Journal of Systems Architecture 120 (2021) 102300

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

SecVKQ: Secure and verifiable kNN queries in sensor–cloud systems

Qin Liu [a,*], Zhengzheng Hao [a], Yu Peng [a], Hongbo Jiang [a], Jie Wu [b], Tao Peng [c], Guojun Wang [c], Shaobo Zhang [d]

[a] *College of Computer Science and Electronic Engineering, Hunan University, Changsha, Hunan Province, 410082, PR China*
[b] *Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, USA*
[c] *School of Computer Science, Guangzhou University, Guangzhou, Guangdong Province, 510006, PR China*
[d] *School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, Hunan Province, 411201, PR China*

ARTICLE INFO

Keywords:
Security
Verification
kNN queries
Sensor–cloud systems
Edge computing

ABSTRACT

Sensor–cloud has gained increasingly popularity since it bridges the physical world and the cyber world through pervasive computation. This paper focuses on secure and verifiable k nearest neighbor (kNN) queries over large-scale outsourced datasets in sensor–cloud systems. Existing work in this line often incurred high computational and communication overheads, remaining far away from practical and scalable. To this end, we propose *SecVKQ*, a two-phase search framework, which mainly includes a preliminary screening phase and an exact search phase. SecVKQ purposely takes advantage of secure data separation and adaptive encryption strategy, embracing edge servers into the classic dual-cloud model, so as to optimize query performance. Under SecVKQ, we design a series of secure protocols and develop a succinct verification strategy to derive a unified solution. The experimental results demonstrate the effectiveness of SecVKQ. Compared to the state-of-the-art work, SecVKQ achieves a speed-up of two orders of magnitude in search latency, and a savings of 50% communication cost for verification.

1. Introduction

In recent years, sensor–cloud as the product of combining wireless sensor networks and cloud computing has received extensive attention from both academia and industry [1–3]. In addition to servers and people, sensors are important components of sensor–cloud systems. People can get relevant information (e.g., location, temperature and humidity) through different sensors [4]. With the rapid advances in mobile communications, location-based services (LBSs) are emerging as the next *killer application* in sensor–cloud systems. An LBS allows a mobile user to query an LBS provider, through location-aware sensors anytime and anywhere, in order to retrieve detailed information (e.g., position, ranking and photos) about points of interest (POIs) in her vicinity [5,6].

Among various LBS applications in sensor–cloud systems, k nearest neighbor (kNN) queries that allow a mobile user to obtain the top-k POIs with the shortest distance between her current position reach the top of popularity. For example, user Alice sitting at Starbucks is seeking for the closest restaurant. Mobile users will query LBS providers in a ubiquitous manner. To improve user experiences, LBS providers may deploy geographically distributed servers (referred to as edge servers) across a large area, so that mobile users can get fast responses no matter where they are [7–10].

However, the extensive popularization of LBS applications renders the explosive growth of location-related data. The data volume is expanding so fast that LBS providers could hardly keep up with the requirement. Hence, it is a promising choice for LBS providers to outsource their data to cloud service providers (CSP) for enhanced services and cost savings [11,12]. For example, the popular LBS providers, Foursquare and Yelp, outsourced their entire datasets to Amazon Web Services. On the other hand, outsourcing data to cloud service providers raises both data security and confidentiality issues [13–17].

Let us consider the following application scenario as shown in Fig. 1. The LBS provider outsources dataset $\mathcal{D}$ to the CSP, which is delegated to provide various location-based services to mobile users. As a typical example of kNN queries, Alice submits her location Q together with parameter $k = 3$ to the cloud server, which then finds out and returns the detailed information about the top-3 nearest POIs by comparing the distances between location Q and nearby POIs' positions. The following requirements are essential to this scenario: *(1) Security.* The CSP is not fully trusted and may leak sensitive data inadvertently or intentionally. For instance, the recent news about Amazon S3 divulging personal medical information. The dataset $\mathcal{D}$ or the mobile user's locations/trajectories are of high commercial value and related to

* Corresponding author.
E-mail address: gracelq628@hnu.edu.cn (Q. Liu).

https://doi.org/10.1016/j.sysarc.2021.102300
Received 2 April 2021; Received in revised form 30 August 2021; Accepted 25 September 2021
Available online 4 October 2021

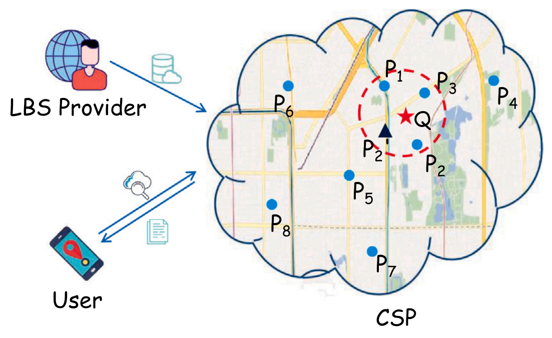


Fig. 1. Application scenario, where $\mathcal{D} = \{\mathcal{P}_1, \ldots, \mathcal{P}_8\}$. $\mathcal{P}_2'$ is an unregistered restaurant outside $\mathcal{D}$ and $\mathcal{P}_8$ is a remote restaurant.

privacy. Therefore, the contents of both dataset $\mathcal{D}$ and query Q need to be preserved against the CSP. *(2) Verifiability.* The CSP may deliberately falsify search results for business benefit. For example, the CSP forges the search result as $\{P_1, P_2', P_8\}$, when the truly top-3 nearest POIs are $\{P_1, P_2, P_3\}$. Therefore, the mobile user should have the ability to verify the search results in the aspects of *authenticity* (i.e., the POIs returned indeed are from the dataset $\mathcal{D}$) and *completeness* (i.e., no POIs satisfying the query condition are omitted).

Most of the existing work addressed the problem of security and proposed a number of secure kNN search schemes based on cryptographic methods with support for comparisons over ciphertexts. Generally speaking, the LBS provider encrypts the dataset before outsourcing so that the mobile user issues encrypted queries to retrieve encrypted POIs. According to the adopted encryption solutions, the secure kNN search schemes can be classified into efficiency-first (EF) schemes and security-first (SF) schemes. The EF schemes [18–22] employ light-weighted cryptography, such as order-preserving encryption (OPE) [23] or distance-recoverable encryption (DRE) [18], to support efficient kNN queries over encrypted data. The main drawback of EF schemes is that OPE and DRE are vulnerable to inference attacks and thus provide only a weak security guarantee [24,25]. The SF schemes [26–28] apply *homomorphic encryption, Paillier* into the *dual-cloud model* [26] for enhanced security. However, the SF schemes are impractical to large datasets because of the high computational overhead of homomorphic encryption.

Until now, little work [28–30] has been done in achieving security and verifiability simultaneously. As the state-of-the-art work in this field, Cui et al. [28] proposed the SVkNN scheme, which unified SF schemes with result integrity. The SVkNN scheme builds grid-based index to improve query performance, but its search complexity is linear to the size of datasets, thus lacking scalability. For example, SVkNN needs around 1 hour on average while performing 10NN queries on a dataset of 100,000 POIs. The LBS dataset usually contains hundreds of thousands or even millions of POIs. *Therefore, how to efficiently achieve secure and verifiable kNN queries on large-scale dataset is still an open problem.*

To solve this problem, we propose a comprehensive search framework, *SecVKQ*, where the search process is divided into the preliminary screening phase performed on edge servers and the exact search phase performed on two collude-resistant clouds. The dual-cloud model has been widely applied in secure kNN search schemes, since it was proposed in [26]. Existing work [27,28] assumed that two clouds will not collude in the search phase. We believe that this assumption is reasonable since two clouds of different interests are usually commercial competitors. For example, two clouds may be Google and Amazon, which are highly improbable to conspire with each other.

The most notable feature of SecVKQ is the integration of edge servers into the classic dual-cloud model, enabling the search complexity on the cloud side to be relevant to the size of preliminary results. Instead of a simple composition, SecVKQ takes advantage of clouds and edges to optimize query efficiency by utilizing secure data separation and adaptive encryption strategy. On the high level, the edge servers are responsible for evaluating a query coverage over an R-tree index built from the dataset to obtain preliminary results, in contrast, the cloud servers process exact positions of POIs and users to perform exact kNN queries. In terms of privacy levels, the R-tree index (resp. the query coverage) contains less sensitive information compared to the POIs' positions (resp. mobile users' exact positions). Hence, the light-weighted comparable inner product encoding (CIPE) scheme and the expensive homomorphic encryption are adopted in the preliminary screening phase and the exact search phase, respectively.

Under *SecVKQ*, we design a series of secure protocols to facilitate secure kNN searches over large-scale datasets and develop a compact verification strategy based on Voronoi diagram [31] and condensed RSA signature [32]. The main contributions of this paper are summarized as follows:

- We design a comprehensive search framework, SecVKQ, to achieve secure and verifiable kNN queries on sensor datasets. SecVKQ utilizes secure data separation and adaptive encryption strategy to take full advantage of clouds and edges in optimizing query efficiency.
- We utilize the Voronoi diagram for verification and propose a series of secure protocols and optimization techniques to improve the query performance further.
- We provide rigorous security analysis and evaluate the performance of SecVKQ on two real-world datasets to show its effectiveness. Experimental results demonstrate that SecVKQ achieves a speedup of two orders of magnitude in search latency compared to the state-of-the-art scheme. At best, it needs only around 10 seconds on average while performing secure and verifiable 10NN queries on a dataset of 100,000 POIs.

2. Preliminary

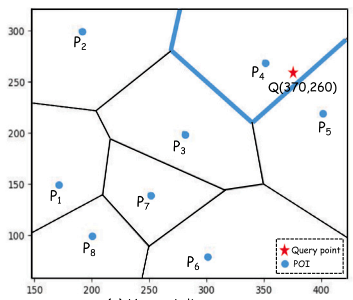
2.1. Voronoi diagram

Suppose that the dataset consists of n data points, $\mathcal{D} = \{\mathcal{P}_1, \ldots, \mathcal{P}_n\}$, where each data point denotes a POI. The Voronoi diagram partitions the space into n disjoint regions, where each region contains exactly one data point. The region where data point $\mathcal{P}_i$ is located is called $\mathcal{P}_i$'s Voronoi cell, denoted by $\mathcal{VC}(\mathcal{P}_i)$, and two data points $\mathcal{P}_i$ and $\mathcal{P}_j$ are Voronoi neighbors if $\mathcal{VC}(\mathcal{P}_i)$ and $\mathcal{VC}(\mathcal{P}_j)$ share a common edge. The set of Voronoi neighbors of data point $\mathcal{P}_i$ is denoted by $\mathcal{VN}(\mathcal{P}_i)$. Let $\mathcal{VD} = (\mathcal{P}_i, \mathcal{VN}(\mathcal{P}_i))_{i\in[n]}$ denote the Voronoi diagram built from the dataset $\mathcal{D}$. $\mathcal{VD}$ has two properties that are useful for the verification of kNN queries [31]:

Property 1. *Given a query point Q, the nearest neighbor of Q is data point $\mathcal{P}$, if $Q \in \mathcal{VC}(\mathcal{P})$.*

Property 2. *If data points $\mathcal{P}_1, \ldots, \mathcal{P}_k$ are the $k(k > 1)$ nearest neighbors of the query point Q, then $\mathcal{P}_i$ belongs to $\mathcal{VN}(\mathcal{P}_1) \cup \cdots \cup \mathcal{VN}(\mathcal{P}_{i-1})$, for $i = 2, \ldots, k$.*

For example, given a dataset $\mathcal{D}$ that contains 8 data points as shown in Fig. 2-(b), the Voronoi diagram is shown in Fig. 2-(a). The Voronoi neighbors of data point $\mathcal{P}_4$ include $\mathcal{VN}(\mathcal{P}_4) = \{\mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_5\}$ since $\mathcal{VC}(\mathcal{P}_4)$ shares a common edge with $\mathcal{VC}(\mathcal{P}_i)$ for $i \in \{2, 3, 5\}$. In this example, the search result of a 3NN query is $\mathcal{SR} = \{\mathcal{P}_4, \mathcal{P}_5, \mathcal{P}_3\}$.



