaptively select different spatial data $\{L'_j\}_{j=1}^m$ and obtain the corresponding ciphertexts $\{\widehat{C_{I_j}}\}_{j=1}^m$. Unfortunately, both $\overline{I_i}$ and $\mathcal{U}_i$ are a one-time random vector and matrix determined by $\mathcal{C}$, and the ciphertexts $\{\widehat{C_{I_j}}\}_{j=1}^m$ are random according to $\mathcal{A}$. That is, for the given ciphertexts $\widehat{C_{I_{0,j}}}$ and $\widehat{C_{I_{1,j}}}$ selected by $\mathcal{A}$, $\mathcal{A}$ cannot distinguish which spatial data are actually encrypted. Therefore, even though $\mathcal{A}$ has oracle access to IndexEnc, it has a random guess b' of b with a negligible advantage:

$$Adv_{\prod,\mathcal{A}}^{IND-SCPA,Data} = |Pr(b' = b) - \frac{1}{2}| \leq negl(\lambda).$$

For the identity privacy in EPTA-T, since the worker's identity id_ϱ is encrypted as c_ϱ by using AES and the secret key of AES is kept private to $\mathcal{A}$, the identity privacy can be well protected. □

6.2. Query Privacy and Result Privacy

Theorem 3. *Our EPTA-T scheme guarantees IND-SCPA query privacy and result privacy.*

Proof. Based on Definition 3, to prove the query privacy, we should demonstrate that $\mathcal{A}$ cannot obtain any sensitive information from the encrypted trapdoor $\widehat{T_Q}$, encrypted attribute set C_x, and matrix trace $tr(C_{I_i}T_Q)$. Similar to the operation in IndexEnc, $\mathcal{A}$ can adaptively select different geometric query range $\{R_{q_j}\}_{j=1}^m$ and observe the corresponding trapdoor $\{\widehat{T_{Q_j}}\}_{j=1}^m$. However, both $\overline{Q}$ and $\mathcal{P}_q$ are a one-time random vector and matrix determined by $\mathcal{C}$, and the encrypted trapdoor $\{\widehat{T_{Q_j}}\}_{j=1}^m$ is random according to $\mathcal{A}$. Therefore, for the given trapdoors $\widehat{T_{Q_{0,j}}}$ and $\widehat{T_{Q_{1,j}}}$ selected by $\mathcal{A}$, even though $\mathcal{A}$ has oracle access to TrapGen, $\mathcal{A}$ only has a random guess b' of b for the geometric query range with a negligible advantage:

$$Adv_{\prod,\mathcal{A}}^{IND-SCPA,Query} = |Pr(b' = b) - \frac{1}{2}| \leq negl(\lambda).$$

In addition, $\mathcal{A}$ can obtain $F_{root} = e(g_p, H)$ and $tr(C_{I_i} T_Q)$ for temporal access control and task allocation, respectively. However, the privacy of the temporal access tree can be well guaranteed by using Bilinear Diffie–Hellman Assumption (DBDH). Furthermore, the query result only reveals whether $tr(C_{I_i} T_Q)$ is equal to 0 or not, and no more sensitive information of the geometric query can be obtained by $\mathcal{A}$. Therefore, our EPTA-T scheme can well guarantee IND-SCPA query privacy and result privacy. □

7. Performance Evaluation

In this section, we provide a thorough experimental evaluation of the EPTA-T scheme. We first depict the configuration of the experimental environment and selection of parameters. Since our EPTA-T scheme is the first work to support fine-grained and temporal access control in privacy-preserving task allocation, we present the comparison of EPTA-T with Privacy-Preserving Boolean Range Query with Temporal Access Control (PBRQ-T) [6] in terms of the computational overhead on IndexEnc, TrapGen, and Query.

