Efficient and Privacy-Preserving Spatial-Feature-Based Reverse kNN Query

Yandong Zheng, *Member, IEEE,* Rongxing Lu, *Fellow, IEEE,* Yunguo Guan, Songnian Zhang, Jun Shao, *Senior Member, IEEE,* Fengwei Wang, *Member, IEEE,* and Hui Zhu, *Senior Member, IEEE*

Abstract—Reverse k nearest neighbor (RkNN) query has been widely applied in the targeted push of information. Many schemes for the RkNN query on encrypted data have been proposed for coordinating the emerging trend of outsourcing data to the cloud. However, none of them supports the spatial data with many features, a prevalent data type in location-based services, e.g., each user in online dating apps usually has a spatial location and many personality trait features. Meanwhile, incorporating features with the spatial data endows the spatial-feature-based RkNN query to provide more precise services than the spatial-based RkNN query. Therefore, as a steppingstone, we propose an efficient and privacy-preserving spatial-feature-based RkNN scheme in this work for the first time. Specifically, we first design a modified intersection and union R tree (MIUR-tree) to index the spatial and feature data. Then, we introduce an MIUR-tree based RkNN query algorithm in the filter and refinement framework to efficiently process RkNN queries. After that, based on a symmetric homomorphic encryption (SHE) scheme, we design a private filter protocol and a private refinement protocol, and leverage them to propose our RkNN query scheme. Rigorous security analysis demonstrates that our scheme is privacy-preserving, and extensive experiments indicate that our scheme is computationally efficient.

Index Terms—Spatial-Feature-Based RkNN Query, MIUR-Tree, Filter and Refinement, Encrypted Data, Homomorphic Encryption.

1 Introduction

The rapid development of communication technologies, the boom of Internet of Things, and the growing adoption of Industry 4.0 have driven the increasing volumes of data, and the big data market is predicted to increase by USD 247 Billion during 2021-2025, as reported by Technavio [1]. The big data are fully exploited to provide a wide range of query services, such as range query [2], [3], aggregation query [4], k nearest neighbors (kNN) query [5], and reverse kNN (RkNN) query [6], [7]. Meanwhile, driven by the prosperity of cloud computing technology, outsourcing big data and the query services to the cloud has become a new paradigm. However, since the cloud servers are run by the third-party organizations (e.g., Google and Amazon) and are not fully trusted, directly outsourcing the plaintext data to the cloud will arise the privacy leakage concern [8], [9]. To allay this concern, data owners usually outsource the encrypted data to the cloud. Although encryption techniques address the privacy concern, it inevitably affects cloud servers to implement query services.

Many schemes have been proposed for different kinds of query services over outsourced encrypted data, but the research on privacy-preserving RkNN query is far less than expected [10]. Especially, the study on spatial-feature-based RkNN is still in the blank stage. RkNN query aims to retrieve the k objects that are the most interested to the query object and has wide applications in the targeted push of information. Spatial-feature data, i.e., spatial data with many features, is a prevalent data type in location-based services, e.g., each user in online dating apps usually has a spatial location and many personality traits. As a result, the spatial-feature-based RkNN query is a fundamental query type in the targeted push of location based services. It goes without saying that incorporating features with the spatial data endows the spatial-feature-based RkNN query to provide more precise services than the RkNN query only based on spatial data.

In Fig. 1, we give an example to explain the spatial-feature-based RkNN query. As shown in Fig. 1, in online dating apps, the dataset has five users, including {"Bob", "James", "Lucas", "Henry", "Aiden"}. Each user is associated with two attribute vectors, including (i) a spatial location $\mathbf{s}_i$ that is specified by the longitude and latitude, e.g., the location of "633 Windsor Street in Fredericton" can be specified as (-66.643822,45.949070); and (ii) a feature vector $\mathbf{t}_i$ that is specified by many personality traits, e.g., extroversion, rational, openness, and the sense of humor, and each trait is scored 1-5 points. Let "Amy" be a query user who is in the spatial location $\mathbf{s}_q = (10, 3)$ and has the personality traits $\mathbf{t}_q = (4, 2, 4, 1)$. If "Amy" intends to know who are interested to her, she can launch an RkNN query to retrieve users who regard her as one of their k nearest neighbors. The spatial and feature similarity between users is a weighted summation value of Euclidean distance based

- Y. Zheng is with the State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, 710071, China, and also with the Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, 130012, P.R. China. (zhengyandong@xidian.edu.cn)
- F. Wang and H. Zhu are with the State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, 710071, China (e-mail: wangfengwei@xidian.edu.cn, zhuhui@xidian.edu.cn) (Corresponding author: Hui Zhu).
- R. Lu, S. Zhang, and Y. Guan are with the Faculty of Computer Science, University of New Brunswick, Fredericton, NB E3B 5A3, Canada (e-mail: rlu1@unb.ca, szhang17@unb.ca, yguan4@unb.ca).
- J. Shao is with School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou, 310018, China (e-mail: chn.junshao@gmail.com).

User	Spatial	Longitude	Latitude	Feature	Extroversion	Rational	Openness	Humor
Bob	s_1	4	3	t_1	4	2	3	1
James	s_2	8	4	t_2	4	1	4	1
Lucas	s_3	5	2	t_3	2	4	3	2
Henry	s_4	10	4	t_4	3	2	4	1
Aiden	s_5	9	3	t_5	4	2	4	2
Amy	s_q	10	3	t_q	4	2	4	1

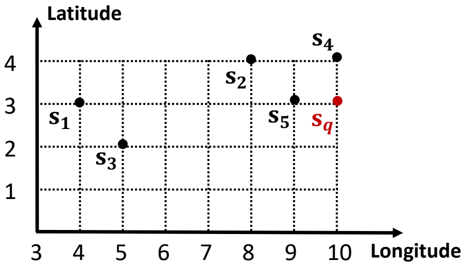


Fig. 1. Example of spatial-feature-based RkNN query

spatial similarity and extended Jaccard based feature similarity, which is formalized by Definition 3.1 in Section 3.1. Obliviously, compared with the spatial-based RkNN query, the spatial-feature-based RkNN query can provide more precise targeted push. For instance, when $k = 2$, the R2NN of "Amy" based on the spatial data is {"Henry", "Aiden"}. However, the R2NN of "Amy" based on the spatial and feature data will be {"James", "Henry", "Aiden"}, because the personality traits of "James" are highly similar to those of "Amy".

Several RkNN query schemes [10]–[14] over encrypted data have been proposed. Unfortunately, none of them supports the spatial-feature-based RkNN query. Specifically, the schemes in [10]–[13] focus on the RkNN queries over spatial data and do not take the feature data into consideration, which are naturally inapplicable to our spatial-feature-based RkNN queries. The scheme in [14] is designed for the Euclidean distance based multi-dimensional RkNN query, and the proposed tree structure is not applicable to the spatial-feature-based RkNN query. Thus, it also cannot be directly applied to achieve the spatial-feature-based RkNN query. Another potential solution is to employ existing privacy-preserving kNN query schemes [5], [15]–[24] to achieve privacy-preserving RkNN queries. However, the kNN query aims to retrieve top-k records having the smallest distance to the query record. Its query goal is different from the RkNN query, leading to the infeasibility of using kNN query schemes to implement RkNN queries. Therefore, as a steppingstone, we consider the spatial-feature-based RkNN query over encrypted data for the first time. Our contributions are three folds as follows.

- First, we design a modified intersection and union R tree, named MIUR-tree, to index the spatial and feature dataset. Then, we introduce an MIUR-tree based RkNN query algorithm in the filter and refinement framework to efficiently process RkNN queries. The filter stage is to retrieve candidate query results, and the refinement stage is further to refine the candidate results.
- Second, based on a symmetric homomorphic encryption (SHE) scheme, we design a private filter protocol and a private refinement protocol in the two-server model to protect the privacy of the RkNN query algorithm.
- Third, based on the above protocols, we propose an efficient and privacy-preserving spatial-feature-based RkNN query scheme. Meanwhile, we rigorously prove its security in the simulation-based real/ideal model, and the results demonstrate that our scheme is privacy-preserving. In addition, extensive experiments indicate that our scheme is computationally efficient.

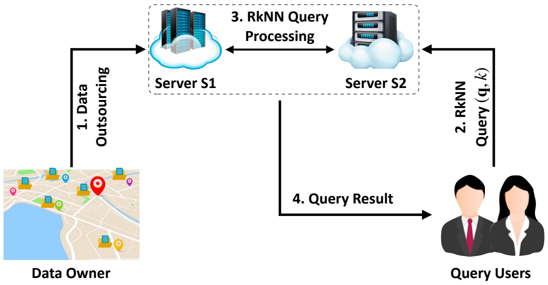


Fig. 2. Spatial-feature-based RkNN query model in our system

The remainder of this paper is organized as follows. We introduce our models and design goals in Section 2 and review some preliminaries in Section 3. In Section 4, we present the building blocks of our scheme, followed by the description of our scheme in Section 5. In Section 6 and Section 7, we respectively analyze the security of our scheme and evaluate its performance. Finally, we review related works and draw our conclusion in Section 8 and Section 9.

2 Models and Design Goals

In this section, we will introduce our system model and security model, and identify our design goals.

2.1 System Model

Our system focuses on a spatial-feature-based RkNN query model in the outsourced scenario, which involves three kinds of participants, including a data owner, two cloud servers, and multiple query users, as shown in Fig. 2.

- Data owner: The data owner possesses a spatial and feature dataset $\mathcal{D} = \{\mathbf{x}_i = (\mathbf{s}_i, \mathbf{t}_i)\}_{i=1}^n$, where $\mathbf{s}_i$ and $\mathbf{t}_i$ respectively denote the spatial and feature vectors. To derive benefits from the dataset $\mathcal{D}$, the data owner offers spatial-feature-based RkNN query services to users in need. However, restrained by its computing capability and storage space, the data owner outsources the dataset $\mathcal{D}$ together with the query services to two powerful cloud servers. Before outsourcing the dataset to the cloud servers, the data owner first represents the dataset to a tree-based index for improving query efficiency and then encrypts the index for protecting data privacy.

• Two Cloud Servers: We have two cloud servers, i.e., S1 and S2, with abundant computing and storage resources. Meanwhile, S1 stores the encrypted dataset outsourced by the data owner and handles RkNN queries with the help of S2. Specifically, on receiving a spatial-feature-based RkNN query request $(\mathbf{q}, k)$ from a query user, S1 and S2 will collaboratively search on the encrypted dataset for the query result and return the result to the user.

• Query Users: There are multiple query users in our system, and they can enjoy the spatial-feature-based RkNN query services from the two servers.

Informally, our scheme contains four algorithms and can be defined as $\Pi = (\mathsf{Setup}, \mathsf{Outsourcing}, \mathsf{TokenGen}, \mathsf{QueryProcessing})$.

- $\mathsf{Setup}(\mathtt{k}_0, \mathtt{k}_1, \mathtt{k}_2)$: In the setup algorithm, on input security parameters $\{\mathtt{k}_0, \mathtt{k}_1, \mathtt{k}_2\}$, the data owner generates a public key $\mathtt{pk}$ and a secret key $\mathtt{sk}$. Then, it publishes $\mathtt{pk}$ and sends $\mathtt{sk}$ to the server S2.
- $\mathsf{Outsourcing}(\mathcal{D}, \mathtt{pk}, \mathtt{sk})$: In the data outsourcing algorithm, the data owner represents the dataset $\mathcal{D}$ to an MIUR-tree $\mathcal{T}$ and uses $\{\mathtt{pk}, \mathtt{sk}\}$ to encrypt $\mathcal{T}$ into an encrypted tree $\mathtt{Cipher}_{\mathcal{T}}$.
- $\mathsf{TokenGen}((\mathbf{q}, k), \mathtt{pk})$: In the token generation algorithm, the query user utilizes $\mathtt{pk}$ to encrypt each RkNN query $(\mathbf{q}, k)$ into a query token $(\mathtt{Token}_{\mathtt{Fil}}, k)$ and sends it to the server S1.
- $\mathsf{QueryProcessing}(\mathsf{S1}{:}\mathtt{Cipher}_{\mathcal{T}}, (\mathtt{Token}_{\mathtt{Fil}}, k); \mathsf{S2}{:}\mathtt{sk})$: In the query processing phase, the server S1 with $\mathtt{Cipher}_{\mathcal{T}}$ and $(\mathtt{Token}_{\mathtt{Fil}}, k)$ collaboratively processes the RkNN query with the server S2 with $\mathtt{sk}$.

2.2 Security Model

In the security model, we specify the security assumptions for each kind of participant. Regarding the data owner, since it is the creator of the entire system and the owner of the dataset, we assume that it is trusted. For the two servers in the cloud, since they are provided by the third parties, we assume that they are honest-but-curious. That is, S1 and S2 will honestly store the encrypted dataset and handle RkNN queries but may be curious about some private information, including the plaintext of dataset and query requests. Meanwhile, we assume that there is no collusion between two servers. The honest and non-collusive assumptions are widely accepted in security communities [25], [26] because (i) cloud servers' dishonest behaviors will incur serious punishments from regulatory authorities; and (ii) different cloud service providers may have conflicts of interest such that they cannot collude with each other. For query users, we assume that they are honest. The honest assumption is reasonable, because users' bad behaviors will be punished, and their enjoyment of query services will be cancelled. Note that our work focuses on privacy preservation, other active attacks are beyond our scope and will be discussed in our future work.

In addition, since our scheme is a searchable encryption scheme, its security can be measured by the leakages to the *honest-but-curious* servers S1 and S2. Meanwhile, similar to the scheme in [27], we can prove the security of our scheme in the simulation-based real/ideal model. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ respectively denote the leakages of our scheme to S1 and S2. Then, the selective security of our scheme can be defined as Definition 2.1.