(a) Voronoi diagram

POI	Coordinate	Voronoi Neighbors
P_1	(170,150)	$\{P_2, P_3, P_7, P_8\}$
P_2	(190,300)	$\{P_1, P_3, P_4\}$
P_3	(280,200)	$\{P_1, P_2, P_4, P_5, P_6, P_7\}$
P_4	(350,270)	$\{P_2, P_3, P_5\}$
P_5	(400,220)	$\{P_3, P_4, P_6\}$
P_6	(300,80)	$\{P_3, P_5, P_7, P_8\}$
P_7	(250,140)	$\{P_1, P_3, P_6, P_8\}$
P_8	(200,100)	$\{P_1, P_6, P_7\}$

(b) Dataset D

Fig. 2. An example of Voronoi diagram.

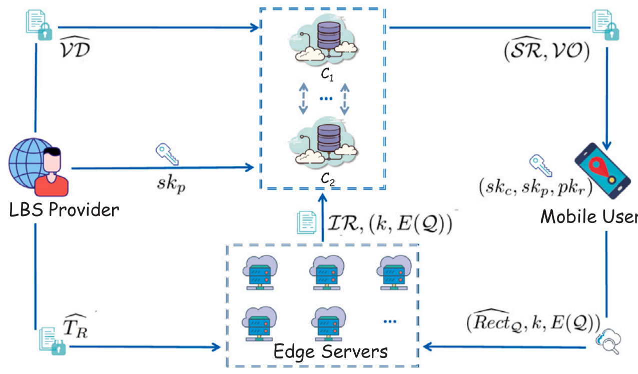

$\widehat{\mathcal{VD}}$
C_1
C_2
$(\widehat{\mathcal{SR}}, \mathcal{VO})$
sk_p
LBS Provider
(sk_c, sk_p, pk_r)
Mobile User
$\mathcal{IR}, (k, E(\mathcal{Q}))$
$\widehat{\mathcal{T}_R}$
Edge Servers
$(\widehat{Rect_{\mathcal{Q}}}, k, E(\mathcal{Q}))$

Fig. 3. System model. Secret keys are transmitted through secure channels.

2.2. Paillier cryptosystem

The Paillier cryptosystem is a probabilistic asymmetric encryption scheme with the additive homomorphic property [33]. Let E denote the encryption algorithm under public key pk_p, let D denote the decryption algorithm under secret key sk_p, and let N denote the product of two large primes. Given two plaintexts $x_1, x_2 \in \mathbb{Z}_N$, Paillier cryptosystem has following additive homomorphic property:

$$D(E(x_1) \times E(x_2) \bmod N^2) = x_1 + x_2$$
$$D(E(x_1)^{x_2} \bmod N^2) = x_1 \times x_2$$

Paillier cryptosystem achieves semantic security [34] and will be used to encrypt the exact positions of data/query points.

2.3. Secure protocols in dual-cloud model

Secure Squared Euclidean Distance (SSED) [26]. Given two cloud servers, $\mathbf{C}_1$ and $\mathbf{C}_2$, SSED allows to securely compute the encrypted squared Euclidean distance of two encrypted points. Specifically, $\mathbf{C}_1$ takes the encrypted points $(E(\mathcal{P}_1), E(\mathcal{P}_2))$ as input and $\mathbf{C}_2$ takes the secret key sk_p as input. At the end of this protocol, $\mathbf{C}_1$ obtains $E(\triangle(\mathcal{P}_1, \mathcal{P}_2))$, where $\triangle(\mathcal{P}_1, \mathcal{P}_2)$ denotes the squared Euclidean distance between $\mathcal{P}_1$ and $\mathcal{P}_2$.

Secure Comparison (SC) [35]. Given two cloud servers, $\mathbf{C}_1$ and $\mathbf{C}_2$, SC allows to securely compare two encrypted integers. Specifically, $\mathbf{C}_1$ takes the encrypted integers $(E(x_1), E(x_2))$ as input, and $\mathbf{C}_2$ takes the secret key sk_p as input. At the end of this protocol, $\mathbf{C}_1$ obtains $E(x_{\min})$, where $x_{\min} = \min(x_1, x_2)$.

2.4. Comparable inner product encoding

The CIPE scheme [36] consists of the following algorithms:

• GenK$(1^\kappa) \rightarrow sk_c$: This algorithm takes a security parameter κ as input and outputs a secret key sk_c.

• EncD$(x, sk_c) \rightarrow \hat{x}$: This algorithm takes a plaintext x and the secret key sk_c as input and outputs the ciphertext $\hat{x}$.

• EncQ$(QC, sk_c) \rightarrow \widehat{QC}$: This algorithm takes a query coverage $QC = [b_l, b_u]$ and the secret key sk_c as input and outputs the encrypted range $\widehat{QC}$.

• Compare$(\hat{x}, \widehat{QC}) \rightarrow \{0, 1\}$: Given the encrypted query range $\widehat{QC}$ and the encrypted value $\hat{x}$, this algorithm outputs -1 if $x < b_l$ and outputs 1 if $x > b_u$. Otherwise, it outputs 0.

The CIPE scheme allows edges to decide whether a value x falls in a query range QC or not based on their ciphertexts. The reason for the adoption of the CIPE scheme in the preliminary screening phase is that it not only can compete with OPE schemes in the aspect of search efficiency, but also can resist some attacks that OPE schemes are vulnerable to. For instance, the CIPE scheme can resist inference attacks by expanding the query matrix with noises [36].

2.5. Condensed RSA signature

Due to the property of multiplicative homomorphism, RSA signature scheme can be used to combine many signatures into one condensed signature [30]. Suppose that N' is a product of two large random primes and that $e \times d = 1 \bmod \phi(N')$, where $e, d \in \mathbb{Z}^*_{N'}$. For the standard RSA signature scheme [32], there are private key $sk_r = d$ and public key $pk_r = (N', e)$. We can compute the RSA signature $\sigma = h(m)^d \bmod N'$, where m is the message and $h()$ is a hash function. Then we can verify the signature σ by checking if $\sigma^e = h(m) \bmod N'$.

The condensed signature is computed by modular multiplying the original signatures. Suppose that $m_1, m_2, \ldots, m_k$ are k different messages, whose original RSA signatures are $\sigma_1, \sigma_2, \ldots, \sigma_k$, respectively. We can compute the condensed signature: $\sigma_{1,k} = (\prod_{i=1}^{k} \sigma_i) \bmod N'$. Then we can verify the condensed signature $\sigma_{1,k}$ by checking if: $\sigma_{1,k}^e = (\prod_{i=1}^{k} h(m_i)) \bmod N'$. The length of the condensed signature is equal to that of the original signature. Thus, the condensed RSA signature scheme can be used to reduce the communication cost.

3. Overview

3.1. System model

As illustrated in Fig. 3, our system model consists of the following entities.

• *LBS Provider.* As the data owner, the LBS provider wishes to outsource dataset D to the cloud to save money. To enable secure and verifiable kNN queries on large-scale datasets, the LBS provider performs secure data separation and adaptive encryption strategy in the data preprocessing stage. For highly private data, it builds a Voronoi diagram $\mathcal{VD}$ from D, encrypts $\mathcal{VD}$ with the Paillier cryptosystem, and signs $\mathcal{VD}$ with the condensed RSA signature. The pretreated Voronoi diagram $\widehat{\mathcal{VD}}$ and the secret key sk_p are sent to cloud servers $\mathbf{C}_1$ and $\mathbf{C}_2$, respectively. For less sensitive data, the LBS provider builds an R-tree index $\mathcal{T}_R$ from D, and encrypts $\mathcal{T}_R$ with the CIPE scheme. The encrypted index $\widehat{\mathcal{T}_R}$ is uploaded to appropriate edge servers to speed up the query process.

• *Mobile Users.* After obtaining authorization from the LBS provider, the mobile user can issue kNN queries to retrieve POIs in her vicinity. The mobile user located at position $\mathcal{Q}$ issues a search token $\mathcal{TK} = (\widehat{\text{Rect}_{\mathcal{Q}}}, k, E(\mathcal{Q}))$ to the nearby edge server, where $\widehat{\text{Rect}_{\mathcal{Q}}}$ is a two-dimensional range query encrypted under the CIPE scheme, and $E(\mathcal{Q})$ is her exact position encrypted under the Paillier cryptosystem. Once she receives an encrypted search result $\widehat{\mathcal{SR}}$ and a verification object $\mathcal{VO}$ from the cloud server, she recovers the plaintext result $\mathcal{SR}$ and validates the correctness of $\mathcal{SR}$ with $\mathcal{VO}$.

• *Edge Servers.* The edge servers possessed by the LBS provider are deployed over geographically distributed areas. Each edge server

Table 1
Summary of Notations.

$\mathcal{D}$	A spatial dataset that includes n points $\{\mathcal{P}_1, \dots, \mathcal{P}_n\}$
$\mathcal{P}_i, E(\mathcal{P}_i)$	A POI point with (x_i, y_i) and its encrypted coordinates
$\mathcal{Q}, E(\mathcal{Q})$	A query point with (x_q, y_q) and its encrypted coordinates
$\triangle(\mathcal{P}_i, \mathcal{Q})$	The squared Euclidean distance between $\mathcal{P}_i$ and $\mathcal{Q}$
sk_c	The secret key for CIPE scheme
sk_p, pk_p	The secret/public key for Paillier cryptosystem
sk_r, pk_r	The secret/public key for RSA signature
$\widehat{\mathcal{T}}_R, \mathcal{T}_R$	The encrypted/clear R-tree index built from $\mathcal{D}$
$\widehat{\mathcal{VD}}, \mathcal{VD}$	The pretreated and clear Voronoi diagram built from $\mathcal{D}$
$\widehat{\text{Rect}}_Q, \text{Rect}_Q$	The encrypted/clear range query generated for $\mathcal{Q}$
$\mathcal{TK}$	The search token consisted of $(\widehat{\text{Rect}}_Q, k, E(\mathcal{Q}))$
$\mathcal{IR}$	The immediate result in the preliminary screening phase
$\widehat{\mathcal{SR}}, \mathcal{SR}$	The encrypted/clear result in the exact search phase
$\mathcal{VO}$	The verification object consisted of $(\mathcal{VO}_{geo}, \mathcal{VO}_{sig})$
σ_i	The RSA signature for the information about $(\mathcal{P}_i, \mathcal{VN}(\mathcal{P}_i))$
$\sigma_{1,k}$	The condensed RSA signature for k signatures $\{\sigma_1, \dots, \sigma_k\}$

is responsible for preliminarily processing the kNN queries of nearby mobile users. Given the encrypted search token $\mathcal{TK} = (\widehat{\text{Rect}}_Q, k, E(\mathcal{Q}))$, it evaluates $\widehat{\text{Rect}}_Q$ over the encrypted R-tree index $\widehat{\mathcal{T}}_R$ to filter out the immediate result $\mathcal{IR}$, and then sends $\mathcal{IR}$ and a part of search token $(k, E(\mathcal{Q}))$ to the cloud server $\mathbf{C}_1$.

• *Dual-Cloud Servers.* Two cloud servers interact with each other without colluding in the exact search phase. At the high level, $\mathbf{C}_1$ with the pretreated Voronoi diagram $\widehat{\mathcal{VD}}$, the immediate result $\mathcal{IR}$, and the partial search token $(k, E(\mathcal{Q}))$, and $\mathbf{C}_2$ with the secret key sk_p, collaboratively execute a series of secure protocols to obtain an encrypted search result $\widehat{\mathcal{SR}}$. Finally, the verification object $\mathcal{VO}$ together with $\widehat{\mathcal{SR}}$ are returned to the mobile user for verification.