7.1. Implementation Settings

Experimental environment. We implemented our EPTA-T scheme and PBRQ-T with JAVA and utilized the JDK library to implement cryptographic primitives such as AES. We conducted experiments on a laptop with a 2.8 GHz, Intel Core i7, 16 GB RAM as $\mathcal{SP}$ and an Android phone with 8G RAM and Octa-core Processors as $\mathcal{W}$ and $\mathcal{DR}$. In order to precisely measure the computational cost of the $\mathcal{SP}$, $\mathcal{W}$, and $\mathcal{DR}$, all experiments were performed on the same devices. Furthermore, all the reported running times were the average of 100 experiments.

Dataset. We randomly selected 10,000 spatial data from the Yelp dataset (https://www.yelp.com/dataset (accessed on 19 May 2023)) as the test dataset, where each spatial datum included the spatial location and attributes. The number of attributes was 44 in the test dataset.

Parameter setting. To compare with PBRQ-T, all primes were set as 60 bits. In addition, we divided the space into 1000×1000 cells; the dimension of the spatial vector was 100; the geometric query range R_q was set as a 64×64 square. In our experimental evaluation, we set the number of the $\mathcal{DR}$'s attributes and system attributes as 10, and we assumed that there was only one temporal attribute in the task allocation system.

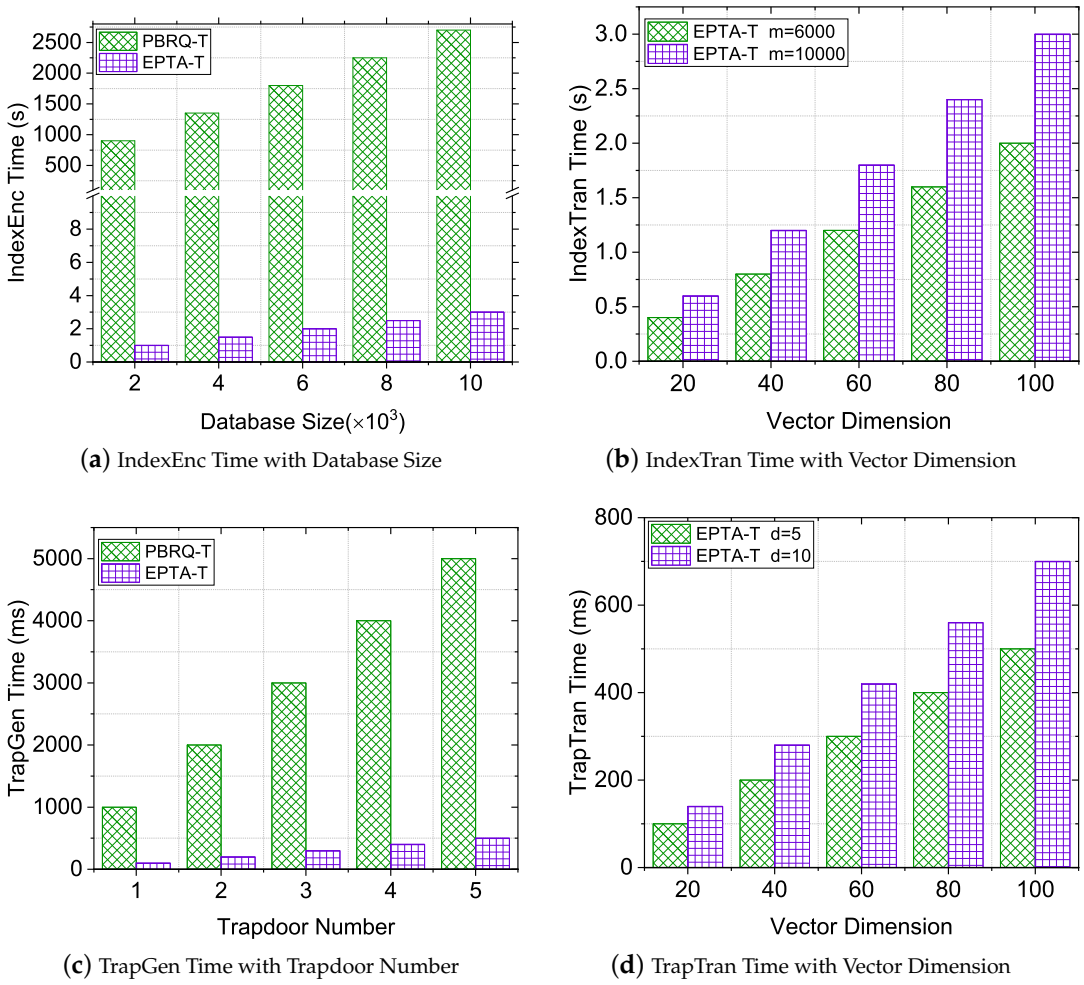
We compared our EPTA-T scheme with the state-of-the-art PBRQ-T scheme [6]. Note that the PBRQ-T scheme does not support task recommendation in mobile crowdsensing systems. To test the cost domination of our EPTA-T over different database sizes m and dimension vectors n, we mainly evaluated the complexity of index encryption, index transformation, trapdoor generation, trapdoor transformation, and query between $\mathcal{W}$, $\mathcal{DR}$, and $\mathcal{SP}$, respectively.