Definition 2.1 (Selective security of RkNN query scheme) *Our scheme is selectively secure with the leakages $\mathcal{L}_1$ and $\mathcal{L}_2$ if for any probabilistic polynomial-time (PPT) adversaries $\mathtt{Adv}_1$ and $\mathtt{Adv}_2$ issuing a polynomial number of RkNN queries in the role of S1 and S2, there exists a PPT simulator $\mathtt{Sim}$ such that the probability that $\mathtt{Adv}_1$ and $\mathtt{Adv}_2$ can distinguish their views in the real and ideal models is negligible, i.e., $|\Pr(b_1 = 1|\mathtt{Real}_{\mathtt{Adv}_1,\mathtt{Adv}_2}) - \Pr(b_1 = 1|\mathtt{Ideal}^{\mathtt{Sim}}_{\mathtt{Adv}_1,\mathtt{Adv}_2})|$ and $|\Pr(b_2 = 1|\mathtt{Real}_{\mathtt{Adv}_1,\mathtt{Adv}_2}) - \Pr(b_2 = 1|\mathtt{Ideal}^{\mathtt{Sim}}_{\mathtt{Adv}_1,\mathtt{Adv}_2})|$ are negligible, where b_1 and b_2 denote the outputs of $\mathtt{Adv}_1$ and $\mathtt{Adv}_2$ in the real model $\mathtt{Real}_{\mathtt{Adv}_1,\mathtt{Adv}_2}$ and the ideal model $\mathtt{Ideal}^{\mathtt{Sim}}_{\mathtt{Adv}_1,\mathtt{Adv}_2}$ that are defined below.*

• $\mathtt{Real}_{\mathtt{Adv}_1,\mathtt{Adv}_2}$: In the real model, $\mathtt{Adv}_1$ first calls the setup algorithm of our scheme to generate the parameters and keys of the system. Then, $\mathtt{Adv}_1$ calls the data outsourcing algorithm of our scheme to obtain an encrypted MIUR-tree of the spatial and feature dataset $\mathtt{Cipher}_{\mathcal{T}}$. After that, $\mathtt{Adv}_1$ gets the query tokens of γ spatial-feature-based RkNN queries $\{(\mathtt{Token}_{\mathtt{Fil},j}, k_j)\}_{j=1}^{\gamma}$ by calling the token generation algorithm of our scheme. Furthermore, $\mathtt{Adv}_1$ runs the RkNN query processing algorithm of our scheme to handle the queries $\{(\mathtt{Token}_{\mathtt{Fil},j}, k_j)\}_{j=1}^{\gamma}$ with $\mathtt{Adv}_2$. Meanwhile, $\mathtt{Adv}_1$ and $\mathtt{Adv}_2$ observe the real data generated in the above algorithms and output two bits $b1$ and b_2, respectively.

• $\mathtt{Ideal}^{\mathtt{Sim}}_{\mathtt{Adv}_1,\mathtt{Adv}_2}$: In the ideal model, $\mathtt{Adv}_1$ first setups the system by calling the simulator's setup algorithm. Then, it obtains an encrypted MIUR-tree of the spatial and feature dataset, denoted by $\mathtt{Cipher}^{\mathtt{Sim}}_{\mathcal{T}}$, by running the simulator's data outsourcing algorithm. After that, $\mathtt{Adv}_1$ gets the query tokens of γ spatial-feature-based RkNN queries, denoted by $\{(\mathtt{Token}^{\mathtt{Sim}}_{\mathtt{Fil},j}, k_j)\}_{j=1}^{\gamma}$, by running the simulator's token generation algorithm. Finally, $\mathtt{Adv}_1$ and $\mathtt{Adv}_2$ handle these queries by running the simulator's query processing algorithm. Meanwhile, $\mathtt{Adv}_1$ and $\mathtt{Adv}_2$ observe the simulated data generated in the above simulator's algorithms and output two bits $b1$ and b_2, respectively.

2.3 Design Goals

In this work, we aim to design an efficient and privacy-preserving spatial-feature-based RkNN query schemes and have the following objectives.

• **Privacy preservation.** Our primary objective is to protect the privacy of the outsourced dataset and query requests against the *honest-but-curious* servers S1 and S2.

• **High query efficiency.** Our secondary objective is to achieve high query efficiency by designing an efficient index and constructing a lightweight privacy-preserving spatial-feature-based RkNN query algorithm.

3 Preliminaries

In this section, we first recall some basic definitions and review a symmetric homomorphic encryption (SHE) scheme.

3.1 Basic Definitions

Suppose that $\mathcal{D} = \{\mathbf{x}_i = (\mathbf{s}_i, \mathbf{t}_i)\}_{i=1}^n$ is a spatial and feature dataset, where $\mathbf{s}_i = (s_{i,1}, s_{i,2})$ denotes a spatial vector, and $\mathbf{t}_i = (t_{i,1}, t_{i,2}, \cdots, t_{i,d})$ denotes a d-dimensional feature vector. The similarity between two spatial and feature records is defined as the weighted sum of their spatial similarity and feature similarity, where the spatial similarity is widely measured by Euclidean distance and the feature similarity is widely measured by the extend Jaccard similarity due to its good similarity measure effect for features [28]. Formal definition is shown in Definition 3.1.

Definition 3.1 (Spatial and Feature Similarity [28]) *The spatial and feature similarity between two records $\mathbf{x}_i = (\mathbf{s}_i, \mathbf{t}_i)$ and $\mathbf{x}_j = (\mathbf{s}_j, \mathbf{t}_j)$ is calculated as*

$$Sim(\mathbf{x}_i, \mathbf{x}_j) = \alpha * (1 - \frac{D(\mathbf{s}_i, \mathbf{s}_j) - \phi_s}{\psi_s - \phi_s}) + (1-\alpha) * \frac{J(\mathbf{t}_i, \mathbf{t}_j) - \phi_t}{\psi_t - \phi_t},$$

where (i) $D(\mathbf{s}_i, \mathbf{s}_j)$ denotes the Euclidean distance between $\mathbf{s}_i$ and $\mathbf{s}_j$; (ii) ψ_s and ϕ_s respectively denote the maximum and minimum Euclidean distance between records in the dataset $\mathcal{D}$ and are used for normalizing $D(\mathbf{s}_i, \mathbf{s}_j)$; (iii) $J(\mathbf{t}_i, \mathbf{t}_j)$ denotes the extended Jaccard similarity between $\mathbf{t}_i$ and $\mathbf{t}_j$; (iv) ψ_t and ϕ_t respectively denote the maximum and minimum extended Jaccard similarity between records in the dataset $\mathcal{D}$ and are used for normalizing $J(\mathbf{t}_i, \mathbf{t}_j)$; and (v) α is the weight of Euclidean distance in the spatial and feature similarity and $0 \leq \alpha \leq 1$. Let $\mathbf{t}_i = (t_{i,1}, t_{i,2}, \cdots, t_{i,d})$ and $\mathbf{t}_j = (t_{j,1}, t_{j,2}, \cdots, t_{j,d})$ be two feature vectors. The extended Jaccard similarity between them is calculated as $J(\mathbf{t}_i, \mathbf{t}_j) = \frac{\sum_{l=1}^d t_{i,l} * t_{j,l}}{\|\mathbf{t}_i\|^2 + \|\mathbf{t}_j\|^2 - \sum_{l=1}^d t_{i,l} * t_{j,l}}$, where $\|\mathbf{t}_i\|^2 = \sum_{l=1}^d t_{i,l}^2$ and $\|\mathbf{t}_j\|^2 = \sum_{l=1}^d t_{j,l}^2$.

Based on the spatial and feature similarity, we define the spatial-feature-based RkNN query as Definition 3.2.

Definition 3.2 (Spatial-Feature-Based RkNN Query [28]) *Let $\mathbf{q} = (\mathbf{s}_q, \mathbf{t}_q)$ be a query record, where $\mathbf{s}_q$ is a spatial vector and $\mathbf{t}_q$ is a feature vector. The spatial-feature-based RkNN query $\mathbf{q}$ over the dataset $\mathcal{D}$ is to search for records regarding $\mathbf{q}$ as their k nearest neighbors. The query result is $\mathcal{R} = \{\mathbf{x}_i | Sim(\mathbf{x}_i, \mathbf{q}) \geq \tau_i^{k\mathtt{NN}}\}$, where $\tau_i^{k\mathtt{NN}}$ denotes the similarity between $\mathbf{x}_i$ and its k-th nearest neighbor.*

3.2 SHE Scheme

The SHE scheme in [29] is a leveled symmetric homomorphic encryption scheme and is proved to be semantically secure in [30]. It has three algorithms as follows.

- $\mathsf{KeyGen}(\mathtt{k}_0, \mathtt{k}_1, \mathtt{k}_2)$: Given security parameters $\mathtt{k}_0$, $\mathtt{k}_1$, and $\mathtt{k}_2$ with $\mathtt{k}_1 \ll \mathtt{k}_2 < \mathtt{k}_0$, the key generation algorithm randomly selects prime numbers $\mathtt{p}, \mathtt{q}$ and a number $\mathtt{L}$, where $|\mathtt{p}| = |\mathtt{q}| = \mathtt{k}_0$ and $|\mathtt{L}| = \mathtt{k}_2$. After that, it calculates $\mathtt{N} = \mathtt{p} * \mathtt{q}$ and outputs the public parameter $\mathtt{pp} = \{\mathtt{k}_0, \mathtt{k}_1, \mathtt{k}_2, \mathtt{N}\}$, the secret key $\mathtt{sk} = \{\mathtt{p}, \mathtt{L}\}$, and the message space $\mathcal{M} = \{\mathtt{m} | \mathtt{m} \in [-2^{\mathtt{k}_1 - 1}, 2^{\mathtt{k}_1 - 1})\}$.
- $\mathsf{Enc}(\mathtt{m}, \mathtt{sk})$: On input a message $\mathtt{m} \in \mathcal{M}$, the encryptor first chooses two random numbers $\mathtt{r} \in \{0,1\}^{\mathtt{k}_2}$ and $\mathtt{r}' \in \{0,1\}^{\mathtt{k}_0}$. Then, it encrypts $\mathtt{m}$ as $[\![\mathtt{m}]\!] = (\mathtt{r} * \mathtt{L} + \mathtt{m})(1 + \mathtt{r}' * \mathtt{p}) \mod \mathtt{N}$.
- $\mathsf{Dec}(\mathtt{sk}, [\![\mathtt{m}]\!])$: A ciphertext $[\![\mathtt{m}]\!]$ is decrypted by i) calculating $\mathtt{m}' = ([\![\mathtt{m}]\!] \mod \mathtt{p}) \mod \mathtt{L}$; and ii) setting $\mathtt{m} = \mathtt{m}'$ if $\mathtt{m}' < \frac{\mathtt{L}}{2}$ and $\mathtt{m} = \mathtt{m}' - \mathtt{L}$ otherwise.

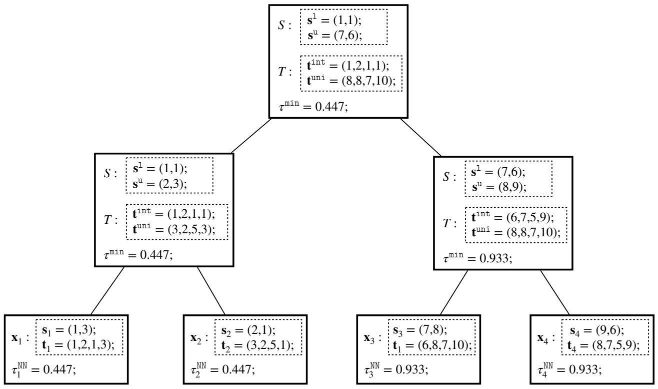

S :
$\mathbf{s}^{\mathtt{l}} = (1,1)$;
$\mathbf{s}^{\mathtt{u}} = (7,6)$;
T :
$\mathbf{t}^{\mathtt{int}} = (1,2,1,1)$;
$\mathbf{t}^{\mathtt{uni}} = (8,8,7,10)$;
$\tau^{\mathtt{min}} = 0.447$;
S :
$\mathbf{s}^{\mathtt{l}} = (1,1)$;
$\mathbf{s}^{\mathtt{u}} = (2,3)$;
T :
$\mathbf{t}^{\mathtt{int}} = (1,2,1,1)$;
$\mathbf{t}^{\mathtt{uni}} = (3,2,5,3)$;
$\tau^{\mathtt{min}} = 0.447$;
S :
$\mathbf{s}^{\mathtt{l}} = (7,6)$;
$\mathbf{s}^{\mathtt{u}} = (8,9)$;
T :
$\mathbf{t}^{\mathtt{int}} = (6,7,5,9)$;
$\mathbf{t}^{\mathtt{uni}} = (8,8,7,10)$;
$\tau^{\mathtt{min}} = 0.933$;
$\mathbf{x}_1$:
$\mathbf{s}_1 = (1,3)$;
$\mathbf{t}_1 = (1,2,1,3)$;
$\tau_1^{\mathtt{NN}} = 0.447$;
$\mathbf{x}_2$:
$\mathbf{s}_2 = (2,1)$;
$\mathbf{t}_2 = (3,2,5,1)$;
$\tau_2^{\mathtt{NN}} = 0.447$;
$\mathbf{x}_3$:
$\mathbf{s}_3 = (7,8)$;
$\mathbf{t}_1 = (6,8,7,10)$;
$\tau_3^{\mathtt{NN}} = 0.933$;
$\mathbf{x}_4$:
$\mathbf{s}_4 = (9,6)$;
$\mathbf{t}_4 = (8,7,5,9)$;
$\tau_4^{\mathtt{NN}} = 0.933$;

Fig. 3. Example of MIUR-tree on a dataset with $\alpha = 0.4$

The SHE scheme supports homomorphic addition and multiplication as: (i) Add-I: $([\![\mathtt{m}_1]\!] + [\![\mathtt{m}_2]\!]) \mod \mathtt{N} \rightarrow [\![\mathtt{m}_1 + \mathtt{m}_2]\!]$; (ii) Add-II: $([\![\mathtt{m}_1]\!] + \mathtt{m}_2) \mod \mathtt{N} \rightarrow [\![\mathtt{m}_1 + \mathtt{m}_2]\!]$; (iii) Multiply-I: $[\![\mathtt{m}_1]\!] * [\![\mathtt{m}_2]\!] \mod \mathtt{N} \rightarrow [\![\mathtt{m}_1 * \mathtt{m}_2]\!]$; and (iv) Multiply-II: $[\![\mathtt{m}_1]\!] * \mathtt{m}_2 \mod \mathtt{N} \rightarrow [\![\mathtt{m}_1 * \mathtt{m}_2]\!]$ $(\mathtt{m}_2 > 0)$. Meanwhile, the maximum number of multiply-I operations, i.e., multiplicative depth, is limited by the security parameters and is up to $\lfloor \frac{\mathtt{k}_0}{2\mathtt{k}_2} \rfloor - 1$. In addition, by leveraging homomorphic properties, the SHE scheme can encrypt messages in a public-key manner. The public key is $\mathtt{pk} = \{\mathtt{pp}, [\![0]\!]_1, [\![0]\!]_2\}$, where $[\![0]\!]_1$ and $[\![0]\!]_2$ are generated by encrypting zero twice. Then, a message is encrypted by $\mathtt{pk}$ as $[\![\mathtt{m}]\!] \leftarrow (\mathtt{m} + \mathtt{r}_1 * [\![0]\!]_1 + \mathtt{r}_2 * [\![0]\!]_2) \mod \mathtt{N}$, where $\mathtt{r}_1, \mathtt{r}_2 \in \{0,1\}^{\mathtt{k}_2}$. It is worth noting that the ciphertexts encrypted in the above public-key manner also satisfy the semantic security, as proved in [31].