3.2. Adversary model

In our threat model, the LBS provider and mobile users are fully trusted. The cloud servers are called *active attackers*, that may not only try to learn additional information outside their permissions, but also return incorrect results for financial gain. In the dual-cloud model, two cloud servers belonging to different CSPs are assumed not to collude based on existing work [26–28]. Edge servers possessed by the LBS provider are called *passive attackers*, which may accidentally leak data due to external attacks. To ensure secure and verifiable kNN queries, our security goals try to preserve the following:

• *Data Privacy.* The content of dataset $\mathcal{D}$ should be protected against attackers.

• *Index Privacy.* The content of the search index should be protected against attackers.

• *Query Privacy.* The contents of queries and results of mobile users should be protected against attackers.

• *Result Correctness.* The mobile user is able to verify the correctness of the search result from the aspects of authenticity and completeness in an efficient way.

3.3. Notations and definitions

We write notations $[x]$ and $[x \sim y]$ to represent the set of integers in $\{1, \dots, x\}$ and $\{x, \dots, y\}$, respectively. The cardinality of set S is denoted by $|S|$. The concatenation of two strings s_1 and s_2 is denoted by $s_1|s_2$. The most relevant notations are shown in Table 1.

Let $\mathcal{D} = \{\mathcal{P}_1, \dots, \mathcal{P}_n\}$ be a spatial dataset, where each data point $\mathcal{P}_i$ is represented as a spatial coordinate (x_i, y_i). The exact position of each query point $\mathcal{Q}$ is also represented as a spatial coordinate (x_q, y_q). Let $\triangle(\mathcal{P}_i, \mathcal{Q})$ denote the squared Euclidean distance between $\mathcal{P}_i$ and $\mathcal{Q}$, and let $\widehat{\mathcal{SR}}$ and $\mathcal{SR}$ denote the encrypted and plaintext search results, respectively.

Definition 1 (*Secure and Verifiable kNN Query*)**.** Given encrypted indexes $(\widehat{\mathcal{T}}_R, \widehat{\mathcal{VD}})$ built from the dataset $\mathcal{D}$, a search token $\mathcal{TK}$ generated for the query point $\mathcal{Q}$, and parameter k, the secure and verifiable kNN query aims to find out the encrypted search result $\widehat{\mathcal{SR}}$, s.t. (1) $\mathcal{SR} \subseteq \mathcal{D}$; (2) $\triangle(\mathcal{P}_i, \mathcal{Q}) \le \triangle(\mathcal{P}_j, \mathcal{Q})$, $\forall \mathcal{P}_i \in \mathcal{SR} \wedge \forall \mathcal{P}_j \in \mathcal{D} - \mathcal{SR}$ while preserving data privacy, index privacy, and query privacy.

From the system point of view, the workflow of SecVKQ is shown in Fig. 4. In the initial phase, the LBS provider firstly runs the GenKey algorithm to generate related secret keys (sk_c, sk_p, pk_r) and sk_p, which will be sent to mobile users and the dual-cloud servers, respectively. Given an R-tree $\mathcal{T}_R$ and a Voronoi diagram $\mathcal{VD}$ constructed from the dataset $\mathcal{D}$, the LBS provider then runs the EncTree algorithm to generate the encrypted R-tree $\widehat{\mathcal{T}}_R$ for edge servers, and runs the PreVD algorithm to generate the pretreated Voronoi diagram $\widehat{\mathcal{VD}}$ for the dual-cloud servers. In the search phase, given an exact query point $\mathcal{Q}$ and a range Rect_Q, the mobile user encrypts $\mathcal{Q}$ with homomorphic encryption, and runs the EncRect algorithm to generate an encrypted range query $\widehat{\text{Rect}}_Q$, and then sends the search token $(\widehat{\text{Rect}}_Q, k, E(\mathcal{Q})$ to the nearby edge server. The edge server runs the SearchTree algorithm to generate the immediate result $\mathcal{IR}$, then sends $\mathcal{IR}$, and partial search token $(k, E(\mathcal{Q}))$ to the dual-cloud servers. The dual-cloud servers run the SEKS protocol to generate the encrypted result $\widehat{\mathcal{SR}}$ and the verification object $\mathcal{VO}$, and then send $(\widehat{\mathcal{SR}}, \mathcal{VO})$ to the mobile user. In the verify phase, the mobile user firstly decrypts $(\widehat{\mathcal{SR}}, \mathcal{VO})$, then verifies the decrypted result $\mathcal{SR}$ according to the verification object $\mathcal{VO}$.

4. Preliminary screening phase

4.1. Main idea

To allow secure and efficient filtering, the LBS provider builds an R-tree [37] index $\mathcal{T}_R$ from the dataset $\mathcal{D}$, encrypts each tree node with the CIPE.EncD algorithm, and uploads the encrypted R-tree index $\widehat{\mathcal{T}}_R$ to edge servers. In order to retrieve the top-k nearest POIs, the mobile user located at position $\mathcal{Q} = (x_q, y_q)$ generates a two-dimensional range query Rect_Q, encrypts each dimension with the CIPE.EncQ algorithm, and sends the encrypted range query $\widehat{\text{Rect}}_Q$ to the nearby edge server. Once receiving a query request, the edge server evaluates the encrypted query $\widehat{\text{Rect}}_Q$ over the encrypted index $\widehat{\mathcal{T}}_R$ to obtain an intermediate result $\mathcal{IR}$, with which the cloud servers can quickly find out the exact search results.

R-tree structure. The basic idea of building an R-tree is to group nearby POIs at the same level and iteratively include them to a minimal bounding box in a higher level of the tree [37]. Specifically, each leaf node $v \in \mathcal{T}_R$ is defined by $v = \langle \text{ID}_v, \mathcal{P}_i \rangle$, where ID_v is the unique identifier of node v in the tree $\mathcal{T}_R$, and $\mathcal{P}_i \in \mathcal{D}$ is a distinct POI associated with node v. In contrast, each non-leaf node $u \in \mathcal{T}_R$ is defined by $u = \langle \text{ID}_u, \text{Rect}_u, \text{child}_1, \dots, \text{child}_f \rangle$, where ID_u is node u's identifier, Rect_u is the rectangle associated with node u, and $\text{child}_1, \dots, \text{child}_f$ are the pointers to each of u's children. For example, Fig. 5 illustrates an R-tree with fanout $f = 3$ that is built from dataset $\mathcal{D} = \{\mathcal{P}_1, \dots, \mathcal{P}_8\}$, where the rectangles with solid lines denote non-leaf nodes of the index tree $\mathcal{T}_R$.

Two-dimensional range query. Suppose that the mobile user located at $\mathcal{Q} = (x_q, y_q)$ wants to issue a range query with scope L. She generates a rectangle Rect_Q by taking $\mathcal{Q}$ as the center and L as the edge length. For example, in Fig. 5, the rectangle with red dashed lines denotes range query $\mathcal{Q}$.

Secure range query on an encrypted R-tree. In clear text, the search process is a recursive procedure upon the index tree $\mathcal{T}_R$. Given a range query Rect_Q, the edge server performs a detection starting

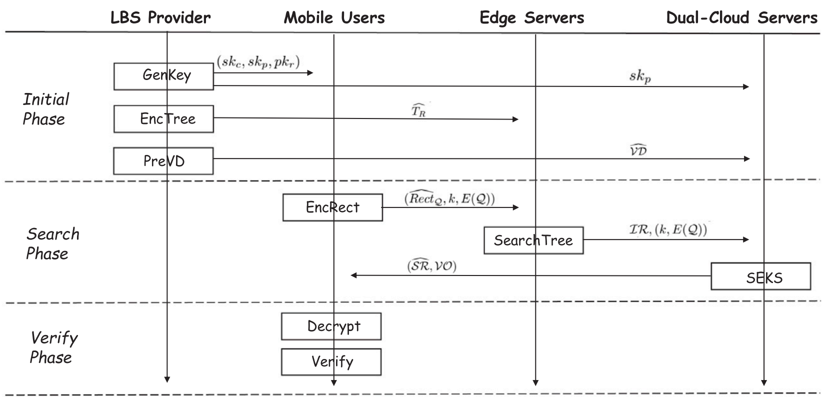


Fig. 4. System flow diagram.

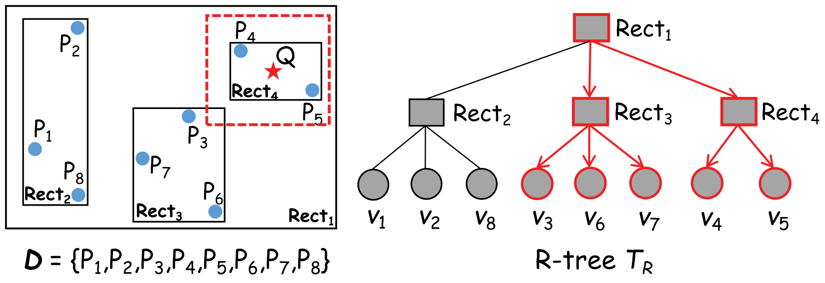


Fig. 5. An example of the construction/searching of an R-tree.

Protocol 1 Preliminary Screening Phase

```
Calculation in LBS provider
1: Run the GenKey algorithm to generate a secret key $SK$
2: Construct an R-tree $\mathcal{T}_R$ from dataset $\mathcal{D}$
3: Run the EncTree algorithm to generate an encrypted R-tree $\widehat{\mathcal{T}_R}$
Calculation in mobile user
4: Express the range query as a rectangle $\mathrm{Rect}_Q$
5: Run the EncRect algorithm to generate an encrypted query $\widehat{\mathrm{Rect}}_Q$
Calculation in edge server
6: Run the SearchTree algorithm to obtain the immediate result $\mathcal{IR}$
```

from the root node of tree $\mathcal{T}_R$. If Rect_u, the rectangle associated with node u, overlaps with Rect_Q, node u's children nodes will be checked; otherwise, the subtree rooted at node u will not be traversed. When this traversal is over, all of the reached leaf nodes are returned. For example, Fig. 5 illustrates the query process over a sample R-tree.

In order to apply the CIPE scheme, Rect_u is represented by the coordinates of its left-bottom point (x_{u_1}, y_{u_1}) and the coordinates of its right-top point (x_{u_2}, y_{u_2}). For each value in Rect_u, the CIPE.EncD algorithm is run to generate an encrypted rectangle $\widehat{\mathrm{Rect}}_u = ((\hat{x}_{u_1}, \hat{y}_{u_1}), (\hat{x}_{u_2}, \hat{y}_{u_2}))$. Let (x_{q_1}, y_{q_1}) and (x_{q_2}, y_{q_2}) denote the coordinates of the left-bottom point and right-top point of Rect_Q. In contrast, Rect_Q is represented by the query coverage on the x-coordinate $QC_x = (x_{q_1}, x_{q_2})$ and the query coverage on the y-coordinate $QC_y = (y_{q_1}, y_{q_2})$. For each query coverage in Rect_Q, the CIPE.EncQ algorithm is run to generate an encrypted rectangle $\widehat{\mathrm{Rect}}_Q = (\widehat{QC}_x, \widehat{QC}_y)$. Rect_Q does not overlap with Rect_u, if either of the following cases happen:

- Case 1: CIPE.Compare$(\widehat{x_{u_1}}, \widehat{QC}_y) = 1$
- Case 2: CIPE.Compare$(\widehat{x_{u_2}}, \widehat{QC}_x) = -1$
- Case 3: CIPE.Compare$(\widehat{y_{u_1}}, \widehat{QC}_y) = 1$
- Case 4: CIPE.Compare$(\widehat{y_{u_2}}, \widehat{QC}_y) = -1$

4.2. Details of the working process

Let $F : \{0,1\}^\kappa \times \{0,1\}^* \rightarrow \{0,1\}^\kappa$ be a pseudo-random function (PRF). As shown in Protocol 1, the preliminary screening phase consists of the following algorithms:

• GenKey$(1^\kappa, \mathcal{D}) \rightarrow SK$: Given a security parameter κ, the LBS provider runs the CIPE.GenK algorithm to generate sk_c, and chooses a random string $k_f \in \{0,1\}^\kappa$ as the key of F. The secret key is in the form of $SK = (sk_c, k_f)$.