7.2. Evaluation and Comparison

Performance evaluation of IndexEnc and IndexTran. The cost of IndexEnc was dominated by the database size m. As shown in Figure 3a, when the database size was 10,000, the index encryption time of EPTA-T was about 3 s, while that of PBRQ-T was more than 2500 s. The reason is that our EPTA-T scheme only executes randomizable matrix multiplication for data privacy protection, while PBRQ-T conducts bilinear mapping and exponentiation operations, which have a more-expensive computational cost than our scheme. Therefore, with the increase of the database's size, EPTA-T can save more computational costs on the worker side. Since PBRQ-T does not involve the index transformation operation, we only evaluated our EPTA-T on IndexTran with varying vector dimensions (i.e., n). As shown in Figure 3b, we noticed that our proposed EPTA-T increased linearly with the variable vector dimension n. In addition, the IndexTran costs over different database sizes m were almost close in the lower-dimensional setting.

Performance evaluation of TrapGen and TrapTran. From Figure 3c, we can notice that our EPTA-T had less computational cost for TrapGen and slightly increased with the

variable number d of trapdoors, and it was at least 10× faster than PBRQ-T. The underlying reason is that the number of trapdoors had little effect on the calculation of the matrix multiplication. As shown in Figure 3d, we can notice that the TrapTran computational cost of EPTA-T linearly grew with the variable n. Specifically, when the vector dimension achieved 100, the time cost of TrapTran was only 0.7 s, even though the $\mathcal{DR}$ submitted 10 geometric queries at one time.



(**a**) IndexEnc Time with Database Size

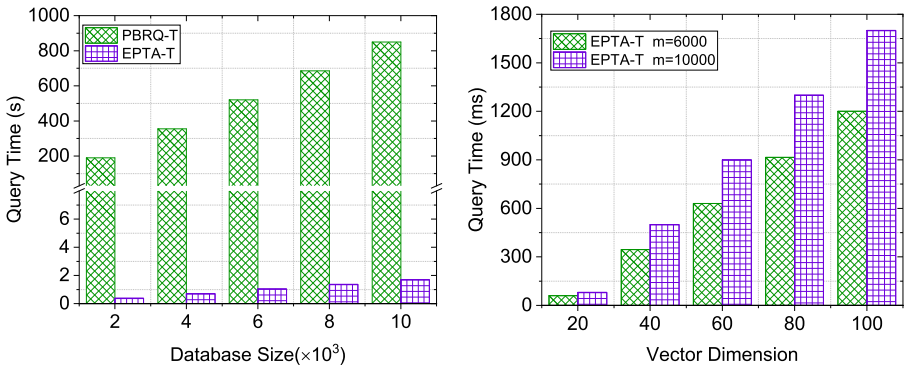
(**b**) IndexTran Time with Vector Dimension

(**c**) TrapGen Time with Trapdoor Number

(**d**) TrapTran Time with Vector Dimension

Figure 3. Performance Evaluation of IndexEnc, IndexTran, TrapGen, and TrapTran.