4 Building Blocks

In this section, we introduce an index to organize the spatial and feature data, denoted by MIUR-tree, and design an MIUR-tree based RkNN query algorithm in a filter and refinement framework. Then, we propose private filter/refinement protocols to protect the privacy of the RkNN query algorithm.

4.1 MIUR-Tree

The MIUR-tree is designed based on the IUR-tree in [28] and used for indexing spatial and feature data. The core idea of MIUR-tree is to recursively group similar records until all records are grouped into one root node. In the MIUR-tree, there are two types of nodes, i.e., leaf nodes and internal nodes. Each leaf node has two attributes $(\mathbf{x}_i, \{\tau_i^{k\mathtt{NN}}\}_{k=1}^K)$, where (i) $\mathbf{x}_i$ is a record; (ii) $\tau_i^{k\mathtt{NN}}$ is the similarity between $\mathbf{x}_i$ and its k-th nearest neighbor for $1 \leq k \leq K$; and (iii) K is the maximum value of k. Each internal node has four attributes $E = (S, T, \{\tau^{k\mathtt{min}}\}_{k=1}^K, children)$. The attribute S is a minimum bounding rectangle (MBR) covering all spatial records rooted at the current node. It has two attributes $\mathbf{s}^{\mathtt{l}}$ and $\mathbf{s}^{\mathtt{u}}$ that respectively denote the lower bound and upper bound of S. The attribute T is an intersection-union record and includes an intersection vector $\mathbf{t}^{\mathtt{int}}$ and a union vector $\mathbf{t}^{\mathtt{uni}}$ that respectively denote the minimum vector and maximum vector of all feature vectors rooted at the current node. For example, given two feature vectors $\mathbf{t}_i = (1,3,2,8)$

and $\mathbf{t}_j = (1, 1, 7, 9)$, the corresponding grouped intersection vector $\mathbf{t}^{\texttt{int}}$ and union vector $\mathbf{t}^{\texttt{uni}}$ will be $\mathbf{t}^{\texttt{int}} = (1, 1, 2, 8)$ and $\mathbf{t}^{\texttt{uni}} = (1, 3, 7, 9)$. The attribute $\tau^{k\texttt{min}}$ denotes the minimum of all similarities between each record rooted at the current node and its k-th nearest neighbor. The attribute $children$ contains a set of pointers pointing to the child nodes of the current node.

Given a dataset $\mathcal{D} = \{\mathbf{x}_i = (\mathbf{s}_i, \mathbf{t}_i)\}_{i=1}^{n}$, the corresponding MIUR-tree is built by recursively grouping similar records. At the beginning, each leaf node is regarded as an internal node with a single record. After that, internal nodes can be recursively grouped together, where the similarity between two internal nodes $E_i = (S_i = (\mathbf{s}_i^{\texttt{l}}, \mathbf{s}_i^{\texttt{u}}), T_i = (\mathbf{t}_i^{\texttt{int}}, \mathbf{t}_i^{\texttt{uni}}), \{\tau_i^{k\texttt{min}}\}_{k=1}^{K}, children_i)$ and $E_j = (S_j = (\mathbf{s}_j^{\texttt{l}}, \mathbf{s}_j^{\texttt{u}}), T_j = (\mathbf{t}_j^{\texttt{int}}, \mathbf{t}_j^{\texttt{uni}}), \{\tau_j^{k\texttt{min}}\}_{k=1}^{K}, children_j)$ is calculated as $Sim(E_i, E_j) = \alpha * (1 - \frac{D(\mathbf{s}_i^{\texttt{l}}, \mathbf{s}_j^{\texttt{l}}) + D(\mathbf{s}_i^{\texttt{u}}, \mathbf{s}_j^{\texttt{u}}) - 2\phi_s}{\psi_t - \phi_t}) + (1 - \alpha) * \frac{J(\mathbf{t}_i^{\texttt{int}}, \mathbf{t}_j^{\texttt{int}}) + J(\mathbf{t}_i^{\texttt{uni}}, \mathbf{t}_j^{\texttt{uni}}) - 2\phi_t}{\psi_t - \phi_t}$. In Fig. 3, we give an example to show the structure of an MIUR-tree built based on a dataset with four records.

4.2 MIUR-Tree Based RkNN Query Algorithm

The MIUR-tree can efficiently support spatial-feature-based RkNN queries. Let $\mathcal{T}$ be an MIUR-tree built based on a dataset $\mathcal{D} = \{\mathbf{x}_i = (\mathbf{s}_i, \mathbf{t}_i)\}_{i=1}^{n}$. Then, given a spatial-feature-based RkNN query $(\mathbf{q}, k)$, we can search on $\mathcal{D}$ for the query result. The query algorithm is designed in a filter and refinement framework, as shown in Alg. 1. Detailed filter and refinement stages are shown as follows.

• ***Filter Stage:*** In this stage, the searcher searches on the MIUR-tree for candidate results. Based on the searched node, two cases will be considered.

Case 1. When an internal node $E = (S, T, \{\tau^{k\texttt{min}}\}_{k=1}^{K}, children)$ is searched, the searcher computes the maximum similarity between E and $\mathbf{q} = (\mathbf{s}_q, \mathbf{t}_q)$ as $MaxSim(E, \mathbf{q}) = \alpha * (1 - \frac{MinD(S, \mathbf{s}_q) - \phi_s}{\psi_s - \phi_s}) + (1 - \alpha) * \frac{MaxJ(T, \mathbf{t}_q) - \phi_t}{\psi_t - \phi_t}$, where (i) $MinD(S, \mathbf{s}_q)$ denotes the minimum Euclidean distance between S and $\mathbf{s}_q$; and (ii) $MaxJ(T, \mathbf{t}_q)$ denotes the maximum extended Jaccard similarity between T and $\mathbf{t}_q$. Meanwhile, as introduced in [32] and [28], $MinD(S, \mathbf{s}_q)$ and $MaxJ(T, \mathbf{t}_q)$ are respectively computed as $MinD(S, \mathbf{s}_q) = \sqrt{\sum_{l=1}^{2}(s_l - s_{q,l})^2}$ and $MaxJ(T, \mathbf{t}_q) = \frac{\sum_{l=1}^{d} t_l * t_{q,l}}{\|\mathbf{t}_i\|^2 + \|\mathbf{t}_q\|^2 - \sum_{l=1}^{d} t_l * t_{q,l}}$, where

$$s_l = \begin{cases} s_l^{\texttt{l}} & s_{q,l} < s_l^{\texttt{l}} \\ s_l^{\texttt{r}} & s_{q,l} > s_l^{\texttt{r}} \\ s_{q,l} & \text{Otherwise;} \end{cases} \qquad t_l = \begin{cases} t_l^{\texttt{int}} & t_{q,l} < t_l^{\texttt{int}} \\ t_l^{\texttt{uni}} & t_{q,l} > t_l^{\texttt{uni}} \\ t_{q,l} & \text{Otherwise.} \end{cases} \tag{1}$$

If $MaxSim(E, \mathbf{q}) < \tau^{k\texttt{min}}$, the maximum similarity between $\mathbf{q}$ and all records in the current node is less than the minimum similarity of these records with their k-th nearest neighbors. Thus, the internal node cannot contain the query result and will be pruned by the searcher. Otherwise, if $MaxSim(E, \mathbf{q}) \geq \tau^{k\texttt{min}}$, the searcher will continue to search each child of the current node.

Case 2. When a leaf node with $(\mathbf{x}_i, \{\tau_i^{\texttt{kNN}}\}_{k=1}^{K})$ is searched, the searcher directly adds it to the candidate result as $\mathcal{C} = \mathcal{C} \cup \{(\mathbf{x}_i, \tau_i^{\texttt{kNN}})\}$.

Algorithm 1 Spatial-feature-based RkNN query

```
Input: The MIUR-tree $\mathcal{T}$ and the query request $(\mathbf{q}, k)$;
Output: The query result $\mathcal{R}$;
    ▷ Filter stage ◁
    Set $\mathcal{C} = \emptyset$;                // Initialize the candidate result
    FILTER($\mathcal{T}.root$, $(\mathbf{q}, k)$, $\mathcal{C}$);
    ▷ Refinement stage ◁
    Set $\mathcal{R} = \emptyset$;                // Initialize the final query result
    for each $\mathbf{x}_i \in \mathcal{C}$ do
        if $Sim(\mathbf{x}_i, \mathbf{q}) \geq \tau_i^{k\texttt{NN}}$ then
            $\mathcal{R} = \mathcal{R} \cup \{\mathbf{x}_i\}$;
    return $\mathcal{R}$;
    function FILTER(Node $node$, Query request $(\mathbf{q}, k)$, Candidate result $\mathcal{C}$)
        if $node$ is an internal node then
            if $MaxSim(E, \mathbf{q}) \geq \tau^{k\texttt{min}}$ then
                for each $child \in node.children$ do
                    FILTER($child$, $(\mathbf{q}, k)$, $\mathcal{C}$);
        else if $node$ is a leaf node then
            $\mathcal{C} = \mathcal{C} \cup \{(\mathbf{x}_i, \tau_i^{k\texttt{NN}})\}$;
```

• ***Refinement Stage:*** In the refinement stage, the searcher verifies whether each candidate $(\mathbf{x}_i, \{\tau_i^{\texttt{kNN}}\}_{k=1}^{K}) \in \mathcal{C}$ satisfies $Sim(\mathbf{x}_i, \mathbf{q}) \geq \tau_i^{k\texttt{NN}}$ or not. If yes, $\mathbf{q}$ is among the kNN of $\mathbf{x}_i$, and $\mathbf{x}_i$ will be added to the final query result, i.e., $\mathcal{R} = \mathcal{R} \cup \{\mathbf{x}_i\}$. Otherwise, $\mathbf{x}_i$ does not satisfy the query request.

Basic Operations. From Alg. 1, we can observe that the basic operations of the refinement and filter stages are the determination of two inequalities as

$$Sim(\mathbf{x}_i, \mathbf{q}) \geq \tau_i^{k\texttt{NN}} \text{ and } MaxSim(E, \mathbf{q}) \geq \tau^{k\texttt{min}}.$$

That is,

$$\begin{cases} \alpha * (1 - \frac{D(\mathbf{s}_i, \mathbf{s}_q) - \phi_s}{\psi_s - \phi_s}) + (1 - \alpha) * \frac{J(\mathbf{t}_i, \mathbf{t}_q) - \phi_t}{\psi_t - \phi_t} \geq \tau_i^{k\texttt{NN}} \\ \alpha * (1 - \frac{MinD(S, \mathbf{s}_q) - \phi_s}{\psi_s - \phi_s}) + (1 - \alpha) * \frac{MaxJ(T, \mathbf{t}_q) - \phi_t}{\psi_t - \phi_t} \geq \tau^{k\texttt{min}}. \end{cases}$$

Since the SHE scheme can only support addition and multiplication operations between integers, we first transform these two inequalities into ones that can be determined by the SHE scheme. Specifically, the inequality $\alpha * (1 - \frac{D(\mathbf{s}_i, \mathbf{s}_q) - \phi_s}{\psi_s - \phi_s}) + (1 - \alpha) * \frac{J(\mathbf{t}_i, \mathbf{t}_q) - \phi_t}{\psi_t - \phi_t} \geq \tau_i^{k\texttt{NN}}$ is equivalent to the following inequality

$$\begin{aligned} &\alpha * \psi_s * (\psi_t - \phi_t) - (1 - \alpha) * \phi_t * (\psi_s - \phi_s) - \tau_i^{k\texttt{NN}} * \\ &(\psi_s - \phi_s) * (\psi_t - \phi_t) + (1 - \alpha) * (\psi_s - \phi_s) * J(\mathbf{t}_i, \mathbf{t}_q) \\ &\geq \alpha * (\psi_t - \phi_t) * D(\mathbf{s}_i, \mathbf{s}_q). \end{aligned} \tag{2}$$

If we let $a_i^{k\texttt{NN}} = \alpha * \psi_s * (\psi_t - \phi_t) - (1 - \alpha) * \phi_t * (\psi_s - \phi_s) - \tau_i^{k\texttt{NN}} * (\psi_s - \phi_s) * (\psi_t - \phi_t)$, $b = (1 - \alpha) * (\psi_s - \phi_s)$, and $c = \alpha * (\psi_t - \phi_t)$, we have Eq. (2) is equivalent to $a_i^{k\texttt{NN}} + b * J(\mathbf{t}_i, \mathbf{t}_q) \geq c * D(\mathbf{s}_i, \mathbf{s}_q)$. That is, $a_i^{k\texttt{NN}} + \frac{b * \sum_{l=1}^{d} t_{i,l} * t_{q,l}}{\|\mathbf{t}_i\|^2 + \|\mathbf{t}_q\|^2 - \sum_{l=1}^{d} t_{i,l} * t_{q,l}} \geq c * D(\mathbf{s}_i, \mathbf{s}_q)$. Then, we can further infer that $a_i^{k\texttt{NN}} * (\|\mathbf{t}_i\|^2 + \|\mathbf{t}_q\|^2) + (b - a_i^{k\texttt{NN}}) * \sum_{l=1}^{d} t_{i,l} * t_{q,l}$
$\geq c * (\|\mathbf{t}_i\|^2 + \|\mathbf{t}_q\|^2 - \sum_{l=1}^{d} t_{i,l} * t_{q,l}) * Dis(\mathbf{s}_i, \mathbf{s}_q)$. If we let $L_{\mathbf{x}_i} = a_i^{k\texttt{NN}} * (\|\mathbf{t}_i\|^2 + \|\mathbf{t}_q\|^2) + (b - a_i^{k\texttt{NN}}) * \sum_{l=1}^{d} t_{i,l} * t_{q,l}$ and $R_{\mathbf{x}_i} = c * (\|\mathbf{t}_i\|^2 + \|\mathbf{t}_q\|^2 - \sum_{l=1}^{d} t_{i,l} * t_{q,l}) * Dis(\mathbf{s}_i, \mathbf{s}_q)$, the filter condition will become $L_{\mathbf{x}_i} \geq R_{\mathbf{x}_i}$. Since $R_{\mathbf{x}_i}$ is obliviously no less than zero, i.e., $R_{\mathbf{x}_i} \geq 0$, we have

$$L_{\mathbf{x}_i} \geq R_{\mathbf{x}_i} \Leftrightarrow L_{\mathbf{x}_i} \geq 0 \text{ and } L_{\mathbf{x}_i}^2 \geq R_{\mathbf{x}_i}^2. \tag{3}$$

The reason we do this transformation is that executing the square root computation in $Dis(\mathbf{s}_i, \mathbf{s}_q) = \sqrt{\sum_{l=1}^{2}(s_{i,l} - s_{q,l})^2}$ is hard over encrypted data. After the transformation, all values in Eq. (3) can be scaled to integers easily such that we can apply the SHE scheme to achieve the determination of Eq. (3), namely the determination of $Sim(\mathbf{x}_i, \mathbf{q}) \geq \tau_i^{k\mathtt{NN}}$.