• EncTree$(\mathcal{T}_R, SK) \rightarrow \widehat{\mathcal{T}_R}$: Given an R-tree $\mathcal{T}_R$ built from the dataset $\mathcal{D}$, an encrypted tree $\widehat{\mathcal{T}_R}$ is generated as follows:

1. For each non-leaf node $u \in \mathcal{T}_R$, the LBS provider encrypts the correlative rectangle $\mathrm{Rect}_u = ((x_{u_1}, y_{u_1}), (x_{u_2}, y_{u_2}))$ with the CIPE.EncD algorithm, and outputs an encrypted rectangle $\widehat{\mathrm{Rect}}_u = ((\hat{x}_{u_1}, \hat{y}_{u_1}), (\hat{x}_{u_2}, \hat{y}_{u_2}))$.
2. For each leaf node v of $\mathcal{T}_R$, the LBS provider generates a label $\mathcal{L}_i = F_{k_f}(x_i|y_i)$ for the correlative POI.

• EncRect$(\mathrm{Rect}_Q, SK) \rightarrow \widehat{\mathrm{Rect}}_Q$: Given a two-dimensional range query $\mathrm{Rect}_Q = (QC_x, QC_y)$, the mobile user encrypts each query coverage with algorithm CIPE.EncQ and generates an encrypted query $\widehat{\mathrm{Rect}}_Q = (\widehat{QC}_x, \widehat{QC}_y)$.

• SearchTree$(\widehat{\mathcal{T}_R}, \widehat{\mathrm{Rect}}_Q) \rightarrow \mathcal{IR}$: For each non-leaf node $u \in \widehat{\mathcal{T}_R}$, the edge server tests whether there is an overlapping between $\widehat{\mathrm{Rect}}_Q$ and $\widehat{\mathrm{Rect}}_u$. If so, node u's children nodes will be checked; otherwise, the subtree rooted at node u will not be traversed. For each reached leaf node, the edge server puts the label of a related POI into $\mathcal{IR}$.

Remark 1. Edge servers are deployed across a large area, each responsible for processing queries of nearby mobile users. Therefore, it is unnecessary for edge servers to store a large R-tree index built from the entire dataset. In order to reduce the storage space incurred on edge servers, the LBS provider can first divide the dataset $\mathcal{D}$ and then build an R-tree index from each partition. In this way, the edge server

Algorithm 2 Pretreatment of Voronoi Diagram (PreVD)

```
Require: The Voronoi diagram $\mathcal{VD}$, the key $k_f$, the maximal number of Voronoi neighbors $M$
Ensure: An encrypted and signed Voronoi diagram $\widehat{\mathcal{VD}}$
 1: Initialize $\widehat{\mathcal{VD}}$ to an empty table
 2: for each $\mathcal{P}_i \in \mathcal{VD}$ do
 3:   Initialize $\mathbf{r}_i$, $\mathsf{EncNL}_i$ and $\mathsf{EncNP}_i$ to empty lists
 4:   $\mathcal{L}_i \leftarrow F_{k_f}(x_i|y_i)$; $c_i \leftarrow |\mathcal{VN}(\mathcal{P}_i)|$
 5:   if $c_i < M$ then
 6:     Pad $\mathcal{VN}(\mathcal{P}_i)$ with $M - c_i$ dummy data points
 7:   for each $\mathcal{P}_j \in \mathcal{VN}(\mathcal{P}_i)$ do
 8:     $\mathcal{L}_j \leftarrow F_{k_f}(x_j|y_j)$
 9:     Append $E(\mathcal{L}_j)$ to $\mathsf{EncNL}_i$ and $E(\mathcal{P}_j)$ to $\mathsf{EncNP}_i$
10:   Generate a signature $\sigma_i$ for $\mathcal{P}_i$ and $\mathcal{VN}(\mathcal{P}_i)$
11:   $\mathbf{r}_i \leftarrow (E(\mathcal{P}_i), \mathsf{EncNL}_i, \mathsf{EncNP}_i, E(c_i), \sigma_i)$
12:   $\widehat{\mathcal{VD}}[\mathcal{L}_i] \leftarrow \mathbf{r}_i$
```

is able to filter results more rapidly by using a compact search index. Note that data points of the search result may be distributed among different edge servers. To avoid such a scenario, the R-tree index is constructed based on the data partitions located in a primary edge server and several adjacent edge servers. Even then, the storage space and search efficiency can be improved because the data scale stored on each edge server is much smaller than the entire dataset.

5. The exact search and verification phase

To allow cloud servers to perform an exact kNN search in a secure and verifiable way, the LBS provider preprocesses the dataset before outsourcing. Once receiving preliminary results and search tokens from edge servers, the dual-cloud servers collaborate to perform a series of secure protocols to find out exact search results from the pretreated dataset, and run the verification protocol with mobile users to validate results.

5.1. Data pre-processing

For enhanced privacy, Paillier cryptosystem is employed to encrypt the coordinates of each data point $\mathcal{P}_i$ and query point $\mathcal{Q}$. For simplicity, the encryption of coordinates (x_i, y_i) and (x_q, y_q) is expressed as $E(\mathcal{P}_i)$ and $E(\mathcal{Q})$, respectively. Furthermore, we assume that the LBS provider's public key pk_p is publicly known, and thus will be omitted from the input of algorithms/protocols for briefness.

The LBS provider first constructs a Voronoi diagram $\mathcal{VD} = (\mathcal{P}_i, \mathcal{VN}(\mathcal{P}_i))_{i\in[n]}$ from the dataset $\mathcal{D}$, and then runs Algorithm 2 to generate a pretreated Voronoi diagram $\widehat{\mathcal{VD}}$. Finally, it sends $\widehat{\mathcal{VD}}$ and the secret key sk_p to cloud servers $\mathbf{C}_1$ and $\mathbf{C}_2$, respectively. In Algorithm 2, $\widehat{\mathcal{VD}}$ is constructed as a search table indexed by the labels of data points. Let k_f be the key of PRF F, and let $M = \max\{|\mathcal{VN}(\mathcal{P}_1)|, \ldots, |\mathcal{VN}(\mathcal{P}_n)|\}$ denote the maximal number of Voronoi neighbors in $\mathcal{VD}$ (M is a small value even in the large-scale dataset). Algorithm 2 works in the following way.

For each data point $\mathcal{P}_i \in \mathcal{VD}$, the LBS provider first calculates its label $\mathcal{L}_i$ and then encrypts its coordinates into $E(\mathcal{P}_i)$ and its real number of Voronoi neighbors into $E(c_i)$. Then, for each Voronoi neighbor $\mathcal{P}_j$ of $\mathcal{P}_i$, the LBS provider calculates $\mathcal{P}_j$'s label and appends the encrypted label $E(\mathcal{L}_j)$ and the encrypted position $E(\mathcal{P}_j)$ to the end of list EncNL_i and list EncNP_i, respectively. For security, the LBS provider pads $\mathcal{VN}(\mathcal{P}_i)$ with dummy data points to hide the real number of Voronoi neighbors. It is worth noticing that the dummy data points are randomly chosen from the dataset and will not affect the accuracy of kNN queries. Finally, the LBS provider generates an RSA signature σ_i for the information about $(\mathcal{P}_i, \mathcal{VN}(\mathcal{P}_i))$, and puts the entry

Protocol 3 Secure Nearest Neighbor

```
Require: $\mathbf{C}_1$ has a set of encrypted data points $\widehat{\mathcal{S}} = \{E(\mathcal{P}_1), \ldots, E(\mathcal{P}_t)\}$ and an encrypted query point $E(\mathcal{Q})$; $\mathbf{C}_2$ has the secret key $sk_p$
Ensure: $\mathbf{C}_1$ obtains an encrypted point $E(\mathcal{P}_{1^*})$
                    Collaboration in $\mathbf{C}_1$ and $\mathbf{C}_2$
1: $\mathbf{C}_1$ sets $E(\mathcal{P}_{1^*}) \leftarrow E(\mathcal{P}_1)$
2: for each encrypted data point $E(\mathcal{P}_i) \in \widehat{\mathcal{S}}$ do
3:   $\mathbf{C}_1$ and $\mathbf{C}_2$ run the SSED protocol and $\mathbf{C}_1$ obtains $E(\triangle\,(\mathcal{P}_{1^*}, \mathcal{Q}))$ and $E(\triangle\,(\mathcal{P}_i, \mathcal{Q}))$
4:   $\mathbf{C}_1$ and $\mathbf{C}_2$ run the SC protocol to compare $E(\triangle\,(\mathcal{P}_{1^*}, \mathcal{Q}))$ and $E(\triangle\,(\mathcal{P}_i, \mathcal{Q}))$
5:   if $E(\triangle\,(\mathcal{P}_{1^*}, \mathcal{Q})) > E(\triangle\,(\mathcal{P}_i, \mathcal{Q}))$ then
6:     $\mathbf{C}_1$ sets $E(\mathcal{P}_{1^*}) \leftarrow E(\mathcal{P}_i)$
```

$(E(\mathcal{P}_i), \mathsf{EncNL}_i, \mathsf{EncNP}_i, E(c_i), \sigma_i)$ at $\widehat{\mathcal{VD}}[\mathcal{L}_i]$. On the basis of the Voronoi diagram shown in Fig. 2, the result of data pre-processing is shown in Fig. 6. For instance, $\mathcal{L}_1$ stores the label that is calculated by coordinates of data point $\mathcal{P}_1$. $E(\mathcal{P}_1)$ stores the encrypted point coordinates. EncNL_1 and EncNP_1 are the encrypted labels and the encrypted data coordinates of $\mathcal{P}_1$'s Voronoi neighbors, respectively. The information of dummy data points is highlighted in blue in the figure. The real Voronoi neighbors of $\mathcal{P}_1$ are $\{\mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_7, \mathcal{P}_8\}$, and dummy neighbors of $\mathcal{P}_1$ are $\{\mathcal{P}_4, \mathcal{P}_5\}$. $E(c_1)$ stores the real number of $\mathcal{P}_1$'s Voronoi neighbors. At last, σ_1 is RSA signature for the information about $(\mathcal{P}_1, \mathcal{VN}(\mathcal{P}_1))$.

5.2. Secure protocols in exact kNN search

In the exact search phase, the dual-cloud servers, $\mathbf{C}_1$ and $\mathbf{C}_2$, collaborate to run the following protocols:

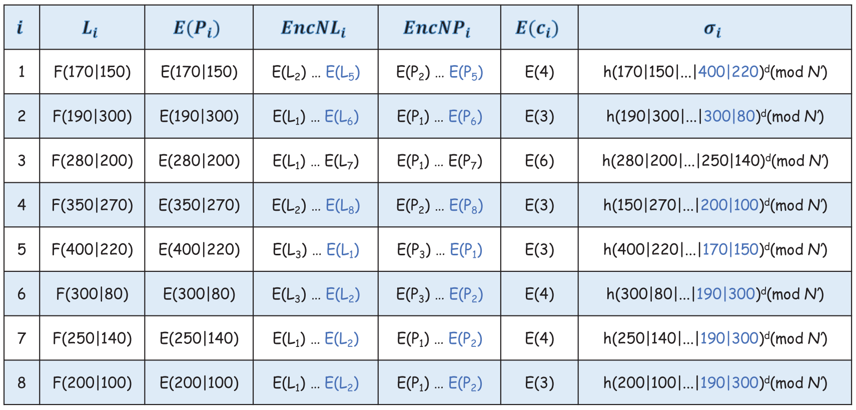
Secure Nearest Neighbor (SNN). This protocol allows cloud servers to securely compute the nearest neighbor of an encrypted query point from a set of encrypted data points. Let $\widehat{\mathcal{S}} = \{E(\mathcal{P}_1), \ldots, E(\mathcal{P}_t)\}$ be the encrypted form of clear set $\mathcal{S} = \{\mathcal{P}_1, \ldots, \mathcal{P}_t\}$. Specifically, $\mathbf{C}_1$ takes a set of encrypted data points $\widehat{\mathcal{S}}$ and an encrypted query point $E(\mathcal{Q})$ as input. $\mathbf{C}_2$ takes the secret key sk_p as input. At the end of this protocol, $\mathbf{C}_1$ obtains an encrypted point $E(\mathcal{P}_1^*)$ s.t. $\triangle\,(\mathcal{P}_1^*, \mathcal{Q}) \leq \triangle\,(\mathcal{P}_i, \mathcal{Q})$ where $\mathcal{P}_1^* \in \mathcal{S}$ and $\mathcal{P}_i \in \mathcal{S} - \mathcal{P}_1^*$.