Performance evaluation of Query. To comprehensively evaluate the query cost, we ran the experiments on the database with different sizes and vectors with different dimensions. As shown in Figure 4, the query overhead was scarcely influenced by the database size and the vector dimension. From Figure 4a, we can notice that the query time of both EPTA-T and PBRQ-T increased linearly with m, and the query performance of EPTA-T outperformed that of PBRQ-T. The query time of EPTA-T was about 500× faster than that of PBRQ-T, which indicates that the calculation operations of matrix trace and functional integration were more efficient than those of exponent arithmetic and bilinear mapping. From Figure 4b, we can notice that a higher vector dimension had a greater query time cost with EPTA-T. The underlying logic is that the increase of the vector dimension d caused the number of matrix dimensions to grow and the computational cost of the matrix-trace-based operation to increase.



(**a**) Query Time with Database Size (**b**) Query Time with Vector Dimension

Figure 4. Performance Evaluation of Query.

8. Conclusions

In this paper, we investigated and studied the problem of privacy-preserving task allocation with temporal access control for mobile crowdsensing. Specifically, we proposed an Efficient and Privacy-Preserving Task Allocation with Temporal Access Control (EPTA-T) scheme that efficiently achieved secure task allocation in multi-user settings by using Gray code and randomizable matrix multiplication. Moreover, EPTA-T incorporates temporal access control by leveraging function differentiation and integration, along with CP-ABE, to regulate access to tasks over time. Furthermore, formal security analysis and experimental evaluations demonstrated that EPTA-T protects data privacy and query privacy and achieves efficient task allocation compared with the state-of-the-art scheme. Regarding open problems, privacy-preserving spatial-keyword-based task allocation is still a challenging issue that needs to be resolved. For future work, we will further consider additional constraints in privacy-preserving task allocation, such as workers' interests and incentives.

Author Contributions: Conceptualization, F.S. and Y.L.; methodology, F.S.; software, F.S. and S.M.; validation, F.S., Y.L. and Q.J.; writing—original draft preparation, F.S. and Y.L.; writing—review and editing, Q.J., X.Z. and Z.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Defense Industrial Technology Development Program (No. JCKY2021602B002), the National Natural Science Foundation of China (Nos. U20A20174, U22B2062, 62202051, 62202234, and 42001247), the National Key R&D Projects (No. 2021YFB00900 and No. 2018YFB0704000), and the China Postdoctoral Science Foundation (No. 2021M700435 and No. 2021TQ0042).

Data Availability Statement: The data presented in this study are available upon request from the corresponding authors. The data are not publicly available due to the privacy requirements of the project.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of the data; in the writing of the manuscript; nor in the decision to publish the results.

References

1. Song, F.; Qin, Z.; Liu, D.; Zhang, J.; Lin, X.; Shen, X. Privacy-preserving task matching with threshold similarity search via vehicular crowdsourcing. *IEEE Trans. Veh. Technol.* **2021**, *70*, 7161–7175. [CrossRef]
2. Ni, J.; Zhang, K.; Xia, Q.; Lin, X.; Shen, X. Enabling strong privacy preservation and accurate task allocation for mobile crowdsensing. *IEEE Trans. Mob. Comput.* **2019**, *19*, 1317–1331. [CrossRef]
3. Liang, J.; Qin, Z.; Xiao, S.; Ou, L.; Lin, X. Efficient and secure decision tree classification for cloud-assisted online diagnosis services. *IEEE Trans. Dependable Secur. Comput.* **2019**, *18*, 1632–1644. [CrossRef]
4. Shu, J.; Jia, X.; Yang, K.; Wang, H. Privacy-preserving task recommendation services for crowdsourcing. *IEEE Trans. Serv. Comput.* **2018**, *14*, 235–247. [CrossRef]