Similarly, if we let $a_i^{k\mathtt{min}} = \alpha * \psi_s * (\psi_t - \phi_t) - (1-\alpha) * \phi_t * (\psi_s - \phi_s) - \tau_i^{k\mathtt{min}} * (\psi_s - \phi_s) * (\psi_t - \phi_t)$, $L_E = a^{k\mathtt{min}} * (\|\mathbf{t}\|^2 + \|\mathbf{t}_q\|^2) + (b - a^{k\mathtt{min}}) * \sum_{l=1}^{d} t_l * t_{q,l}$, and $R_E = c * (\|\mathbf{t}\|^2 + \|\mathbf{t}_q\|^2 - \sum_{l=1}^{d} t_l * t_{q,l}) * \sqrt{\sum_{l=1}^{2}(s_l - s_{q,l})^2}$, the determination of $MaxSim(E, \mathbf{q}) \geq \tau^{k\mathtt{min}}$ can be transformed into $L_E \geq 0$ and $L_E^2 \geq R_E^2$, where (i) $\mathbf{t} = (t_1, t_2, \cdots, t_d)$; and (ii) s_l and t_l are set as Eq. (1).

4.3 Private Protocols

In this subsection, we design a private refinement protocol and a private filter protocol to determine the following inequalities.

$$\begin{cases} Sim(\mathbf{x}_i, \mathbf{q}) \geq \tau_i^{k\mathtt{NN}} \Leftrightarrow L_{\mathbf{x}_i} \geq 0 \text{ and } L_{\mathbf{x}_i}^2 \geq R_{\mathbf{x}_i}^2 \\ MaxSim(E, \mathbf{q}) \geq \tau^{k\mathtt{min}} \Leftrightarrow L_E \geq 0 \text{ and } L_E^2 \geq R_E^2. \end{cases}$$

Private Refinement Protocol. The private refinement protocol is used to determine whether $Sim(\mathbf{x}_i, \mathbf{q}) \geq \tau_i^{k\mathtt{NN}}$. The protocol is executed by the two servers S1 and S2, where S1 has the SHE ciphertexts of $(\mathbf{x}_i, \{\tau_i^{k\mathtt{NN}}\}_{k=1}^{K})$ and $\mathbf{q}$, and S2 has the corresponding secret key used for encrypting the data. Specifically, $(\mathbf{x}_i = (\mathbf{s}_i, \mathbf{t}_i), \{\tau_i^{k\mathtt{NN}}\}_{k=1}^{K})$ is encrypted into a collection of ciphertexts as

$$\begin{aligned} \mathtt{Cipher}_{\mathbf{x}_i} \leftarrow & \{\{\llbracket a_i^{k\mathtt{NN}} \rrbracket, \llbracket b - a_i^{k\mathtt{NN}} \rrbracket\}_{k=1}^{K}, \llbracket c^2 \rrbracket, \llbracket \mathbf{s}_i^2 \rrbracket, \llbracket \mathbf{s}_i \rrbracket, \\ & \llbracket \mathbf{t}_i \rrbracket, \llbracket \|\mathbf{t}_i\|^2 \rrbracket\}. \end{aligned} \tag{4}$$

Meanwhile, $\mathbf{q} = (\mathbf{s}_q, \mathbf{t}_q)$ is encrypted into ciphertexts as

$$\mathtt{Token}_{\mathtt{Ref}} \leftarrow \{\llbracket \mathbf{s}_q^2 \rrbracket, \llbracket -2\mathbf{s}_q \rrbracket, \llbracket \mathbf{t}_q \rrbracket, \llbracket \|\mathbf{t}_q\|^2 \rrbracket\}. \tag{5}$$

Then, S1 with the ciphertexts $\{\mathtt{Cipher}_{\mathbf{x}_i}, \mathtt{Token}_{\mathtt{Ref}}, k\}$ and S2 with the secret key $\mathtt{sk}$ collaboratively determine whether $Sim(\mathbf{x}_i, \mathbf{q}) \geq \tau_i^{k\mathtt{NN}}$ as follows.

Step 1. S1 uses homomorphic properties to compute

$$\begin{cases} \llbracket L_{\mathbf{x}_i} \rrbracket \leftarrow & (\llbracket a_i^{k\mathtt{NN}} \rrbracket * (\llbracket \|\mathbf{t}_i\|^2 \rrbracket + \llbracket \|\mathbf{t}_q\|^2 \rrbracket) + \llbracket b - a_i^{k\mathtt{NN}} \rrbracket * \\ & \sum_{l=1}^{d} \llbracket t_{i,l} \rrbracket * \llbracket t_{q,l} \rrbracket) \bmod \mathtt{N} \\ \llbracket R_{\mathbf{x}_i}^2 \rrbracket \leftarrow & \llbracket c^2 \rrbracket * (\llbracket \|\mathbf{t}_i\|^2 \rrbracket + \llbracket \|\mathbf{t}_q\|^2 \rrbracket + \llbracket -1 \rrbracket * \sum_{l=1}^{d} \llbracket t_{i,l} \rrbracket * \\ & \llbracket t_{q,l} \rrbracket)^2 * \sum_{l=1}^{2} (\llbracket s_{i,l}^2 \rrbracket + \llbracket s_{i,l} \rrbracket * \llbracket -2 * s_{q,l} \rrbracket + \\ & \llbracket s_{q,l}^2 \rrbracket) \quad \bmod \mathtt{N}. \end{cases}$$

Then, S1 computes

$$\begin{cases} \llbracket y_i^L \rrbracket \leftarrow & \llbracket f_i^{L,\mathtt{tag}} \rrbracket * (r_{i,1}^L * \llbracket L_{\mathbf{x}_i} \rrbracket + r_{i,2}^L) \quad \bmod \mathtt{N} \\ \llbracket y_i^R \rrbracket \leftarrow & \llbracket f_i^{R,\mathtt{tag}} \rrbracket * (r_{i,1}^R * (\llbracket L_{\mathbf{x}_i} \rrbracket^2 + \llbracket -1 \rrbracket * \llbracket R_{\mathbf{x}_i}^2 \rrbracket) + \\ & r_{i,2}^R) \bmod \mathtt{N}, \end{cases}$$

where (i) $f_i^{L,\mathtt{tag}}$ and $f_i^{R,\mathtt{tag}}$ are either 1 or -1; and (ii) $r_{i,1}^L, r_{i,2}^L, r_{i,1}^R, r_{i,2}^R \in \mathcal{M}$ satisfying $r_{i,1}^L > r_{i,2}^L > 0$ and $r_{i,1}^R > r_{i,2}^R > 0$. Then, S1 sends $\{\llbracket y_i^L \rrbracket, \llbracket y_i^R \rrbracket\}$ to S2.

Step 2. On receiving $\{\llbracket y_i^L \rrbracket, \llbracket y_i^R \rrbracket\}$, S2 recovers y_i^L and y_i^R using the secret key $\mathtt{sk}$ and constructs two flags f_i^L and f_i^R as

$$f_i^L = \begin{cases} 1 & y_i^L > 0 \\ -1 & y_i^L < 0; \end{cases} \qquad f_i^R = \begin{cases} 1 & y_i^R > 0 \\ -1 & y_i^R < 0. \end{cases} \tag{6}$$

After that, S2 sends f_i^L and f_i^R to S1. If $f_i^L * f_i^{L,\mathtt{tag}} = 1$ and $f_i^R * f_i^{R,\mathtt{tag}} = 1$, it denotes $Sim(\mathbf{x}_i, \mathbf{q}) \geq \tau_i^{k\mathtt{NN}}$. Otherwise, it denotes $Sim(\mathbf{x}_i, \mathbf{q}) < \tau_i^{k\mathtt{NN}}$.

Correctness. The private refinement protocol is correct if $f_i^L * f_i^{L,\mathtt{tag}} = 1$ and $f_i^R * f_i^{R,\mathtt{tag}} = 1$ are equivalent to $Sim(\mathbf{x}_i, \mathbf{q}) \geq \tau_i^{k\mathtt{NN}}$. Actually, in our scheme, $f_i^L * f_i^{L,\mathtt{tag}} = 1 \Leftrightarrow L_{\mathbf{x}_i} \geq 0$ and $f_i^R * f_i^{R,\mathtt{tag}} = 1 \Leftrightarrow L_{\mathbf{x}_i}^2 \geq R_{\mathbf{x}_i}^2$. Specifically, when $f_i^{L,\mathtt{tag}} = 1$, we have $f_i^L * f_i^{L,\mathtt{tag}} = 1 \Leftrightarrow f_i^L = 1$, and it means that $y_i^L > 0$. Since $r_{i,1}^L > r_{i,2}^L > 0$, we have $y_i^L > 0 \Leftrightarrow f_i^{L,\mathtt{tag}} * (r_{i,1}^L * L_{\mathbf{x}_i} + r_{i,2}^L) > 0 \Leftrightarrow L_{\mathbf{x}_i} \geq 0$. In the same way, when $f_i^{L,\mathtt{tag}} = -1$, we have $f_i^L * f_i^{L,\mathtt{tag}} = 1 \Leftrightarrow f_i^L = -1$ and can further infer that $L_{\mathbf{x}_i} \geq 0$. Thus, $f_i^L * f_i^{L,\mathtt{tag}} = 1 \Leftrightarrow L_{\mathbf{x}_i} \geq 0$. Similarly, we can prove that $f_i^R * f_i^{R,\mathtt{tag}} = 1 \Leftrightarrow L_{\mathbf{x}_i}^2 \geq R_{\mathbf{x}_i}^2$. Hence, the protocol is correct.

Private Filter Protocol. The private filter protocol is used to determine whether $MaxSim(E, \mathbf{q}) \geq \tau^{k\mathtt{min}}$. The protocol is executed by the two servers S1 and S2, where S1 has the SHE ciphertexts of E and $\mathbf{q}$, and S2 has the corresponding secret key used for encrypting the data. Specifically, E is in the form of $E = (S = (\mathbf{s}^{\mathtt{l}}, \mathbf{s}^{\mathtt{r}}), T = (\mathbf{t}^{\mathtt{int}}, \mathbf{t}^{\mathtt{uni}}), \{\tau^{k\mathtt{min}}\}_{k=1}^{K}, children)$ and is encrypted into a collection of ciphertexts by the SHE scheme as

$$\begin{aligned} \mathtt{Cipher}_E \leftarrow & \{\{\llbracket a^{k\mathtt{min}} \rrbracket, \llbracket b - a^{k\mathtt{min}} \rrbracket\}_{k=1}^{K}, \llbracket c^2 \rrbracket, \llbracket \mathbf{s}^{\mathtt{l}} \rrbracket, \\ & \llbracket \mathbf{s}^{\mathtt{r}} \rrbracket, \llbracket \mathbf{t}^{\mathtt{int}} \rrbracket, \llbracket \mathbf{t}^{\mathtt{uni}} \rrbracket\}. \end{aligned} \tag{7}$$

Meanwhile, $\mathbf{q} = (\mathbf{s}_q, \mathbf{t}_q)$ is encrypted into ciphertexts as

$$\mathtt{Token}_{\mathtt{Fil}} \leftarrow \{\llbracket \mathbf{s}_q^2 \rrbracket, \llbracket -2\mathbf{s}_q \rrbracket, \llbracket \mathbf{s}_q \rrbracket, \llbracket \mathbf{t}_q \rrbracket, \llbracket \|\mathbf{t}_q\|^2 \rrbracket\}. \tag{8}$$

Then, S1 with the ciphertexts $\{\mathtt{Cipher}_E, \mathtt{Token}_{\mathtt{Fil}}, k\}$ and S2 with the secret key $\mathtt{sk}$ determine whether $MaxSim(E, \mathbf{q}) \geq \tau^{k\mathtt{min}}$ as the following steps.