Protocol 3 describes the details of SSN. Specifically, $\mathbf{C}_1$ uses $E(\mathcal{P}_1^*)$ to denote the nearest neighbor of the encrypted query point $E(\mathcal{Q})$. For each encrypted point $E(\mathcal{P}_i) \in \widehat{\mathcal{S}}$, $\mathbf{C}_1$ and $\mathbf{C}_2$ run the SSED protocol to obtain the encrypted distance $E(\triangle\,(\mathcal{P}_i, \mathcal{Q}))$ and then they run the SC protocol to set $E(\mathcal{P}_1^*)$ to the encrypted point with minimal distance from $E(\mathcal{Q})$.

Secure Set Difference (SSD). This protocol allows cloud servers to securely perform set difference on two encrypted sets. Let $\widehat{S}_1 = \{E(x_1), \ldots, E(x_N)\}$ and $\widehat{S}_2 = \{E(y_1), \ldots, E(y_T)\}$ be the encrypted forms of clear sets $S_1 = \{x_1, \ldots, x_N\}$ and $S_2 = \{y_1, \ldots, y_T\}$, respectively. Initially, $\mathbf{C}_1$ takes two encrypted sets, $\widehat{S}_1$ and $\widehat{S}_2$, as input, and $\mathbf{C}_2$ takes the secret key sk_p as input. At last, $\mathbf{C}_1$ obtains an encrypted set $\widehat{S}' = \{E(x_1^*), \ldots, E(x_{t^*})\}_{x_{i^*} \in S_1 - S_2}$, where the plaintext of each element belongs to S_1 but not to S_2.

Protocol 4 describes the details of SSD. Let π and π^{-1} denote a pseudo-random permutation and an inverse permutation, respectively. For each element $E(x_i) \in \widehat{\mathrm{S}}_1$, $\mathbf{C}_1$ checks whether there exists an element $E(y_j) \in \widehat{\mathrm{S}}_2$, such that $x_i = y_j$. Due to the semantic security of Paillier cryptosystem, $\mathbf{C}_1$ cannot determine the equality of two clear values based on their ciphertexts. Therefore, $\mathbf{C}_1$ requests $\mathbf{C}_2$ to perform decryption. To protect data privacy, $\mathbf{C}_1$ sends $t_{i,j} \leftarrow E(r_{i,j} \times (x_i - y_j))$, instead of two encrypted sets, to $\mathbf{C}_2$. Therefore, if $x_i = y_j$, $\mathbf{C}_2$ obtains a value of zero after decryption, otherwise, $\mathbf{C}_2$ knows nothing about the data content.

Secure Exact kNN Search (SEKS). This protocol allows cloud servers to securely compute the top-k nearest encrypted data points of an encrypted query point. Specifically, $\mathbf{C}_1$ takes the immediate result



i	L_i	$E(P_i)$	$EncNL_i$	$EncNP_i$	$E(c_i)$	σ_i
1	F(170|150)	E(170|150)	E(L_2) ... E(L_5)	E(P_2) ... E(P_5)	E(4)	h(170|150|...|400|220)^d(mod N')
2	F(190|300)	E(190|300)	E(L_1) ... E(L_6)	E(P_1) ... E(P_6)	E(3)	h(190|300|...|300|80)^d(mod N')
3	F(280|200)	E(280|200)	E(L_1) ... E(L_7)	E(P_1) ... E(P_7)	E(6)	h(280|200|...|250|140)^d(mod N')
4	F(350|270)	E(350|270)	E(L_2) ... E(L_8)	E(P_2) ... E(P_8)	E(3)	h(150|270|...|200|100)^d(mod N')
5	F(400|220)	E(400|220)	E(L_3) ... E(L_1)	E(P_3) ... E(P_1)	E(3)	h(400|220|...|170|150)^d(mod N')
6	F(300|80)	E(300|80)	E(L_3) ... E(L_2)	E(P_3) ... E(P_2)	E(4)	h(300|80|...|190|300)^d(mod N')
7	F(250|140)	E(250|140)	E(L_1) ... E(L_2)	E(P_1) ... E(P_2)	E(4)	h(250|140|...|190|300)^d(mod N')
8	F(200|100)	E(200|100)	E(L_1) ... E(L_2)	E(P_1) ... E(P_2)	E(3)	h(200|100|...|190|300)^d(mod N')

Fig. 6. Data Pre-processing example.

Protocol 4 Secure Set Difference.

```
Require: $\mathbf{C}_1$ has two sets of encrypted values $\widehat{S}_1 = \{E(x_1), \dots, E(x_N)\}$
    and $\widehat{S}_2 = \{E(y_1), \dots, E(y_T)\}$; $\mathbf{C}_2$ has the secret key $sk_p$
Ensure: $\mathbf{C}_1$ obtains an encrypted difference set $\widehat{S}'$
                    Calculation in $\mathbf{C}_1$
 1: Initialize T to an empty table
 2: for the $i$-th element $E(x_i) \in \widehat{S}_1$ do
 3:     Initialize t to an empty list
 4:     for all $E(y_j) \in \widehat{S}_2$ in random order do
 5:         Generate a random number $r_{i,j}$
 6:         $t_{i,j} \leftarrow (E(x_i) \times E(y_j)^{-1})^{r_{i,j}}$
 7:         {equivalent to $t_{i,j} \leftarrow E(r_{i,j} \times (x_i - y_j))$}
 8:         Append $t_{i,j}$ to t
 9:     T[$\pi(i)$] $\leftarrow$ t
10: Send T to $\mathbf{C}_2$
                    Calculation in $\mathbf{C}_2$
11: Initialize $V$ to an empty set
12: for $i \in [N]$ do
13:     Parse T[$i$] as $(t_{i,1}, \dots, t_{i,T})$
14:     if $\exists t_{i,j} \in$ T[$i$] $\wedge D(t_{i,j}) = 0$ then
15:         Add $i$ into set V
16: Send set $V$ to $\mathbf{C}_1$
                    Calculation in $\mathbf{C}_1$
17: for each element $i$ in $V$ do
18:     $j \leftarrow \pi^{-1}(i)$
19:     Remove the $j$-th element $E(x_j)$ from $\widehat{S}_1$
20: $\widehat{S}' \leftarrow \widehat{S}_1$
```

$\mathcal{IR}$, the encrypted query point $E(Q)$, parameter k, and the encrypted Voronoi diagram $\widehat{\mathcal{VD}}$ as input. $\mathbf{C}_2$ takes the secret key sk_p as input. At the end of this protocol, $\mathbf{C}_1$ obtains the encrypted search result $\widehat{\mathcal{SR}} = ((\mathcal{L}_1^*, E(\mathcal{P}_1^*)), \dots, (\mathcal{L}_k^*, E(\mathcal{P}_k^*))$, s.t. (1) $\mathcal{SR} \subseteq \mathcal{D}$; (2) $d(\mathcal{P}_i^*, Q) \le d(\mathcal{P}_j, Q)$, $\forall \mathcal{P}_i^* \in \mathcal{SR} \wedge \forall \mathcal{P}_j \in \mathcal{D} - \mathcal{SR}$.

As shown in Protocol 5, SEKS, as the main protocol, invokes the SNN and SSD protocols as sub-routines as follows. For all labels $\mathcal{L}_i \in \mathcal{IR}$, $\mathbf{C}_1$ puts the corresponding encrypted data point $E(\mathcal{P}_i)$ into set $\widehat{C}$. If the size of $\widehat{C}$ is equal or greater than k, $\mathbf{C}_1$ and $\mathbf{C}_2$ run the SNN protocol k times to obtain the top-k nearest encrypted data points $\{E(\mathcal{P}_1^*), \dots, E(\mathcal{P}_k^*)\}$. Otherwise, $\mathbf{C}_1$ and $\mathbf{C}_2$ collaborate to find 1NN, …, kNN in order based on $\widehat{\mathcal{VD}}$. For the correctness of result, the SSD protocol is used to remove repeated neighbors and avoid repeated comparisons with the data points already in $\widehat{\mathcal{SR}}$.

For example, as shown in Fig. 5, after the preliminary screening phase, the edge server returns the intermediate result $\mathcal{IR} = \{\mathcal{L}_3, \mathcal{L}_6, \mathcal{L}_7, \mathcal{L}_4, \mathcal{L}_5\}$. If parameter $k = 3$, the dual-cloud servers run

Protocol 5 Secure Exact kNN Search.

```
Require: $\mathbf{C}_1$ has the intermediate result $\mathcal{IR}$, the encrypted query point
    $E(Q)$, parameter $k$, and the encrypted Voronoi diagram $\widehat{\mathcal{VD}}$; $\mathbf{C}_2$ has
    the secret key $sk_p$
Ensure: $\mathbf{C}_1$ obtains the encrypted search result $\widehat{\mathcal{SR}}$
                    Collaboration in $\mathbf{C}_1$ and $\mathbf{C}_2$
 1: $\mathbf{C}_1$ initializes $\widehat{C}$, $\widehat{\mathcal{SR}}$, and $\widehat{V}$ to empty sets
 2: for each label $\mathcal{L}_i \in \mathcal{IR}$ do
 3:     $\mathbf{C}_1$ locates entry $\widehat{\mathcal{VD}}[\mathcal{L}_i]$ to obtain $(E(\mathcal{P}_i), \mathsf{EncNL}_i,$
        $\mathsf{EncNP}_i, E(c_i), \sigma_i)$ and appends $E(\mathcal{P}_i)$ to $\widehat{C}$
 4: if $|\widehat{C}| \ge k$ then
 5:     $\mathbf{C}_1$ with input $(\widehat{C}, E(Q))$ and $\mathbf{C}_2$ with input $sk_p$ run the SNN
        protocol $k$ times, and $\mathbf{C}_1$ obtains $\{E(\mathcal{P}_{1^*}), \dots, E(\mathcal{P}_{k^*})\}$
 6:     $\mathbf{C}_1$ puts $(\mathcal{L}_{i^*}, E(\mathcal{P}_{i^*}))_{i=1}^k$ into $\widehat{\mathcal{SR}}$
 7: else
 8:     $\mathbf{C}_1$ with input $(\widehat{C}, E(Q))$ and $\mathbf{C}_2$ with input $sk_p$ run the SNN
        protocol, and $\mathbf{C}_1$ obtains $E(\mathcal{P}_{1^*})$
 9:     $\mathbf{C}_1$ puts $(\mathcal{L}_{1^*}, E(\mathcal{P}_{1^*}))$ into $\widehat{\mathcal{SR}}$
10:     $\mathbf{C}_1$ puts $E(\mathcal{L}_{1^*})$ into $\widehat{V}$ and sets $\widehat{C}$ to an empty set
11:     while $|\widehat{\mathcal{SR}}| < k$ do
12:         $\mathbf{C}_1$ locates $\widehat{\mathcal{VD}}[\mathcal{L}_{1^*}]$ to obtain $(E(\mathcal{P}_{1^*}),$
            $\mathsf{EncNL}_{1^*}, \mathsf{EncNP}_{1^*}, E(c_{1^*}), \sigma_{1^*})$
13:         $\mathbf{C}_1$ with input $(\mathsf{EncNL}_{1^*}, \widehat{V})$ and $\mathbf{C}_2$ with input $sk_p$ run the SSD
            protocol, and $\mathbf{C}_1$ obtains the encrypted difference set $\mathsf{EncNL}'_{1^*}$
14:         for $E(\mathcal{P}_j)$ in $\mathsf{EncNP}_{1^*} \wedge E(\mathcal{L}_j) \in \mathsf{EncNL}'_{1^*}$ do
15:             $\mathbf{C}_1$ puts $E(\mathcal{P}_j)$ into $\widehat{C}$ and $E(\mathcal{L}_j)$ into $\widehat{V}$
16:         $\mathbf{C}_1$ with input $(\widehat{C}, E(Q))$ and $\mathbf{C}_2$ with input $sk_p$ run the SNN
            protocol, $\mathbf{C}_1$ obtains $E(\mathcal{P}_{1^*})$ and removes it from the set $\widehat{C}$
17:         $\mathbf{C}_1$ obtains $\mathcal{L}_{1^*}$ by asking $\mathbf{C}_2$ to perform decryption
18:         $\mathbf{C}_1$ puts $(\mathcal{L}_{1^*}, E(\mathcal{P}_{1^*}))$ into $\widehat{\mathcal{SR}}$
```