5. Zhu, Y.; Hu, H.; Ahn, G.J.; Huang, D.; Wang, S. Towards temporal access control in cloud computing. In Proceedings of the INFOCOM, Orlando, FL, USA, 25–30 March 2012; pp. 2576–2580.
6. Tong, Q.; Li, X.; Miao, Y.; Liu, X.; Weng, J.; Deng, R.H. Privacy-preserving Boolean range query with temporal access control in mobile computing. *IEEE Trans. Knowl. Data Eng.* **2022**, *35*, 5159–5172. [CrossRef]
7. Deng, H.; Qin, Z.; Wu, Q.; Deng, R.H.; Guan, Z.; Hu, Y.; Li, F. Achieving fine-grained data sharing for hierarchical organizations in clouds. *IEEE Trans. Dependable Secur. Comput.* **2022**, *20*, 1364–1377. [CrossRef]
8. Zhang, C.; Zhu, L.; Xu, C.; Ni, J.; Huang, C.; Shen, X. Location privacy-preserving task recommendation with geometric range query in mobile crowdsensing. *IEEE Trans. Mob. Comput.* **2021**, *21*, 4410–4425. [CrossRef]
9. Wang, B.; Li, M.; Wang, H. Geometric range search on encrypted spatial data. *IEEE Trans. Inf. Forensics Secur.* **2015**, *11*, 704–719. [CrossRef]
10. Wang, B.; Li, M.; Xiong, L. FastGeo: Efficient geometric range queries on encrypted spatial data. *IEEE Trans. Dependable Secur. Comput.* **2017**, *16*, 245–258. [CrossRef]
11. Zhang, S.; Ray, S.; Lu, R.; Guan, Y.; Zheng, Y.; Shao, J. Efficient and Privacy-Preserving Spatial Keyword Similarity Query over Encrypted Data. *IEEE Trans. Dependable Secur. Comput.* 2022, *early access*. [CrossRef]
12. Song, F.; Qin, Z.; Xue, L.; Zhang, J.; Lin, X.; Shen, X. Privacy-preserving keyword similarity search over encrypted spatial data in cloud computing. *IEEE Internet Things J.* **2021**, *9*, 6184–6198. [CrossRef]
13. Liang, J.; Qin, Z.; Ni, J.; Lin, X.; Shen, X. Practical and secure SVM classification for cloud-based remote clinical decision services. *IEEE Trans. Comput.* **2020**, *70*, 1612–1625. [CrossRef]
14. Song, F.; Qin, Z.; Zhang, J.; Liu, D.; Liang, J.; Shen, X. Efficient and privacy-preserving outsourced image retrieval in public clouds. In Proceedings of the GLOBECOM, Taipei, Taiwan, 7–11 December 2020; pp. 1–6.
15. Zheng, Y.; Lu, R.; Zhang, S.; Guan, Y.; Wang, F.; Shao, J.; Zhu, H. PRkNN: Efficient and Privacy-Preserving Reverse kNN Query Over Encrypted Data. *IEEE Trans. Dependable Secur. Comput.* 2022, *early access*. [CrossRef]
16. Zhang, S.; Ray, S.; Lu, R.; Guan, Y.; Zheng, Y.; Shao, J. Toward Privacy-Preserving Aggregate Reverse Skyline Query with Strong Security. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 2538-2552. [CrossRef]
17. Zhang, C.; Hu, C.; Wu, T.; Zhu, L.; Liu, X. Achieving Efficient and Privacy-Preserving Neural Network Training and Prediction in Cloud Environments. *IEEE Trans. Dependable Secur. Comput.* 2022, *early access*. [CrossRef]
18. Wang, H.; Wang, E.; Yang, Y.; Wu, J.; Dressler, F. Privacy-Preserving online task assignment in spatial crowdsourcing: A graph-based approach. In Proceedings of the INFOCOM, Virtual Conference, 2–5 May 2022; pp. 570–579.