Step 1. S1 and S2 privately determine $\llbracket \mathbf{s} \rrbracket = (\llbracket s_1 \rrbracket, \llbracket s_2 \rrbracket)$ as Eq. (1), where

$$s_l = \begin{cases} s_l^{\mathtt{l}} & s_{q,l} < s_l^{\mathtt{l}} \\ s_l^{\mathtt{r}} & s_{q,l} > s_l^{\mathtt{r}} \\ s_{q,l} & \text{Otherwise}, \end{cases} \tag{9}$$

for $l = 1, 2$ as the following steps. (1) S1 uses homomorphic properties to compute

$$\llbracket z_l^{\mathtt{l}} \rrbracket \leftarrow \llbracket f_l^{\mathtt{l},\mathtt{tag}} \rrbracket * (r_{l,1}^{\mathtt{l}} * (\llbracket s_l^{\mathtt{l}} \rrbracket + \llbracket -1 \rrbracket * \llbracket s_{q,l} \rrbracket) - r_{l,2}^{\mathtt{l}}) \bmod \mathtt{N}$$
$$\llbracket z_l^{\mathtt{r}} \rrbracket \leftarrow \llbracket f_l^{\mathtt{r},\mathtt{tag}} \rrbracket * (r_{l,1}^{\mathtt{r}} * (\llbracket s_{q,l} \rrbracket + \llbracket -1 \rrbracket * \llbracket s_l^{\mathtt{r}} \rrbracket) - r_{l,2}^{\mathtt{r}}) \bmod \mathtt{N},$$

where (i) $f_l^{\mathtt{l},\mathtt{tag}}$ and $f_l^{\mathtt{r},\mathtt{tag}}$ are set to be either 1 or -1 by flipping a coin; and (ii) $\{r_{l,1}^{\mathtt{l}}, r_{l,2}^{\mathtt{l}}, r_{l,1}^{\mathtt{r}}, r_{l,2}^{\mathtt{l}}\} \in \mathcal{M}$ with $r_{l,1}^{\mathtt{l}} > r_{l,2}^{\mathtt{l}} > 0$ and $r_{l,1}^{\mathtt{r}} > r_{l,2}^{\mathtt{r}} > 0$. Then, S1 sends $\{\llbracket z_l^{\mathtt{l}} \rrbracket, \llbracket z_l^{\mathtt{r}} \rrbracket\}$ to S2.

(2) On receiving $\{\llbracket z_l^{\mathtt{l}} \rrbracket, \llbracket z_l^{\mathtt{r}} \rrbracket\}$, S2 first uses the secret key $\mathtt{sk}$ to recover $\{z_l^{\mathtt{l}}, z_l^{\mathtt{r}}\}$ and constructs two flags as

$$h_l^{\mathtt{l}} = \begin{cases} 1 & z_l^{\mathtt{l}} > 0 \\ 0 & z_l^{\mathtt{l}} < 0, \end{cases} \qquad h_l^{\mathtt{r}} = \begin{cases} 1 & z_l^{\mathtt{r}} > 0 \\ 0 & z_l^{\mathtt{l}} < 0. \end{cases}$$

Then, S2 returns $\{\llbracket h_l^{\mathtt{l}}\rrbracket, \llbracket h_l^{\mathtt{r}}\rrbracket\}$ to S1.

(3) On receiving $\{\llbracket h_l^{\mathtt{l}}\rrbracket, \llbracket h_l^{\mathtt{r}}\rrbracket\}$, S1 computes $\llbracket f_l^{\mathtt{l}}\rrbracket = \llbracket h_l^{\mathtt{l}}\rrbracket * \llbracket f_l^{\mathtt{l,tag}}\rrbracket$ and $\llbracket f_l^{\mathtt{r}}\rrbracket = \llbracket h_l^{\mathtt{r}}\rrbracket * \llbracket f_l^{\mathtt{r,tag}}\rrbracket$. Then, it computes $\llbracket s_l\rrbracket$ as $\llbracket s_l\rrbracket \leftarrow (\llbracket f_l^{\mathtt{l}}\rrbracket * \llbracket s_l^{\mathtt{l}}\rrbracket + (1 + \llbracket -1\rrbracket * \llbracket f_l^{\mathtt{l}}\rrbracket) * (\llbracket f_l^{\mathtt{r}}\rrbracket * \llbracket s_l^{\mathtt{r}}\rrbracket + (1 + \llbracket -1\rrbracket * \llbracket f_l^{\mathtt{r}}\rrbracket) * \llbracket s_{q,l}\rrbracket)) \mod \mathtt{N}$.

Step 2. S1 and S2 privately determine $\llbracket \mathbf{t}\rrbracket = (\llbracket t_1\rrbracket, \llbracket t_2\rrbracket, \cdots, \llbracket t_d\rrbracket)$ as Eq. (1). It is easy to find that the determination of each $\llbracket t_l\rrbracket$ can be achieved in the same way as *Step 1* for $1 \le l \le d$. Due to the page limitation, we omit the details here. After this step, S1 obtains the ciphertexts $\{\{\llbracket a^{k\mathtt{min}}\rrbracket, \llbracket b - a^{k\mathtt{min}}\rrbracket\}_{k=1}^K, \llbracket c^2\rrbracket, \llbracket \mathbf{s}\rrbracket, \llbracket \mathbf{t}\rrbracket\}$. Then, it further uses homomorphic properties to compute $\llbracket \mathbf{s}^2\rrbracket$ and $\llbracket ||\mathbf{t}||^2\rrbracket$.

Step 3. On inputs $\{\llbracket a^{k\mathtt{min}}\rrbracket, \llbracket b - a^{k\mathtt{min}}\rrbracket, \llbracket c^2\rrbracket, \llbracket \mathbf{s}^2\rrbracket, \llbracket \mathbf{s}\rrbracket, \llbracket \mathbf{t}\rrbracket, \llbracket ||\mathbf{t}||^2\rrbracket\}$, S1 and S2 call the private refinement protocol to determine whether the similarity between $(\mathbf{s}, \mathbf{t})$ and $\mathbf{q}$ is equal to or larger than $a^{k\mathtt{min}}$. If yes, it means that $MaxSim(E, \mathbf{q}) \ge \tau^{k\mathtt{min}}$. Otherwise, it means that $MaxSim(E, \mathbf{q}) < \tau^{k\mathtt{min}}$.

Correctness. The private filter protocol is correct. First, based on the proof of private refinement protocol, we can see that $\llbracket f_l^{\mathtt{l,tag}}\rrbracket$ and $\llbracket f_l^{\mathtt{r,tag}}\rrbracket$ are used for preventing S2 knowing the true sign of $\{z_l^{\mathtt{l}}, z_l^{\mathtt{r}}\}$. Thus, we set them as $f_l^{\mathtt{l,tag}} = f_l^{\mathtt{r,tag}} = 1$ in the correctness proof of the filter protocol by default. In this case, when $s_{q,l} < s_l^{\mathtt{l}}$, we have $s_{q,l} < s_l^{\mathtt{r}}$ due to $s_l^{\mathtt{l}} < s_l^{\mathtt{r}}$. Then, from $r_{l,1}^{\mathtt{l}} > r_{l,2}^{\mathtt{l}} > 0$ and $r_{l,1}^{\mathtt{r}} > r_{l,2}^{\mathtt{r}} > 0$, we can infer that $z_l^{\mathtt{l}} > 0$ and $z_l^{\mathtt{r}} < 0$. Meanwhile, we have $f_l^{\mathtt{l}} = h_l^{\mathtt{l}} = 1$ and $f_l^{\mathtt{r}} = h_l^{\mathtt{r}} = 0$. Then, based on homomorphic properties, we have $s_l = s_l^{\mathtt{l}}$. Similarly, when $s_{q,l} > s_l^{\mathtt{r}}$, we have $s_l = s_l^{\mathtt{r}}$. Otherwise, we have $s_l = s_{q,l}$. Therefore, the private filter protocol is correct.

Informally, our scheme contains four algorithms and can be defined as $\Pi = (\mathtt{Setup}, \mathtt{Outsourcing}, \mathtt{TokenGen}, \mathtt{QueryProcessing})$.

- $\mathtt{Setup}(\mathtt{k}_0, \mathtt{k}_1, \mathtt{k}_2, \mathcal{D})$: In the setup algorithm, on input security parameters $\{\mathtt{k}_0, \mathtt{k}_1, \mathtt{k}_2\}$ and the dataset $\mathcal{D}$, the data owner generates many system parameters, a public key pk, and a secret key sk. Then, it publishes the system parameters and pk, and then sends sk to the server S2.
- $\mathtt{Outsourcing}(\mathcal{D}, \mathtt{pk}, \mathtt{sk})$: In the data outsourcing algorithm, the data owner represents the dataset $\mathcal{D}$ to an MIUR-tree $\mathcal{T}$ and uses $\{\mathtt{pk}, \mathtt{sk}\}$ to encrypt $\mathcal{T}$ into an encrypted tree $\mathtt{Cipher}_{\mathcal{T}}$.
- $\mathtt{TokenGen}((\mathbf{q}, k), \mathtt{pk})$: In the token generation algorithm, the query user utilizes pk to encrypt each RkNN query $(\mathbf{q}, k)$ into a query token $(\mathtt{Token}_{\mathtt{Fil}}, k)$ and sends it to the server S1.
- $\mathtt{QueryProcessing}(\text{S1:}\mathtt{Cipher}_{\mathcal{T}}, (\mathtt{Token}_{\mathtt{Fil}}, k); \text{S2:}\mathtt{sk})$: In the query processing phase, the server S1 with $\mathtt{Cipher}_{\mathcal{T}}$ and $(\mathtt{Token}_{\mathtt{Fil}}, k)$ collaboratively processes the RkNN query with the server S2 with sk.

5 Our Proposed Scheme

In this section, we present our RkNN query scheme, which contains four algorithms, i.e., $\Pi = (\mathtt{Setup}, \mathtt{Outsourcing}, \mathtt{TokenGen}, \mathtt{QueryProcessing})$.

• $\mathsf{Setup}(\mathtt{k}_0, \mathtt{k}_1, \mathtt{k}_2)$: In the setup algorithm, on input the security parameters $\{\mathtt{k}_0, \mathtt{k}_1, \mathtt{k}_2\}$, the data owner first generates public parameters and the secret key as $\{\mathtt{pp}, \mathtt{sk}, \mathcal{M}\} \leftarrow \mathsf{KeyGen}(\mathtt{k}_0, \mathtt{k}_1, \mathtt{k}_2)$. Then, it generates the public key $\mathtt{pk} = \{\mathtt{pp}, \llbracket 0\rrbracket_1, \llbracket 0\rrbracket_2, \llbracket -1\rrbracket\}$, where $\llbracket -1\rrbracket$ is used for assisting the execution of the private filter and refinement protocols. After that, the data owner publishes $\{\mathtt{pk}, \mathcal{M}\}$ and sends sk to S2.

• $\mathsf{Outsourcing}(\mathcal{D}, \mathtt{pk}, \mathtt{sk})$: In the data outsourcing algorithm, the data owner outsources its dataset $\mathcal{D} = \{\mathbf{x}_i = (\mathbf{s}_i, \mathbf{t}_i)\}_{i=1}^n$ to the cloud server S1 as follows.

Step 1. The data owner first identifies the parameters α and K, where $0 \le \alpha \le 1$. Then, it computes $\psi_s = \max_{\mathbf{s}_i, \mathbf{s}_j \in \mathcal{D}} D(\mathbf{s}_i, \mathbf{s}_j)$, $\phi_s = \min_{\mathbf{s}_i, \mathbf{s}_j \in \mathcal{D}} D(\mathbf{s}_i, \mathbf{s}_j)$, $\psi_t = \max_{\mathbf{t}_i, \mathbf{t}_j \in \mathcal{D}} J(\mathbf{t}_i, \mathbf{t}_j)$, and $\phi_t = \min_{\mathbf{s}_i, \mathbf{s}_j \in \mathcal{D}} J(\mathbf{t}_i, \mathbf{t}_j)$.

Step 2. The data owner builds an MIUR-tree $\mathcal{T}$ for the dataset $\mathcal{D}$ and encrypts the built tree $\mathcal{T}$. During the process of tree encryption, each internal node E is encrypted into the ciphertext $\mathtt{Cipher}_E$ as Eq. (7), and each leaf node $(\mathbf{x}_i, \{\tau_i^{k\mathtt{NN}}\}_{k=1}^K)$ is encrypted into $\mathtt{Cipher}_{\mathbf{x}_i}$ as Eq. (4). Then, the data owner outsources the encrypted tree, denoted by $\mathtt{Cipher}_{\mathcal{T}}$, to the server S1.

• $\mathsf{TokenGen}((\mathbf{q}, k), \mathtt{pk})$: In the token generation algorithm, the query user uses the public key pk to encrypt a spatial-feature-based RkNN query $(\mathbf{q}, k)$ into a ciphertext, which is used for searching the encrypted MIUR-tree $\mathtt{Cipher}_{\mathcal{T}}$. The refinement token and filter token are respectively $\mathtt{Token}_{\mathtt{Ref}}$ and $\mathtt{Token}_{\mathtt{Fil}}$. Since all ciphertexts in $\mathtt{Token}_{\mathtt{Ref}}$ have been contained in $\mathtt{Token}_{\mathtt{Fil}}$, as shown in Eq. (5) and Eq. (8), we only use $\mathtt{Token}_{\mathtt{Fil}}$ as the query token. Then, the user sends the RkNN query request $(\mathtt{Token}_{\mathtt{Fil}}, k)$ to the server S1.

• $\mathsf{QueryProcessing}(\text{S1:}\mathtt{Cipher}_{\mathcal{T}}, (\mathtt{Token}_{\mathtt{Fil}}, k); \text{S2:}\mathtt{sk})$: In the query processing algorithm, S1 searches on the encrypted MIUR-tree $\mathtt{Cipher}_{\mathcal{T}}$ for the qualified query result of the query $(\mathtt{Token}_{\mathtt{Fil}}, k)$ with the assistance of S2. The query algorithm is similar to that over the plaintext MIUR-tree in Alg. 1. Differently, the determination of $MaxSim(E, \mathbf{q}) \ge \tau^{k\mathtt{min}}$ is implemented by the private filter protocol, and the determination of $Sim(\mathbf{x}_i, \mathbf{q}) \ge \tau_i^{k\mathtt{NN}}$ is implemented by the private refinement protocol. Finally, S1 can obtain the final query result $\mathcal{R} = \{\llbracket \mathbf{x}_i\rrbracket = (\llbracket \mathbf{s}_i\rrbracket, \llbracket \mathbf{t}_i\rrbracket) | Sim(\mathbf{x}_i, \mathbf{q}) \ge \tau_i^{k\mathtt{NN}}\}$. Then, S1 returns the query result to the query user as the following steps.