the SNN protocol for 3 times and $\mathbf{C}_1$ obtains the search result $\widehat{\mathcal{SR}} = \{(\mathcal{L}_4, E(\mathcal{P}_4)), (\mathcal{L}_5, E(\mathcal{P}_5)), (\mathcal{L}_3, E(\mathcal{P}_3))\}$. If parameter $k = 6$, the dual-cloud servers first run the SNN protocol to obtain the top-1 nearest neighbor $E(\mathcal{P}_4)$. After putting $E(\mathcal{L}_4)$ into set $\widehat{V}$, $\mathbf{C}_1$ locates $\widehat{\mathcal{VD}}[\mathcal{L}_4]$ and puts $\{E(\mathcal{P}_2), E(\mathcal{P}_3), E(\mathcal{P}_5)\}$ and $\{E(\mathcal{L}_2), E(\mathcal{L}_3), E(\mathcal{L}_5)\}$ into $\widehat{C}$ and $\widehat{V}$, respectively. Next, $\mathbf{C}_1$ and $\mathbf{C}_2$ run the SSD protocol to remove the repeated encrypted labels from EncNL_4 and then run the SNN protocol to get the top-2 nearest neighbor $E(\mathcal{P}_5)$. Then, $\mathbf{C}_1$ asks $\mathbf{C}_2$ to decrypt $E(\mathcal{L}_5)$, and the dual-cloud servers perform in the same way to find out the top-i nearest neighbor, for $i = 3, \dots, 6$.

Remark 2. The data packing/unpacking technique [38] allows the LBS provider/cloud servers to pack multiple small values $x_1, \dots, x_t$ into one large value X before encryption and unpack X into $x_1, \dots, x_t$ after decryption. By using this technique, the computational and communication overheads incurred by Paillier cryptosystem can be largely reduced.

Remark 3. In order to reduce the computational costs on mobile users, the dual-cloud servers may participate in the decryption process as follows: $\mathbf{C}_1$ blinds $\widehat{\mathcal{SR}}$ with a set of random numbers R and sends the blinded result $\widehat{\mathcal{SR}}'$ and R to $\mathbf{C}_2$ and the mobile user, respectively. $\mathbf{C}_2$ decrypts the elements in $\widehat{\mathcal{SR}}'$ with sk_p and returns $\mathcal{SR}'$ to the mobile user. Therefore, the mobile user with $\mathcal{SR}'$ and R can quickly recover the plaintexts without performing Paillier decryption.

5.3. Verification of search results

To allow mobile users to validate search results, the cloud servers generate a verification object $\mathcal{VO} = (\mathcal{VO}_{sig}, \mathcal{VO}_{geo})$ as follows. For each $\mathcal{L}(\mathcal{P}_i) \in \widehat{\mathcal{SR}}$, $\mathbf{C}_1$ puts the RSA signature σ_i into $\mathcal{VO}_{sig}$ and puts $(\mathsf{EncNP}_i, E(c_i))$ into $\mathcal{VO}_{geo}$. Specifically, $\mathbf{C}_1$ and $\mathbf{C}_2$ may participate in the process as described in Remark 3 to partially decrypt $\mathcal{VO}_{geo}$. Furthermore, $\mathcal{VO}_{sig}$ can be compressed into a single signature $\sigma_{1,k}$ for batch verification due to the aggregation property of a condensed RSA signature. In our experiments, the size of $\mathcal{VO}_{sig}$ is constant (512bits), thus the communication overhead is largely reduced.

Once receiving the search result $\widehat{\mathcal{SR}}$ and verification object $\mathcal{VO} = (\mathcal{VO}_{sig}, \mathcal{VO}_{geo})$, the mobile user first decrypts $\widehat{\mathcal{SR}}$ to obtain the plaintext result $\mathcal{SR}$, and then recovers the contents in $\mathcal{VO}_{geo}$. $\mathcal{SR}$ can be validated from the following aspects:

• **Authenticity.** The mobile user aggregates the information about $\mathcal{P}_i$ and $\mathcal{VN}(\mathcal{P}_i)$ for $\mathcal{P}_i \in \mathcal{SR}$, and then verifies the correctness of $\mathcal{VO}_{sig}$ with this aggregated information. If $\mathcal{VO}_{sig}$ passes the test, she determines that all of the data points returned are indeed from the dataset $\mathcal{D}$.

• **Completeness.** Suppose that $\mathcal{P}_i^*$ denotes the top-i nearest result in $\mathcal{SR}$ for $i \in [k]$. As shown in Section 2, the mobile user first reconstructs $\mathcal{VC}(\mathcal{P}_1^*)$, based on $\mathcal{VN}(\mathcal{P}_1^*)$ and $\mathcal{P}_1^*$, and then checks whether $\mathcal{Q}$ locates in $\mathcal{VC}(\mathcal{P}_1^*)$ or not. If so, she confirms that $\mathcal{P}_1^*$ is the nearest neighbor. Otherwise, she determines that $\mathcal{SR}$ is incomplete and terminates the verification process. If $\mathcal{P}_1^*$ passes the test, for $i = 2, \dots, k$, she calculates the distance between $\mathcal{Q}$ and all points in $\mathcal{VN}(\mathcal{P}_1^*) \cup \cdots \cup \mathcal{VN}(\mathcal{P}_i^*)$ and tests whether $\mathcal{P}_i^*$ is the top-i nearest neighbor. If so, she confirms the correctness of the top-i result. Otherwise, she determines that $\mathcal{SR}$ is incomplete and terminates the verification process.

For example, the search result of a 3NN query is $\mathcal{SR} = \{\mathcal{P}_4, \mathcal{P}_5, \mathcal{P}_3\}$ in Fig. 2. The mobile user can quickly recover the plaintexts of the search result $\mathcal{SR}$ and the verification object $\mathcal{VO}$ according to Remark 3. Specifically, $\mathcal{VO}$ is composed of two parts: $\mathcal{VO}_{sig} = \sigma_{1,k}$ and $\mathcal{VO}_{geo} = \{\mathcal{VN}(\mathcal{P}_4), c_4, \mathcal{VN}(\mathcal{P}_5), c_5, \mathcal{VN}(\mathcal{P}_3), c_3\}$, which can be used to verify the authenticity and completeness of $\mathcal{SR}$, respectively. The mobile user first eliminates dummy data points in $\mathcal{VO}_{geo}$. In terms of authenticity, $\mathcal{VO}_{sig}$ contains the condensed RSA signature signed $\sigma_{1,k}$, with which the mobile user can confirm that the data points in $\mathcal{SR}$ are from dataset $\mathcal{D}$. Specifically, the mobile user verifies the condensed signature $\sigma_{1,k}$ by checking if: $\sigma_{1,k}^e = (\prod_{i=1}^k h(\mathcal{P}_i|\mathcal{VN}(\mathcal{P}_i))) \mod N'$, where $\mathcal{P}_i \in \mathcal{SR}$ and $\mathcal{VN}(\mathcal{P}_i) \in \mathcal{VO}_{geo}$. In terms of completeness, The mobile user reconstructs $\mathcal{VC}(\mathcal{P}_4)$ based on $\mathcal{VN}(\mathcal{P}_4)$ and $\mathcal{P}_4$, then determines that $\mathcal{Q}$ is located in $\mathcal{VC}(\mathcal{P}_4)$. According to Property 1 in Section 2, the mobile user confirms that the nearest neighbor of $\mathcal{Q}$ is $\mathcal{P}_4$. According to Property 2 in Section 2, $\mathcal{P}_5$ should belong to $\mathcal{VN}(\mathcal{P}_4)$, and $\mathcal{P}_3$ should belong to $\mathcal{VN}(\mathcal{P}_4) \cup \mathcal{VN}(\mathcal{P}_5)$. Therefore, for each data point $\mathcal{P}_i \in \mathcal{VN}(\mathcal{P}_4)$, the mobile user calculates the distance between $\mathcal{P}_i$ and $\mathcal{Q}$ and confirms that $\mathcal{P}_5$ is the second nearest neighbor of $\mathcal{Q}$. Similarly, the mobile user confirms that $\mathcal{P}_3$ is the third nearest neighbor of $\mathcal{Q}$ by computing the distance between $\mathcal{Q}$ and each data point in $\mathcal{VN}(\mathcal{P}_4) \cup \mathcal{VN}(\mathcal{P}_5)$.

6. Complexity and security analysis

6.1. Computational complexity

To show the efficiency of our SecVKQ framework, we will theoretically analyze the computational complexity of the two-phase search process and compare it with existing secure kNN search schemes. For the sake of illustration, we only consider the operations related to data encryption/decryption. Let enc_l and dec_l denote the encryption and decryption operations in the light-weighted CIPE scheme, and let enc_p and dec_p denote the encryption and decryption operations in the expensive Paillier cryptosystem, respectively.

In the data pre-processing stage, the LBS provider builds an R-tree index $\mathcal{T}_R$ of fanout 2 from the dataset $\mathcal{D}$ and encrypts $\mathcal{T}_R$ with the CIPE scheme. Furthermore, it constructs a Voronoi diagram $\mathcal{VD}$ from $\mathcal{D}$, and encrypts $\mathcal{VD}$ with the Paillier cryptosystem. The computational complexity is $O(n)enc_l + O(n \times M)enc_p$, where n is the size of $\mathcal{D}$ and M is the maximal number of Voronoi neighbors in $\mathcal{VD}$.

To retrieve the top-k nearest POIs, the mobile user generates a search token $\mathcal{TK} = (\widehat{\text{Rect}}_Q, k, E(Q))$ to the nearby edge server, where $\widehat{\text{Rect}}_Q$ is the encrypted two-dimensional range query under the CIPE scheme, and $E(Q)$ is the encrypted query point under the Paillier cryptosystem. Once receiving the blinded search result $\mathcal{SR}'$ and a set of random numbers R from the cloud server, the mobile user can recover the plaintext result without performing Paillier decryption. Therefore, the computational complexity is $O(1)(enc_l + enc_p)$.

In the preliminary screening phase, the edge server evaluates encrypted range query $\widehat{\text{Rect}}_Q$ over the encrypted R-tree index $\widehat{\mathcal{T}}_R$ to find out the intermediate result $\mathcal{IR}$. Therefore, the computational complexity on the mobile user is $O(\log n)dec_l$.