19. Xia, Y.; Zhao, B.; Tang, S.; Wu, H.T. Repot: Real-time and privacy-preserving online task assignment for mobile crowdsensing. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4035. [CrossRef]
20. Zhou, P.; Chen, W.; Ji, S.; Jiang, H.; Yu, L.; Wu, D. Privacy-preserving online task allocation in edge-computing-enabled massive crowdsensing. *IEEE Internet Things J.* **2019**, *6*, 7773–7787. [CrossRef]
21. Xu, Z.; Wu, L.; Qin, C.; Li, S.; Zhang, S.; Lu, R. PPTA: Privacy-Preserving Task Assignment Based on Inner Product Functional Encryption in SAM. *IEEE Internet Things J.* **2022**, *10*, 254–267. [CrossRef]
22. Xue, L.; Liu, D.; Huang, C.; Shen, X.; Zhuang, W.; Sun, R.; Ying, B. Blockchain-Based Data Sharing with Key Update for Future Networks. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 3437–3451. [CrossRef]
23. Huang, C.; Liu, D.; Yang, A.; Lu, R.; Shen, X. Multi-client secure and efficient dpf-based keyword search for cloud storage. *IEEE Trans. Dependable Secur. Comput.* 2023, *early access*. [CrossRef]
24. Song, F.; Qin, Z.; Liang, J.; Lin, X. An efficient and privacy-preserving multi-user multi-keyword search scheme without key sharing. In Proceedings of the ICC, Montreal, QC, Canada, 14–23 June 2021; pp. 1–6.
25. Li, R.; Liu, A.X.; Wang, A.L.; Bruhadeshwar, B. Fast and scalable range query processing with strong privacy protection for cloud computing. *IEEE/ACM Trans. Netw.* **2015**, *24*, 2305–2318. [CrossRef]
26. Liu, D.; Wu, H.; Huang, C.; Ni, J.; Shen, X. Blockchain-based credential management for anonymous authentication in sagvn. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 3104–3116. [CrossRef]
27. Li, S.; Zhang, Y.; Xu, C.; Cheng, N.; Liu, Z.; Du, Y.; Shen, X. HealthFort: A Cloud-Based Ehealth System with Conditional Forward Transparency and Secure Provenance via Blockchain. *IEEE Trans. Mob. Comput.* 2022, *early access*. [CrossRef]
28. Ren, H.; Li, H.; Liu, D.; Xu, G.; Shen, X. Enabling Secure and Versatile Packet Inspection with Probable Cause Privacy for Outsourced Middlebox. *IEEE Trans. Cloud Comput.* **2021**, *10*, 2580–2594. [CrossRef]
29. Hu, C.; Zhang, C.; Lei, D.; Wu, T.; Liu, X.; Zhu, L. Achieving Privacy-Preserving and Verifiable Support Vector Machine Training in the Cloud. *IEEE Trans. Inf. Forensics Secur.* 2023, *early access*. [CrossRef]
30. Wang, X.; Ma, J.; Liu, X.; Miao, Y.; Liu, Y.; Deng, R.H. Forward/backward and Content Private DSSE for Spatial Keyword Queries. *IEEE Trans. Dependable Secur. Comput.* 2022, *early access*. [CrossRef]
31. Chen, D.; Zhang, N.; Cheng, N.; Zhang, K.; Qin, Z.; Shen, X. Physical layer based message authentication with secure channel codes. *IEEE Trans. Dependable Secur. Comput.* **2018**, *17*, 1079–1093. [CrossRef]

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