Step 1. For each $\llbracket \mathbf{x}_i\rrbracket = (\llbracket \mathbf{s}_i\rrbracket, \llbracket \mathbf{t}_i\rrbracket)$, the data owner chooses a two-dimensional random vector $\mathbf{r}_{i,s} \in \mathcal{M}^2$ and a d-dimensional random vector $\mathbf{r}_{i,t} \in \mathcal{M}^d$. Then, it computes $\llbracket \mathbf{s}_i + \mathbf{r}_{i,s}\rrbracket \leftarrow (\llbracket \mathbf{s}_i\rrbracket + \mathbf{r}_{i,s}) \mod \mathtt{N}$ and $\llbracket \mathbf{t}_i + \mathbf{r}_{i,t}\rrbracket \leftarrow (\llbracket \mathbf{t}_i\rrbracket + \mathbf{r}_{i,t}) \mod \mathtt{N}$. After that, S1 sends $\mathcal{R}_1 = \{(\mathbf{r}_{i,s}, \mathbf{r}_{i,t}) | Sim(\mathbf{x}_i, \mathbf{q}) \ge \tau_i^{k\mathtt{NN}}\}$ to the query user and sends $\llbracket \mathcal{R}_2\rrbracket = \{(\llbracket \mathbf{s}_i + \mathbf{r}_{i,s}\rrbracket, \llbracket \mathbf{t}_i + \mathbf{r}_{i,t}\rrbracket) | Sim(\mathbf{x}_i, \mathbf{q}) \ge \tau_i^{k\mathtt{NN}}\}$ to the cloud server S2.

Step 2. On receiving $\llbracket \mathcal{R}_2\rrbracket = \{(\llbracket \mathbf{s}_i + \mathbf{r}_{i,s}\rrbracket, \llbracket \mathbf{t}_i + \mathbf{r}_{i,t}\rrbracket) | Sim(\mathbf{x}_i, \mathbf{q}) \ge \tau_i^{k\mathtt{NN}}\}$, S2 utilizes the secret key sk to recover each pair of $(\mathbf{s}_i + \mathbf{r}_{i,s}, \mathbf{t}_i + \mathbf{r}_{i,t})$ and returns $\mathcal{R}_2 = \{(\mathbf{s}_i + \mathbf{r}_{i,s}, \mathbf{t}_i + \mathbf{r}_{i,t}) | Sim(\mathbf{x}_i, \mathbf{q}) \ge \tau_i^{k\mathtt{NN}}\}$ to the query user.

Step 3. On receiving the query results $\mathcal{R}_1$ and $\mathcal{R}_2$, the query user recovers the query result $\mathcal{R} = \{\mathbf{x}_i = (\mathbf{s}_i, \mathbf{t}_i) | Sim(\mathbf{x}_i, \mathbf{q}) \ge \tau_i^{k\mathtt{NN}}\}$. Specifically, for each $(\mathbf{s}_i + \mathbf{r}_{i,s}, \mathbf{s}_i + \mathbf{r}_{i,s}) \in \mathcal{R}_2$ and $(\mathbf{r}_{i,s}, \mathbf{r}_{i,t}) \in \mathcal{R}_1$, it recovers $\mathbf{x}_i = (\mathbf{s}_i, \mathbf{t}_i)$ as $\mathbf{s}_i = \mathbf{s}_i + \mathbf{r}_{i,s} - \mathbf{r}_{i,s}$ and $\mathbf{t}_i = \mathbf{t}_i + \mathbf{r}_{i,t} - \mathbf{r}_{i,t}$.

6 Security Analysis

As described in our security model, the security of scheme is measured by the leakages $\mathcal{L}_1$ and $\mathcal{L}_2$ to the servers S1 and S2, where $\mathcal{L}_1$ and $\mathcal{L}_2$ are defined as follows.

- $\mathcal{L}_1$ includes the public key $\mathtt{pk}$, the message space $\mathcal{M}$, the encrypted MIUR-tree $\mathtt{Cipher}_{\mathcal{T}}$, the access pattern of $\mathtt{Cipher}_{\mathcal{T}}$, and the ciphertext of each query request $\mathtt{Token}_{\mathtt{Fil}}$. In addition, when executing private refinement and filter protocols, there are a series of values $\{\llbracket y_i^L \rrbracket, \llbracket y_i^R \rrbracket, \llbracket f_l^{\mathtt{l}} \rrbracket, \llbracket f_l^{\mathtt{r}} \rrbracket\}$ in the leakages.
- $\mathcal{L}_2$ includes the secret key $\mathtt{sk}$ and the sign of a series of random numbers $\{y_i^L, y_i^R, z_l^{\mathtt{l}}, z_l^{\mathtt{r}}\}$ in the private refinement and filter protocols.

Based on the leakages $\mathcal{L}_1$, $\mathcal{L}_2$, we can construct the ideal model of our scheme and prove that our scheme is selectively secure with the leakages $\mathcal{L}_1$ and $\mathcal{L}_2$.

Ideal model. The ideal model is executed between two simulators $\{\mathtt{Sim}_1, \mathtt{Sim}_2\}$ and two adversaries $\{\mathtt{Adv}_1, \mathtt{Adv}_2\}$. $\mathtt{Sim}_1$ holds the leakage $\mathcal{L}_1$, and $\mathtt{Sim}_2$ holds the leakage $\mathcal{L}_2$. $\mathtt{Sim}_1$ and $\mathtt{Sim}_2$ simulate the views of $\mathtt{Adv}_1$ and $\mathtt{Adv}_2$ based on $\mathcal{L}_1$ and $\mathcal{L}_2$, respectively. Specifically, the ideal model has four algorithms as $\Pi^{\mathtt{Sim}} = (\mathsf{Setup}^{\mathtt{Sim}}, \mathsf{Outsourcing}^{\mathtt{Sim}}, \mathsf{TokenGen}^{\mathtt{Sim}}, \mathsf{QueryProcessing}^{\mathtt{Sim}})$.

- $\mathsf{Setup}^{\mathtt{Sim}}(\mathcal{L}_1, \mathcal{L}_2)$: In the setup algorithm, $\mathtt{Sim}_1$ publishes the public key $\mathtt{pk}$ and the message space $\mathcal{M}$ in $\mathcal{L}_1$. $\mathtt{Sim}_2$ sends $\mathtt{sk} \in \mathcal{L}_2$ to $\mathtt{Adv}_2$.
- $\mathsf{Outsourcing}^{\mathtt{Sim}}(\mathcal{L}_1, \mathcal{D})$: In the data outsourcing algorithm, $\mathtt{Sim}_1$ utilizes the leakage $\mathtt{Cipher}_{\mathcal{T}}$ to construct an encrypted MIUR-tree for the dataset $\mathcal{D}$. Specifically, for each SHE ciphertext $\llbracket \mathtt{m} \rrbracket \in \mathtt{Cipher}_{\mathcal{T}}$, $\mathtt{Sim}_1$ chooses a random number $\llbracket \mathtt{m} \rrbracket^{\mathtt{Sim}}$ in $\mathbb{Z}_{\mathtt{N}}$ and replaces $\llbracket \mathtt{m} \rrbracket$ with $\llbracket \mathtt{m} \rrbracket^{\mathtt{Sim}}$.

After replacing all SHE ciphertexts in $\mathtt{Cipher}_{\mathcal{T}}$, the MIUR-tree $\mathtt{Cipher}_{\mathcal{T}}$ will become a simulated tree, denoted by $\mathtt{Cipher}_{\mathcal{T}}^{\mathtt{Sim}}$. Then, $\mathtt{Sim}_1$ sends $\mathtt{Cipher}_{\mathcal{T}}^{\mathtt{Sim}}$ to $\mathtt{Adv}_1$.

- $\mathsf{TokenGen}^{\mathtt{Sim}}(\mathcal{L}_1, (\mathbf{q}, k))$: In the token generation algorithm, $\mathtt{Sim}_1$ constructs the query token for $(\mathbf{q}, k)$ based on the leakage $\mathtt{Token}_{\mathtt{Fil}}$. Specifically, for all SHE ciphertexts in $\mathtt{Token}_{\mathtt{Fil}}$, $\mathtt{Sim}_1$ replaces them with random values in $\mathbb{Z}_{\mathtt{N}}$, and the simulated token is denoted by $(\mathtt{Token}_{\mathtt{Fil}}^{\mathtt{Sim}}, k)$.
- $\mathsf{QueryProcessing}(\mathcal{L}_1, \mathcal{L}_2, (\mathbf{q}, k))$: In the query processing algorithm, $\mathtt{Sim}_1$ and $\mathtt{Sim}_2$ respectively simulate the views of $\mathtt{Adv}_1$ and $\mathtt{Adv}_2$ during the process of handling the query $(\mathbf{q}, k)$. The simulation includes a filter stage and a refinement stage.

Refinement stage. In the refinement stage, $\mathtt{Sim}_1$ and $\mathtt{Sim}_2$ respectively simulate the views of $\mathtt{Adv}_1$ and $\mathtt{Adv}_2$ in the private refinement protocol. Specifically, S1 can view $\{f_i^L, f_i^R\}$ that have been in the access pattern. Thus, $\mathtt{Sim}_1$ can directly simulate $\{f_i^L, f_i^R\}$ using the access pattern of $\mathtt{Cipher}_{\mathcal{T}}$. S2 can view $\{\llbracket y_i^L \rrbracket, \llbracket y_i^R \rrbracket\}$. $\mathtt{Sim}_2$ will do the simulation by generating two random numbers $\{y_i^{L,\mathtt{Sim}}, y_i^{R,\mathtt{Sim}}\} \in \mathcal{M}$ and encrypt them into ciphertexts $\{\llbracket y_i^{L,\mathtt{Sim}} \rrbracket, \llbracket y_i^{R,\mathtt{Sim}} \rrbracket\}$. Then, $\mathtt{Sim}_2$ sends $\{\llbracket y_i^{L,\mathtt{Sim}} \rrbracket, \llbracket y_i^{R,\mathtt{Sim}} \rrbracket\}$ to S2 as the simulation of $\{\llbracket y_i^L \rrbracket, \llbracket y_i^R \rrbracket\}$.

Filter stage. $\mathtt{Sim}_1$ and $\mathtt{Sim}_2$ respectively simulate the views of $\mathtt{Adv}_1$ and $\mathtt{Adv}_2$ in the private filter protocol.

Step 1. When determining $\llbracket \mathbf{s} \rrbracket = (\llbracket s_1 \rrbracket, \llbracket s_2 \rrbracket)$, S1 can view $\{\llbracket h_l^{\mathtt{l}} \rrbracket, \llbracket h_l^{\mathtt{r}} \rrbracket\}$, and S2 can view $\{\llbracket z_l^{\mathtt{l}} \rrbracket, \llbracket z_l^{\mathtt{r}} \rrbracket\}$. $\mathtt{Sim}_1$ simulates the views of $\mathtt{Adv}_1$ using two random numbers $\{\llbracket h_l^{\mathtt{l}} \rrbracket^{\mathtt{Sim}}, \llbracket h_l^{\mathtt{r}} \rrbracket^{\mathtt{Sim}}\} \in \mathbb{Z}_{\mathtt{N}}$. $\mathtt{Sim}_2$ simulates the views of $\mathtt{Adv}_2$ by choosing random numbers $\{z_l^{\mathtt{l},\mathtt{Sim}}, z_l^{\mathtt{r},\mathtt{Sim}}\} \in \mathcal{M}$ and encrypting them into ciphertexts $\{\llbracket z_l^{\mathtt{l},\mathtt{Sim}} \rrbracket, \llbracket z_l^{\mathtt{r},\mathtt{Sim}} \rrbracket\}$, where

$$\begin{cases} z_l^{\mathtt{l},\mathtt{Sim}} > 0 & z_l^{\mathtt{l}} > 0 \\ z_l^{\mathtt{l},\mathtt{Sim}} < 0 & z_l^{\mathtt{l}} < 0 \end{cases} \qquad \begin{cases} z_l^{\mathtt{r},\mathtt{Sim}} > 0 & z_l^{\mathtt{r}} > 0 \\ z_l^{\mathtt{r},\mathtt{Sim}} < 0 & z_l^{\mathtt{r}} < 0. \end{cases}$$

Step 2. When determining $\llbracket \mathbf{t} \rrbracket = (\llbracket t_1 \rrbracket, \llbracket t_2 \rrbracket, \cdots, \llbracket t_d \rrbracket)$, $\mathtt{Sim}_1$ and $\mathtt{Sim}_2$ simulate the views of $\mathtt{Adv}_1$ and $\mathtt{Adv}_2$ in the same way as the above ***Step 1***.

Step 3. When determining whether $MaxSim(E, \mathbf{q}) \geq \tau^{k\mathtt{min}}$, $\mathtt{Sim}_1$ and $\mathtt{Sim}_2$ simulate the views of $\mathtt{Adv}_1$ and $\mathtt{Adv}_2$ in the same way as the simulated refinement stage.

Return query results. When returning the query result to users, S1 views $\mathcal{R}_1 = \{(\mathbf{r}_{i,s}, \mathbf{r}_{i,t}) | Sim(\mathbf{x}_i, \mathbf{q}) \geq \tau_i^{k\mathtt{NN}}\}$, and S2 views $\llbracket \mathcal{R}_2 \rrbracket = \{(\llbracket \mathbf{s}_i + \mathbf{r}_{i,s} \rrbracket, \llbracket \mathbf{t}_i + \mathbf{r}_{i,t} \rrbracket) | Sim(\mathbf{x}_i, \mathbf{q}) \geq \tau_i^{k\mathtt{NN}}\}$. Thus, $\mathtt{Sim}_1$ simulates the view of $\mathtt{Adv}_1$ by constructing a set with $|\mathcal{R}_1|$ random vectors, i.e., $\mathcal{R}_1^{\mathtt{Sim}} = \{(\mathbf{r}_{i,s}^{\mathtt{Sim}}, \mathbf{r}_{i,t}^{\mathtt{Sim}})\}_{i=1}^{|\mathcal{R}_1|}$, where $\mathbf{r}_{i,s}^{\mathtt{Sim}}$ and $\mathbf{r}_{i,t}^{\mathtt{Sim}}$ have the same number of dimensions as $\mathbf{r}_{i,s}$ and $\mathbf{r}_{i,t}$. $\mathtt{Sim}_2$ simulates the view of $\mathtt{Adv}_2$ by constructing a set with $|\mathcal{R}_2|$ random vectors, i.e., $\mathcal{R}_2^{\mathtt{Sim}} = \{(\mathbf{w}_{i,s}^{\mathtt{Sim}}, \mathbf{w}_{i,t}^{\mathtt{Sim}})\}_{i=1}^{|\mathcal{R}_2|}$, where $\mathbf{w}_{i,s}^{\mathtt{Sim}}$ and $\mathbf{w}_{i,t}^{\mathtt{Sim}}$ have the same number of dimensions as $\mathbf{s}_i + \mathbf{r}_{i,s}$ and $\mathbf{t}_i + \mathbf{r}_{i,t}$.