The case in the exact search phase is relatively complicated, since the dual-cloud servers need to run a series of secure protocols. As described in [26,35], the preliminary SSED and SC protocols need a constant number of Paillier encryption and decryption operations. The sub-protocol SNN needs to run the SSED and SC protocols for $|\widehat{S}|$ times to find out the query point's nearest neighbor from a set of encrypted data points $\widehat{S}$. The sub-protocol SSD needs to perform Paillier decryption operations for $|\widehat{S}_1| \times |\widehat{S}_2|$ times to compute the set difference on two encrypted sets, $\widehat{S}_1$ and $\widehat{S}_2$. Note that by using the data packing/unpacking technique [38], the time of decryption operations can be reduced to $|\widehat{S}_1|$. If the size of $\mathcal{IR}$ is larger than k, the SEKS protocol needs to run the SSED protocol I times and the SNN protocol k times, where I is the size of the intermediate result. Therefore, the search complexity is $O(k \times I)(enc_p + dec_p)$. Otherwise, the SEKS protocol needs to run the SSD and SNN protocols for k times based on the pretreated Voronoi diagram $\widehat{\mathcal{VD}}$. Therefore, the search complexity is $O(I + k^2 \times M)(enc_p + dec_p)$, where M is the maximal number of Voronoi neighbors. The comparison results between SecVKQ and existing secure kNN search schemes are summarized in Table 2. Note that the work in [26,27] cannot support result verification and thus consumes less overhead in the data preprocessing stage.

6.2. Security analysis

The security of the preliminary screening phase is based on the CIPE scheme, which has been proven to be secure in the known-plaintext model [36]. Therefore, we mainly analyze the exact search phase through proving the security of the proposed SNN, SSD, and SEKS protocols. Following the formal definition of multi-party computation introduced in [28,38], we adopt the framework of simulation paradigm [39] to analyze the security of the proposed protocols.

Theorem 1 (*Composition Theorem [39]*). *Given a protocol Ω consisting of multiple sub-protocols, if all the sub-protocols are secure and all the intermediate results are random or pseudo-random, the protocol Ω is considered secure.*

Table 2
Complexity Analysis.

	LBS Provider	Mobile User	Edge	Cloud
SecVKQ	$O(n)enc_l + O(n \times M)enc_p$	$O(1)(enc_l + enc_p)$	$O(\log n)dec_l$	$O(I + k^2 \times M)(enc_p + dec_p)$
Ref. [28]	$O(m^2 \times g + n \times M)enc_p$	$O(1)enc_p$	–	$O(k \times (n + M))enc_p + O(k \times (\sqrt{n} + M))dec_p$
Ref. [27]	$O(n)enc_p$	$O(1)enc_p$	–	$O(\sqrt{n} \times (l + k \times l \times \log n))(enc_p + dec_p)$
Ref. [26]	$O(n)enc_p$	$O(1)enc_p$	–	$O(n \times (l + k \times l \times \log n))(enc_p + dec_p)$

Notations: n denotes the size of dataset $\mathcal{D}$, k denotes the search parameter for kNN search, I denotes the size of immediate result, and M denotes the maximal number of Voronoi neighbors. m denotes the number of grids and g denotes the maximal number of grid points in [28]. l denotes the domain size (in bits) of the squared Euclidean distance in [26,27].

Theorem 2. *The SNN protocol is secure if the SSED and SC protocols are secure.*

Proof. The SNN protocol is based on the SSED and SC protocols, the security of which has been proven in previous work [26,35]. Since all of intermediate results are random, the SNN protocol is secure based on Theorem 1. ■

To analyze protocols SSD and SEKS, a simulator is constructed to simulate the actual execution view. According to the simulation paradigm, a protocol is considered to be secure if a probabilistic polynomial-time (PPT) adversary cannot distinguish between the real view and the simulated view.

Theorem 3. *The SSD protocol is secure for all efficient adversaries* $\mathbf{Adv}$*, if there exists a simulator* Sim *and the probability* $\Pr(\mathrm{Real}^{\mathbf{Adv}}_{SSD}) - \Pr(\mathrm{Sim}^{\mathbf{Adv}}_{SSD})$ *is negligible.*

Proof. The real view $\mathrm{Real}^{\mathbf{Adv}}_{SSD}$ and simulated view $\mathrm{Sim}^{\mathbf{Adv}}_{SSD}$ are defined as follows:

• $\mathrm{Real}^{\mathbf{Adv}}_{SSD}$: $\mathbf{C}_1$ has two sets of encrypted values $\hat{S}_1 = \{E(x_1), \dots, E(x_N)\}$, $\hat{S}_2 = \{E(y_1), \dots, E(y_T)\}$ and $\mathbf{C}_2$ has the secret key sk_p. In the SSD protocol, $\mathbf{C}_1$ first uses the additive homomorphism property of Paillier cryptosystem to compute an obfuscated difference $t_{i,j}$ between each encrypted data $E(x_i)$ in $\hat{S}_1$ and $E(y_j)$ in $\hat{S}_2$. Next, all differences are permuted before being sent to $\mathbf{C}_2$, which decrypts the differences and puts the permutated indexes corresponding to the value of zero into the intermediate result. At last, $\mathbf{C}_1$ outputs the final result $\hat{S}' = \{E(x_{1^*}), \dots, E(x_{t^*})\}_{x_{i^*} \in S_1 - S_2}$.

• $\mathrm{Sim}^{\mathbf{Adv}}_{SSD}$: The simulator Sim receives $\hat{S}'$, and then generates two sets of random values $x'_1, \dots, x'_{N-t}$ and $y'_1, \dots, y'_{T-N+t}$ where $x'_i < 0$ and $y'_j < 0$ for $i \in [N-t]$ and $j \in [T-N+t]$, respectively. This guarantees that x'_i and y'_j are outside the set $\{x_{1^*}, \dots, x_{t^*}\}$. Then, Sim encrypts $x'_1, \dots, x'_{N-t}$ twice and $y'_1, \dots, y'_{T-N+t}$ once with Paillier cryptosystem. Next, Sim constructs the simulated sets by setting $\tilde{S}_1 = \{E(x_{1^*}), \dots, E(x_{t^*}), E_1(x'_1), \dots, E_1(x'_{N-t})\}$, and $\tilde{S}_2 = \{E_2(x'_1), \dots, E_2(x'_{N-t}), E(y'_1), \dots, E(y'_{T-N+t})\}$, where $E_1(x'_i)$ and $E_2(x'_i)$ denote the first and second encryption operations, respectively. Due to the semantic security of Paillier cryptosystem, the ciphertext of $E_2(x'_i)$ looks different from that of $E_1(x'_i)$. After shuffling $\tilde{S}_1$ and $\tilde{S}_2$, Sim executes the SSD protocol and outputs the results $\tilde{S}'$.

Intuitively, the simulated view is computationally indistinguishable from the actual execution view because their outputs are identical. Therefore, the SSD protocol is secure. ■

Theorem 4. *The SEKS protocol is secure for all efficient adversaries* $\mathbf{Adv}$*, if there exists a simulator* Sim *and the probability* $\Pr(\mathrm{Real}^{\mathbf{Adv}}_{SEKS}) - \Pr(\mathrm{Sim}^{\mathbf{Adv}}_{SEKS})$ *is negligible.*

Proof. Similarly, we define the real view $\mathrm{Real}^{\mathbf{Adv}}_{SEKS}$ and the simulated view $\mathrm{Sim}^{\mathbf{Adv}}_{SEKS}$ as follows.

• $\mathrm{Real}^{\mathbf{Adv}}_{SEKS}$: $\mathbf{C}_1$ has the intermediate result $\mathcal{IR}$, the encrypted query point $E(Q)$, parameter k, and the encrypted Voronoi diagram $\widehat{\mathcal{VD}}$, and $\mathbf{C}_2$ has the secret key sk_p. During the SEKS protocol, $\mathbf{C}_1$ first utilizes the SSED protocol to compute the distance between $E(Q)$ and each encrypted point whose label are in the intermediate result $\mathcal{IR}$. If the size of $\mathcal{IR}$ is not less than k, $\mathbf{C}_1$ and $\mathbf{C}_2$ run the SNN protocol k times to get the search result. Otherwise, based on encrypted Voronoi diagram $\widehat{\mathcal{VD}}$, $\mathbf{C}_1$ and $\mathbf{C}_2$ run the SNN and SSD protocols to find 1NN, …, kNN in order. At last, $\mathbf{C}_1$ outputs the final result $\widehat{\mathcal{SR}}$.

• $\mathrm{Sim}^{\mathbf{Adv}}_{SEKS}$: The simulator Sim receives $\widehat{\mathcal{SR}} = \{(\mathcal{L}^*_1, E(\mathcal{P}^*_1)), \dots, (\mathcal{L}^*_k, E(\mathcal{P}^*_k))\}$. Then, Sim generates a set of encrypted points $\hat{S} = \{E(\mathcal{P}'_1), \dots, E(\mathcal{P}'_{n-k})\}$ in the following way: For $i \in [n-k]$, it generates a random point $\mathcal{P}_i$, computes its label as $\mathcal{L}_i$, encrypts its coordinates into $E(\mathcal{P}_i)$, and powers $E(\mathcal{P}_i)$ by a large random number r_i to obtain $E(\mathcal{P}'_i)$. This guarantees that all points in $\hat{S}$ are farther from the encrypted query point $E(Q)$ compared to any point in $\widehat{\mathcal{SR}}$. Next, Sim constructs a simulated dataset $E(\mathcal{D}') = \{E(\mathcal{P}^*_1), \dots, E(\mathcal{P}^*_k), E(\mathcal{P}'_1), \dots, E(\mathcal{P}'_{n-k})\}$, and generates the simulated Voronoi diagram $\widetilde{\mathcal{VD}}$ as follows: For $i \in [k]$, it chooses random c_i points from $E(\mathcal{D}')$ to be $E(\mathcal{P}^*_i)$'s Voronoi neighbors under the condition that $\mathcal{P}^*_i \in \mathcal{VP}(\mathcal{P}^*_1) \cup \dots \mathcal{VP}(\mathcal{P}^*_{i-1})$, and then it sets $\widehat{\mathcal{VD}}[\mathcal{L}^*_i]$ to $(E(\mathcal{P}^*_i), \mathsf{EncNL}_{i^*}, \mathsf{EncNP}_{i^*}, E(c_i), \sigma_i)$. For $j \in [n-k]$, it chooses c_j points from $E(\mathcal{D}')$ to be $E(\mathcal{P}'_j)$'s Voronoi neighbors and then sets $\widehat{\mathcal{VD}}[\mathcal{L}_j]$ to $(E(\mathcal{P}'_j), \mathsf{EncNL}_j, \mathsf{EncNP}_j, E(c_j), \sigma_j)$. With the same intermediate result $\mathcal{IR}$, same encrypted query point $E(Q)$, parameter k, and simulated Voronoi diagram $\widetilde{\mathcal{VD}}$, Sim executes the SEKS protocol and outputs the results $\widetilde{\mathcal{SR}}$.

All sub-protocols have been proven to be secure. Given the identical outputs, no PPT adversary can distinguish $\mathrm{Sim}^{\mathbf{Adv}}_{SEKS}$ from $\mathrm{Real}^{\mathbf{Adv}}_{SEKS}$. Hence, the SEKS protocol is secure. ■

7. Evaluation

This section will evaluate the performance of our SecVKQ framework in terms of computational and communication costs. The experiment results related to the CIPE scheme are omitted since its execution time is negligible compared to the Paillier cryptosystem. For example, the CIPE scheme requires less than 1 s to find preliminary results from 1 million data points. To validate the effectiveness of SecVKQ in practice, we conduct experiments on two real datasets, and compare SecVKQ to the state-of-the-art work, SVkNN, which also utilizes Paillier cryptosystem and the dual-cloud model to achieve secure exact kNN queries.

7.1. Parameter setting

The protocols run by the dual-cloud servers are deployed on an Aliyun ECS instance (Intel Xeon E5-2682 2.5 GHz CPU and 128G RAM); the protocols run by the mobile users are deployed on a PC machine (Intel Core i5 3.2 GHz CPU and 8G RAM); the rest of protocols are deployed on a local server (Intel Xeon Gold 5218 2.3 GHz CPU and 64G RAM). The programs implemented in Java are evaluated on two real datasets, Gowalla and Brightkite,[1] both of which include over 1 million location check-ins from mobile users.