In the following, we prove that our scheme is selective security with the leakages $\mathcal{L}_1$ and $\mathcal{L}_2$.

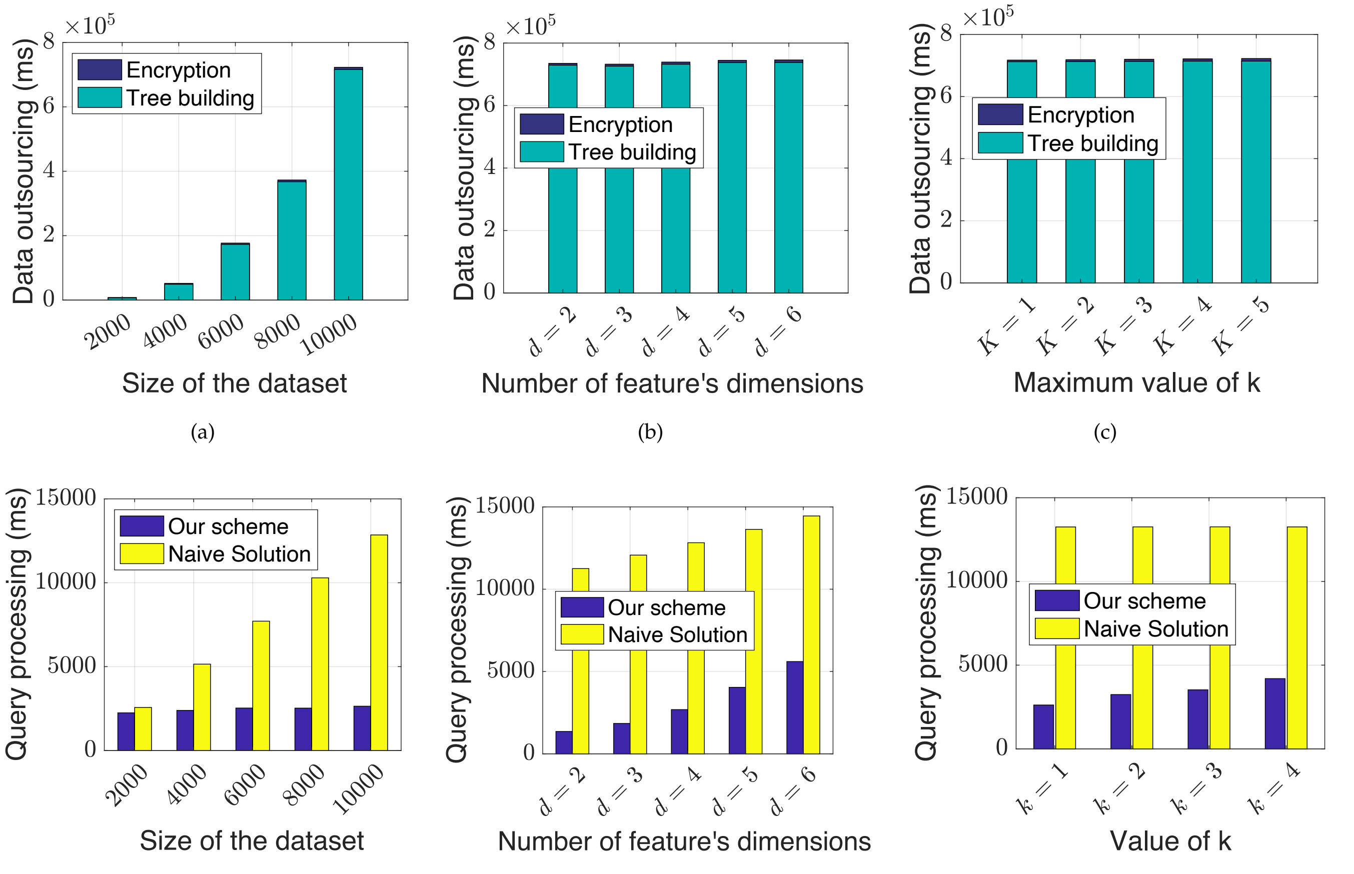
Theorem 6.1 *Our scheme is selectively secure with the leakages $\mathcal{L}_1$ and $\mathcal{L}_2$.*

Proof 6.1 *Based on Definition 2.1, our scheme is selectively secure if $\mathtt{Adv}_1$ and $\mathtt{Adv}_2$ only have a negligible probability to distinguish their views in real and ideal models. In the real model, all values that $\mathtt{Adv}_1$ can view are either SHE ciphertexts or in the access pattern. In the ideal model, the views of $\mathtt{Adv}_1$ are simulated using either random numbers or the access pattern. Thus, the semantic security of the SHE scheme can guarantee that $\mathtt{Adv}_1$ cannot distinguish the ciphertexts in the real and ideal models, i.e., $|\Pr(b_1 = 1 | \mathtt{Real}_{\mathtt{Adv}_1,\mathtt{Adv}_2}) - \Pr(b_1 = 1 | \mathtt{Ideal}_{\mathtt{Adv}_1,\mathtt{Adv}_2}^{\mathtt{Sim}_1,\mathtt{Sim}_2})|$ is negligible. For $\mathtt{Adv}_2$, it can view a series of values including $\{y_i^L, y_i^R, z_l^{\mathtt{l}}, z_l^{\mathtt{r}}, (\mathbf{s}_i + \mathbf{r}_{i,s}, \mathbf{t}_i + \mathbf{r}_{i,t})\}$ that contain random numbers. Meanwhile, all simulated numbers $\{y_i^{L,\mathtt{Sim}}, y_i^{R,\mathtt{Sim}}, z_l^{\mathtt{l},\mathtt{Sim}}, z_l^{\mathtt{r},\mathtt{Sim}}, (\mathbf{w}_{i,s}^{\mathtt{Sim}}, \mathbf{w}_{i,t}^{\mathtt{Sim}})\}$ are also random numbers. Thus, $\mathtt{Adv}_2$ cannot distinguish its views in real and ideal models, i.e., $|\Pr(b_2 = 1 | \mathtt{Real}_{\mathtt{Adv}_1,\mathtt{Adv}_2}) - \Pr(b_2 = 1 | \mathtt{Ideal}_{\mathtt{Adv}_1,\mathtt{Adv}_2}^{\mathtt{Sim}_1,\mathtt{Sim}_2})|$ is negligible. Therefore, our scheme is selectively secure with the leakages $\mathcal{L}_1$ and $\mathcal{L}_2$.*

7 Performance Evaluation

In this section, we evaluate the computational costs of our scheme with respect to the data outsourcing, token generation, and query processing. Since our scheme is the first spatial-feature-based RkNN query scheme, we compare our scheme with a naive solution with respect to query efficiency.

Experimental Setting. We use Java to implement our scheme and perform evaluation on a machine with Intel(R) Core(TM) i5-9400H CPU 2.50 GHz, 24GB RAM, and Windows 10 operating system. In our evaluation, we set $\alpha = 0.5$ and the security parameters of SHE scheme as $\mathtt{k}_0 = 4096$,



(a) (b) (c)

(d) (e) (f)

Fig. 4. Performance evaluation. (a) Data outsourcing with n, where $d = 4$ and $K = 3$; (b) Data outsourcing with d, where $n = 10000$ and $K = 3$; (c) Data outsourcing with K, where $n = 10000$ and $d = 4$. (d) Query processing with n, where $d = 4$ and $k = 1$; (e) Query processing with d, where $n = 10000$ and $k = 1$; (f) Query processing with K, where $n = 10000$ and $d = 4$.

TABLE 1
Time of token generation with d

d	2	3	4	5	6
Token generation (ms)	0.094	0.1001	0.1091	0.1181	0.1285

$\mathtt{k}_1 = 30$, and $\mathtt{k}_2 = 160$. We evaluate the performance on a real Airbnb dataset [33]. Specifically, we extract 10000 records from four cities, including Amsterdam, Antwerp, Asheville, and Athens. For each record, we extract 8 features, including latitude, longitude, etc. Each experiment is conducted 50 times, and the average results are reported.

- **Data Outsourcing.** In the data outsourcing phase, the data owner builds an MIUR-tree $\mathcal{T}$ for the dataset $\mathcal{D}$ and encrypts $\mathcal{T}$ into an encrypted tree $\mathtt{Cipher}_{\mathcal{T}}$. The computational costs are impacted by (i) n: the size of the dataset; (ii) d: the number of dimensions in the feature vector; and (iii) K: the maximum value of k. In Fig. 4(a), Fig. 4(b), and Fig. 4(c), we draw the impact of $\{n, d, K\}$ on the computational costs of data outsourcing that are separately depicted as tree building and encryption. As shown in those figures, the computational costs of tree encryption are greatly lower than that of tree building. The reason is that (i) tree building needs to compute the K nearest neighbors for each record and recursively retrieve the most similar pair of internal nodes to group; and (ii) our adopted SHE scheme is highly efficient. Besides, Fig. 4(a) demonstrates that the computational costs of data outsourcing significantly rise with the increase of n, while the impact of d and K on the data outsourcing is not significant, as shown in Fig. 4(b) and Fig. 4(c). This is rational because increasing n will result in the growth of recursive rounds and the computational costs of each round in tree building. While the increase of d and K only increases the computational costs of feature similarity calculation in each recursive round of tree building.

- **Token Generation.** In the token generation phase, the query user encrypts each query $\mathbf{q} = (\mathbf{s}_q, \mathbf{t}_q)$ into the token $\mathtt{Token}_{\mathtt{Fil}} \leftarrow \{\llbracket \mathbf{s}_q^2 \rrbracket, \llbracket -2\mathbf{s}_q \rrbracket, \llbracket \mathbf{s}_q \rrbracket, \llbracket \mathbf{t}_q \rrbracket, \llbracket \|\mathbf{t}_q\|^2 \rrbracket\}$. Since the number of dimensions in the spatial data is fixed to 2, the computational costs of token generation are impacted by the number of dimensions in the feature data, i.e., d. In Table 1, we plot the computational costs of token generation varying with d. This table indicates that the computational costs of token generation rise with d. Most importantly, the token generation phase is highly efficient, e.g., generating a token for a data with 6 features only takes 0.1285 ms.

- **Query Processing.** We evaluate the computational costs of query processing in our scheme and compare it with the naive solution. In the naive solution, each record is encrypted into a ciphertext $\texttt{Cipher}_{\mathbf{x}_i}$ based on the private refinement protocol, and each query is processed by linearly traversing all records in the dataset based on the refinement protocol. According to the description of our scheme, the computational costs of query processing are impacted by the parameters n, d, and k. As a result, we evaluate the performance of query processing with these parameters in Fig. 4(d), Fig. 4(e), and Fig. 4(f). As shown in these figures, the computational costs of our scheme and the naive solution rise as the growth of n, d, and k. This is because the increase of n will result in the growth of the MIUR-tree, the increase of d will lead to the growth of computational costs in searching internal and leaf nodes of the tree, and the increase of k will result in the increasing number of the records satisfying the the query request. Meanwhile, thanks to the filter strategy, our scheme is much more efficient than that of the naive solution.

8 Related Works

In this section, we review some existing privacy-preserving kNN query and RkNN query schemes that are closely related to our work.

- **Privacy-Preserving kNN Query.** Privacy-preserving kNN query has been widely studied in the literature, and various schemes have been proposed. Specifically, Wong et al. [15] utilized matrix encryption to design a secure Euclidean distance based kNN query scheme, named ASPE. The ASPE scheme is computationally efficient and drives a series of matrix encryption based kNN query scheme [16]–[21] to be proposed. However, these schemes only have a weak security, and some of them even cannot resist ciphertext-only attacks [34]. To address the security issue, in the work [35], we present a modified ASPE scheme by introducing more random numbers into ciphertexts. Another line of designing kNN query schemes is to employ homomorphic encryption to protect the data privacy, and various schemes were thus proposed [5], [22]–[24]. These schemes have a strong security, and even some of them [23], [24] are access pattern privacy-preserving. However, the above schemes cannot be directly employed to achieve efficient spatial-feature-based kNN queries. To bridge this gap, Su et al. [36] proposed a privacy-preserving spatial-feature-based kNN query scheme by designing a secure index based on the IR-tree and introducing two effective strategies, including anchor-based position determination and position-distinguished trapdoor generation. Recently, Tong et al. [37] proposed a ranked spatial-feature-based query scheme to retrieve records falling in a spatial range and having top-k feature similarity. Nevertheless, it is non-trivial to employ the schemes in [36], [37] to achieve spatial-feature-based RkNN queries.
- **Privacy-Preserving RkNN Query.** The research on privacy-preserving RkNN query is significantly less than that on privacy-preserving kNN query. Only a few schemes were proposed. Specifically, Du et al. [11] proposed a privacy-preserving spatial RkNN query scheme using anonymous techniques. Similarly, Lin et al. [12] presented a privacy-preserving RkNN query scheme using anonymous techniques in the context of road network. However, the anonymizing techniques in [11], [12] only ensure the privacy in a statistical manner and sacrifice the accuracy of query results. To obtain accurate query results, Pournajaf et al. [13] utilized the PIR technique to propose a privacy-preserving RkNN query scheme, but it ignores the dataset privacy. Recently, Tzouramanis et al. [14] proposed an RkNN query scheme for Euclidean distance based multi-dimensional data by indexing the dataset with a tree and protecting the data privacy with the ASPE scheme. Meanwhile, Li et al. [10] proposed a spatial RkNN query scheme, supporting dynamic updates of the dataset. The scheme indexes the dataset with Delaunay Triangulation and protects the data privacy with structure encryption, order-preserving encryption, etc. However, existing RkNN query schemes cannot be trivially employed to achieve efficient spatial-feature-based RkNN queries.

Different from existing schemes, our scheme is the first one to consider privacy-preserving spatial-feature-based RkNN queries.

9 Conclusion

In this paper, we have proposed an efficient and privacy-preserving spatial-feature-based RkNN query scheme in the outsourced scenario, which is the first spatial-feature-based RkNN query scheme over encrypted data. Specifically, we first designed an MIUR-tree to index the spatial and feature dataset and introduced the corresponding RkNN query algorithm over the MIUR-tree. Then, we presented a private filter protocol and a private refinement protocol to protect the privacy of the RkNN query algorithm. Based on these protocols, we proposed our scheme in detail. In addition, we proved the security of our scheme and validated its performance through extensive experiments. In our future work, we plan to design RkNN query schemes that can support dynamic data updates.

Acknowledgements

This research was supported in part by NSERC Discovery Grants (04009, RGPIN-2022-03244), NSFC (61972304, 61932015, U22B2030), ZJNSF (LQ22F020022), OPPO Research Fund, Henan Key Laboratory of Network Cryptography Technology, and the Fundamental Research Funds for the Central Universities, JLU.

References

[1] "Big data market: Increasing data generation to drive growth post the crisis," [Online]. Available: https://www.prnewswire.com/news-releases/big-data-market-increasing-data-generation-to-drive-growth-post-the-crisis-301315941.html, [Accessed: Jan 24, 2022].

[2] Y. Zheng, R. Lu, Y. Guan, J. Shao, and H. Zhu, "Towards practical and privacy-preserving multi-dimensional range query over cloud," *IEEE Trans. Dependable Secur. Comput.*, pp. 1–1, 2021, doi=10.1109/TDSC.2021.3101120.

[3] Y. Zheng, R. Lu, S. Zhang, Y. Guan, J. Shao, F. Wang, and H. Zhu, "Pmrq: Achieving efficient and privacy-preserving multi-dimensional range query in ehealthcare," *IEEE Internet Things J.*, pp. 1–1, 2022, doi=10.1109/JIOT.2022.3158321.

[4] S. Zhang, S. Ray, R. Lu, Y. Zheng, and J. Shao, "Preserving location privacy for outsourced most-frequent item query in mobile crowdsensing," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 9139–9150, 2021.

[5] Y. Zheng, R. Lu, and J. Shao, "Achieving efficient and privacy-preserving k-nn query for outsourced ehealthcare data," *J. Medical Systems*, vol. 43, no. 5, pp. 123:1–123:13, 2019.

[6] S. Wang, Z. Bao, J. S. Culpepper, T. Sellis, and G. Cong, "Reverse k nearest neighbor search over trajectories," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 4, pp. 757–771, 2018.

[7] G. Casanova, E. Englmeier, M. E. Houle, P. Kröger, M. Nett, E. Schubert, and A. Zimek, "Dimensional testing for reverse k-nearest neighbor search," *Proc. VLDB Endow.*, vol. 10, no. 7, pp. 769–780, 2017.

[8] Y. Guan, R. Lu, Y. Zheng, S. Zhang, J. Shao, and G. Wei, "Achieving privacy-preserving discrete frchet distance range queries," *IEEE Trans. Dependable Secur. Comput.*, pp. 1–1, 2022, doi=10.1109/TDSC.2022.3171980.

[9] S. Zhang, S. Ray, R. Lu, Y. Zheng, Y. Guan, and J. Shao, "Towards efficient and privacy-preserving interval skyline queries over time series data," *IEEE Trans. Dependable Secur. Comput.*, pp. 1–1, 2022, doi=10.1109/TDSC.2022.3153759.

[10] X. Li, T. Xiang, S. Guo, H. Li, and Y. Mu, "Privacy-preserving reverse nearest neighbor query over encrypted spatial data," *IEEE Trans. Serv. Comput.*, pp. 1–1, 2021, doi=10.1109/TSC.2021.3065356.

[11] Y. Du, "Privacy-aware RNN query processing on location-based services," in *8th International Conference on Mobile Data Management (MDM 2007), Mannheim, Germany, May 7-11, 2007*, 2007, pp. 253–257.

[12] X. Lin, L. Zhou, and P. C. J. Gu, "Privacy preserving reverse nearest-neighbor queries processing on road network," in *Web-Age Information Management - WAIM 2012 International Workshops: GDMM, IWSN, MDSP, USDM, and XMLDM Harbin, China, August 18-20, 2012 Proceedings*, ser. Lecture Notes in Computer Science, vol. 7419, 2012, pp. 19–28.

[13] L. Pournajaf, F. Tahmasebian, L. Xiong, V. S. Sunderam, and C. Shahabi, "Privacy preserving reverse k-nearest neighbor queries," in *19th IEEE International Conference on Mobile Data Management, MDM 2018, Aalborg, Denmark, June 25-28, 2018*, 2018, pp. 177–186.

[14] T. Tzouramanis and Y. Manolopoulos, "Secure reverse k-nearest neighbours search over encrypted multi-dimensional databases," in *Proceedings of the 22nd International Database Engineering & Applications Symposium, IDEAS 2018, Villa San Giovanni, Italy, June 18-20, 2018*, 2018, pp. 84–94.

[15] W. K. Wong, D. W. Cheung, B. Kao, and N. Mamoulis, "Secure knn computation on encrypted databases," in *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2009, Providence, Rhode Island, USA, June 29 - July 2, 2009*, 2009, pp. 139–152.

[16] W. Sun, B. Wang, N. Cao, M. Li, W. Lou, Y. T. Hou, and H. Li, "Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking," in *8th ACM Symposium on Information, Computer and Communications Security, ASIA CCS '13, Hangzhou, China - May 08 - 10, 2013*, 2013, pp. 71–82.

[17] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," *IEEE Trans. Parallel Distributed Syst.*, vol. 25, no. 1, pp. 222–233, 2014.

[18] B. Wang, S. Yu, W. Lou, and Y. T. Hou, "Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud," in *2014 IEEE Conference on Computer Communications, INFOCOM 2014, Toronto, Canada, April 27 - May 2, 2014*, 2014, pp. 2112–2120.

[19] H. Li, Y. Yang, T. H. Luan, X. Liang, L. Zhou, and X. S. Shen, "Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data," *IEEE Trans. Dependable Secur. Comput.*, vol. 13, no. 3, pp. 312–325, 2016.

[20] J. Yu, P. Lu, Y. Zhu, G. Xue, and M. Li, "Toward secure multi-keyword top-k retrieval over encrypted cloud data," *IEEE Trans. Dependable Secur. Comput.*, vol. 10, no. 4, pp. 239–250, 2013.

[21] Y. Zhu, R. Xu, and T. Takagi, "Secure k-nn computation on encrypted cloud data without sharing key with query users," in *Proceedings of the 2013 International Workshop on Security in Cloud Computing, SCC@ASIACCS '13, Hangzhou, China, May 8, 2013*, 2013, pp. 55–60.

[22] S. Rane and P. T. Boufounos, "Privacy-preserving nearest neighbor methods: Comparing signals without revealing them," *IEEE Signal Process. Mag.*, vol. 30, no. 2, pp. 18–28, 2013.

[23] Y. Elmehdwi, B. K. Samanthula, and W. Jiang, "Secure k-nearest neighbor query over encrypted data in outsourced environments," in *IEEE 30th International Conference on Data Engineering, Chicago, ICDE 2014, IL, USA, March 31 - April 4, 2014*, 2014, pp. 664–675.

[24] Y. Guan, R. Lu, Y. Zheng, J. Shao, and G. Wei, "Toward oblivious location-based k-nearest neighbor query in smart cities," *IEEE Internet Things J.*, pp. 1–1, 2021, doi=10.1109/JIOT.2021.3068859.

[25] V. Nikolaenko, S. Ioannidis, U. Weinsberg, M. Joye, N. Taft, and D. Boneh, "Privacy-preserving matrix factorization," in *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013*, 2013, pp. 801–812.

[26] Y. Zheng, H. an, and C. Wang, "Learning the truth privately and confidently: Encrypted confidence-aware truth discovery in mobile crowdsensing," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 10, pp. 2475–2489, 2018.

[27] Y. Zheng, R. Lu, J. Shao, F. Yin, and H. Zhu, "Achieving practical symmetric searchable encryption with search pattern privacy over cloud," *IEEE Trans. Serv. Comput.*, pp. 1–1, 2020, doi=10.1109/TSC.2020.2992303.

[28] J. Lu, Y. Lu, and G. Cong, "Reverse spatial and textual k nearest neighbor search," in *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2011, Athens, Greece, June 12-16, 2011*, 2011, pp. 349–360.

[29] H. Mahdikhani, R. Lu, Y. Zheng, J. Shao, and A. A. Ghorbani, "Achieving o(log^{3}n) communication-efficient privacy-preserving range query in fog-based iot," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5220–5232, 2020.

[30] Y. Zheng, R. Lu, Y. Guan, J. Shao, and H. Zhu, "Efficient and privacy-preserving similarity range query over encrypted time series data," *IEEE Trans. Dependable Secur. Comput.*, 2021, doi=10.1109/TDSC.2021.3061611.

[31] Y. Guan, R. Lu, Y. Zheng, S. Zhang, J. Shao, and G. Wei, "Toward privacy-preserving cybertwin-based spatio-temporal keyword query for its in 6g era," *IEEE Internet Things J.*, 2021, doi=10.1109/JIOT.2021.3096674.

[32] N. Roussopoulos, S. Kelley, and F. Vincent, "Nearest neighbor queries," in *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, San Jose, California, USA, May 22-25, 1995*, 1995, pp. 71–79.

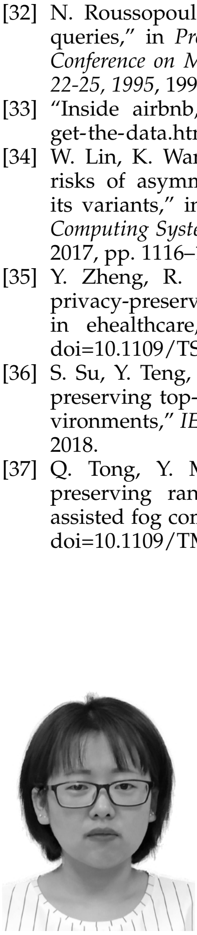
[33] "Inside airbnb," [Online], Available: http://insideairbnb.com/get-the-data.html, [Accessed: Jan 18, 2021].

[34] W. Lin, K. Wang, Z. Zhang, and H. Chen, "Revisiting security risks of asymmetric scalar product preserving encryption and its variants," in *37th IEEE International Conference on Distributed Computing Systems, ICDCS 2017, Atlanta, GA, USA, June 5-8, 2017*, 2017, pp. 1116–1125.

[35] Y. Zheng, R. Lu, Y. Guan, J. Shao, and H. Zhu, "Efficient privacy-preserving similarity range query with quadsector tree in ehealthcare," *IEEE Trans. Serv. Comput.*, pp. 1–1, 2021, doi=10.1109/TSC.2021.3081350.

[36] S. Su, Y. Teng, X. Cheng, K. Xiao, G. Li, and J. Chen, "Privacy-preserving top-k spatial keyword queries in untrusted cloud environments," *IEEE Trans. Serv. Comput.*, vol. 11, no. 5, pp. 796–809, 2018.

[37] Q. Tong, Y. Miao, H. Li, X. Liu, and R. Deng, "Privacy-preserving ranked spatial keyword query in mobile cloud-assisted fog computing," *IEEE Trans. Mob. Comput.*, pp. 1–1, 2021, doi=10.1109/TMC.2021.3134711.



Yandong Zheng (Member, IEEE) received the M.S. degree from the Department of Computer Science, Beihang University, China, in 2017, and received the Ph.D. degree from the Department of Computer Science, University of New Brunswick, Canada, in 2022.

Since 2022, she has been an Associate Professor with the School of Cyber Engineering, Xidian University. Her research interest includes cloud computing security, big data privacy, and applied privacy.

Rongxing Lu (Fellow, IEEE) received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, Canada, in 2012. He worked as a Post-Doctoral Fellow with the University of Waterloo from May 2012 to April 2013. He is currently a Mastercard IoT Research Chair, a University Research Scholar, and an Associate Professor with the Faculty of Computer Science (FCS), University of New Brunswick (UNB), Canada. Before that, he worked as an Assistant Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore, from April 2013 to August 2016. His research interests include applied cryptography, privacy enhancing technologies, and IoT-big data security, and privacy. He also serves as the Chair of IEEE Communications and Information Security Technical Committee (ComSoc CISTC), and the Founding Co-Chair of IEEE TEMS Blockchain and Distributed Ledgers Technologies Technical Committee (BDLT-TC).

Songnian Zhang received his M.S. degree from Xidian University, China, in 2016 and he is currently pursuing his Ph.D. degree in the Faculty of Computer Science, University of New Brunswick, Canada. His research interest includes cloud computing security, big data query and query privacy.

Jun Shao (Senior Member, IEEE) received the Ph.D. degree from the Department of Computer and Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2008.

He was a Post-Doctoral Fellow with the School of Information Sciences and Technology, Pennsylvania State University, Pennsylvania, PA, USA, from 2008 to 2010. He is currently a Professor with the School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou, China. His current research interests include network security and applied cryptography.

Fengwei Wang (Member, IEEE) received his B.Sc. degree from Xidian University in 2016 and Ph.D. degree from Xidian University in 2021. In 2019, he was with the Faculty of Computer Science, University of New Brunswick as a visiting scholar.

Since 2021, he has been the lecturer with the School of Cyber Engineering, Xidian University, Xi’an, China. His research interests include the areas of applied cryptography, cyber security, and privacy.

Hui Zhu (Senior Member, IEEE) received the B.Sc. degree from Xidian University, Xian, China, in 2003, the M.Sc. degree from Wuhan University, Wuhan, China, in 2005, and the Ph.D. degree from Xidian University, in 2009.

He was a Research Fellow with the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore, in 2013. Since 2016, he has been a Professor with the School of Cyber Engineering, Xidian University. His current research interests include applied cryptography, data security, and privacy.