[1] http://snap.stanford.edu/data/.

Table 3
Comparison of Setup Time (s) under the Setting of $K = \{512, 1024\}$.

$n(\times 10^3)$	SecVKQ				SVkNN			
	Gowalla		Brightkite		Gowalla		Brightkite	
	512	1024	512	1024	512	1024	512	1024
2	123.6	657.7	115.7	509.4	546.1	3,084	492.7	2,639
10	446.3	2,286	425.9	2,940	2,361	13,232	2,137	11,505
100	6,181	32,886	5,791	30,028	17,782	97,353	16,629	86,959

Table 4
Comparison of Communication Costs (MB) under the Setting of $K = \{512, 1024\}$.

$n(\times 10^3)$	SecVKQ				SVkNN			
	Gowalla		Brightkite		Gowalla		Brightkite	
	512	1024	512	1024	512	1024	512	1024
2	1.5	2.7	1.5	2.6	3.3	6.3	3.0	5.9
10	7.9	13.9	7.6	13.0	15.2	30.7	13.7	29.5
100	81.2	139.6	76.9	132.4	126.7	251.5	121.2	242.3

In the experiment, the dataset size n is set to [2000 ∼ 100,000]. The parameter k in the search phase and verification phase is set to [1 ∼ 20] and [1 ∼ 100], respectively. The size of the intermediate results is set to $\alpha = \{0.4k, 0.8k, 10k, 20k\}$. The key size of the condensed RSA encryption is set to 512 bits, and the key size K of the Paillier cryptosystem are chosen from $\{256, 512, 1024\}$. The PRF F is instantiated using HMAC-SHA-512. While implementing the SVkNN scheme, the grid granularity is set to 16 and the cryptographic hash functions are implemented via HMAC-SHA-512.

7.2. *Experiment results*

LBS provider. The execution time and communication cost in data preprocessing are shown in Tables 3 and 4, respectively. The computational cost includes two components: the cost of encrypting the Voronoi diagram and the cost of generating signatures. From the experiment results, SVkNN requires additional operations like grid partition, grid padding, and grid encryption, and thus performs worse in this stage.

Dual-cloud servers. As described in Section 6.1, our search time is influenced by parameters n, k, K, and α. Therefore, we conduct experiments under different parameter settings to show the effectiveness of SecVKQ. From Fig. 7, we know that the search time of SecVKQ increases with either k or K. The parameter α represents the ratio of the number of data points produced by the range query to the query parameter k. In the case of α less than 1, cloud servers need to search results according to the encrypted Voronoi diagram, thus it will take more search time than the case that α is more than 1. In the same situation (i.e., $\alpha < 1$ or $\alpha > 1$), the search time of SecVKQ also increases with the parameter α because the cloud servers need to perform more calculations related to homomorphic encryption. This result agrees with the theoretical analysis. The search time in the case of $\alpha = 0.4k$ and $\alpha = 0.8k$ (i.e., $|\mathcal{IR}| < k$) is larger than that in the case of $\alpha = 10k$ and $\alpha = 20k$ (i.e., $|\mathcal{IR}| \geq k$). Furthermore, the parameter α has a greater impact on the search time when $|\mathcal{IR}| \geq k$ compared to the case of $|\mathcal{IR}| < k$. Fig. 8 shows the comparison results between SecVKQ and SVkNN with a varied n. From the result, we know that the search time of SVkNN is linearly correlated with n, but our search time is insensitive to n. For example, in Gowalla, when $k = 10$, $\alpha = 0.8k$, and $K = 512$, the search time of SVkNN and SecVKQ grows from 77.02 s to 3600.96 s and from 4.14 s to 10.94 s, respectively, while n grows from 2000 to 100,000. The reason for this is that the preliminary screening phase enables the cloud servers to check a small set of data points instead of the whole dataset. Figs. 9 and 10 show that both schemes are impacted by parameter k and K, but SecVKQ is much more efficient than SVkNN even under the setting of a small value of α. For ease of comparison, Figs. 8–10 use log-scale y-coordinates.

Mobile users. To generate a search token, the mobile user encrypts the query point with the Paillier cryptosystem, which incurs about 9.65 ms. With the help of cloud servers, the mobile user can recover plaintext results without performing Paillier decryption. As shown in Fig. 11, the verification cost in both SecVKQ and SVkNN increases with k. Furthermore, the VO size increases with the size of the dataset, since a larger dataset generates a larger $\mathcal{VO}_{geo}$. The comparative results show that our verification process incurs more execution time and less communication cost compared to SVkNN. The reason is that the condensed RSA signature allows for short signature length, but requires additional costs for the compression of signatures.

8. Related work

8.1. *Secure kNN query*

Nowadays, to protect data security and query privacy, more and more works have been done focusing on the secure kNN queries. Wong et al. [18] proposed a novel encryption scheme Asymmetric-Scalar-Product-Preserving Encryption (ASPE) which enabled the comparisons to be performed based on ciphertexts. Since then, Wang et al. [19] and Hu et al. [20] proposed EF schemes which utilized the light-weighted OPE and DRE to support efficient kNN queries on encrypted data. However, the above schemes achieve only weak security, and suffer from various attacks, e.g., known-plaintext attacks and inference attacks.

For enhanced security, Yi et al. [40] exploited private information retrieval (PIR) to achieve privacy-preserving kNN queries over clear datasets. Elmehdwi et al. [26] proposed the classic dual-cloud model, and utilized the Paillier cryptosystem to achieve secure kNN queries. Inspired by their work, Kim et al. [27] improved query efficiency through building secure kd-tree indexes. The main drawback of these schemes is that it is hard from them to be applied to large-scale datasets due to the high computational overhead of the Paillier cryptosystem. Besides, Li et al. [41] proposed a secure nearest neighbor queries scheme and proved that it was secure against adaptive chosen keyword attacks. However, their scheme cannot support secure kNN queries. Lei et al. [42] designed a secure and efficient kNN queries scheme, which employed the projection function-based approach to code neighbor regions of a given location. Chen et al. [43] solved the problem of secure kNN queries by using lattice-based additively homomorphic encryption, distributed oblivious RAM and garbled circuits. Unfortunately, both schemes of Lei et al. and Chen et al. only achieve approximate kNN queries over encrypted data. Meanwhile, all the above schemes cannot support the verification of search results.

8.2. *Verifiable query*

Merkle Hash Tree (MHT) [44] and signature chain [45] are two main techniques in the verification of multi-dimensional range queries and skyline queries [46–49]. However, kNN queries need to compute the squared Euclidean distance between a query point and data point, and thus it is harder for mobile users to verify the correctness of search results.

Although a number of schemes [50–52] were proposed to verify queries over clear datasets, the research in verification of secure kNN queries is still in its infancy. Jiang et al. [53] proposed a verifiable dynamic searchable symmetric encryption scheme based on the accumulation tree. However the scheme only supports secure and verifiable boolean queries, rather than kNN queries. Rong et al. [29] were the first to solve the problem of secure and verifiable kNN queries. Unfortunately, their scheme lacked security due to the adoption of ASPE. Yang et al. [30] proposed a scheme to support verification of kNN queries on encrypted road networks. As the state-of-the-art work in this field, Cui et al. [28] proposed the SVkNN scheme, which utilized the Voronoi diagram for verification. However, the SVkNN scheme is based on the Paillier cryptosystem and its search complexity is linearly correlated to the size of datasets.

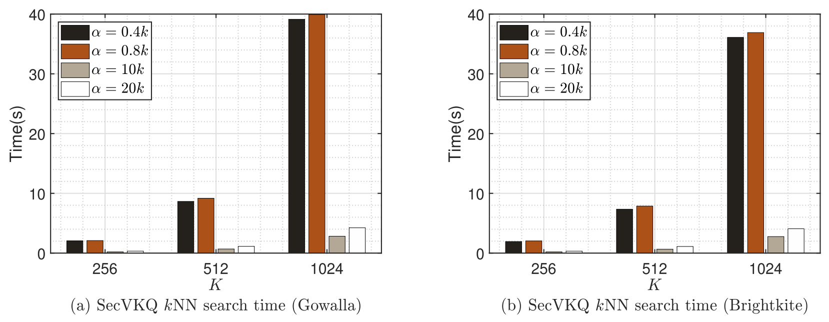


Fig. 7. The search time of SecVKQ with fixed $n = 20{,}000$ and $k = 10$ while K ranges from 256 to 1024.

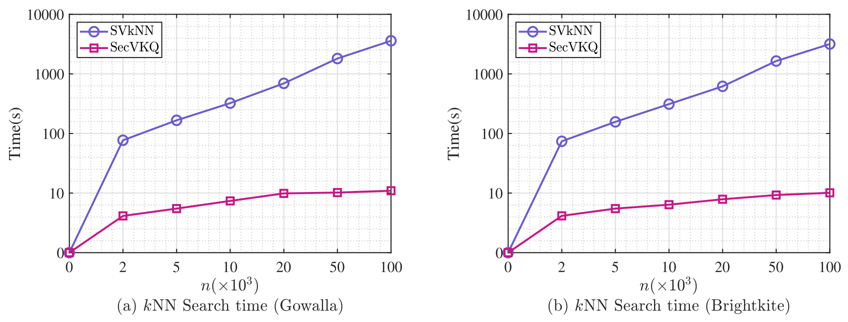


Fig. 8. Comparison of search time between SecVKQ and SVkNN with fixed $k = 10$, $\alpha = 0.8k$, and $K = 512$ while n ranges from 2000 to 100,000.

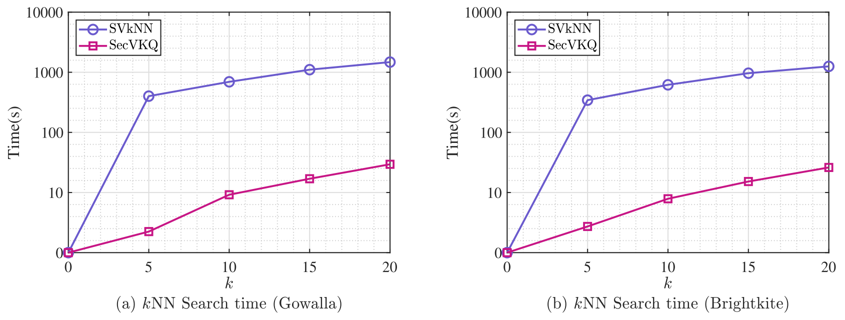


Fig. 9. Comparison of search time between SecVKQ and SVkNN with fixed $n = 20{,}000$, $\alpha = 0.8k$, and $K = 512$ while k ranges from 1 to 20.

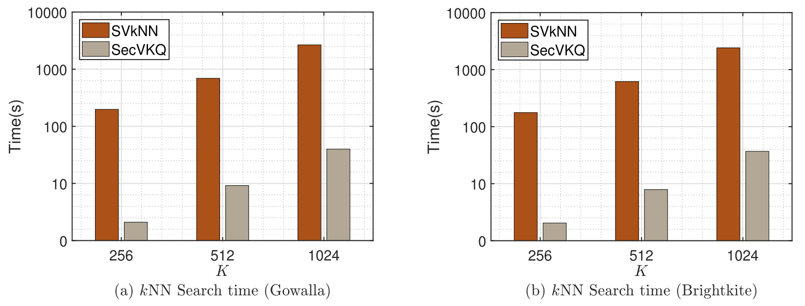


Fig. 10. Comparison of search time between SecVKQ and SVkNN with fixed $n = 20{,}000$, $\alpha = 0.8k$, and $k = 10$ while K ranges from 256 to 1024.

9. Conclusion

In this paper, we propose a SecVKQ framework to achieve secure and verifiable kNN queries in sensor–cloud systems. By utilizing secure data separation and adaptive encryption strategy, SecVKQ subtly integrates edge servers into the dual-cloud model so as to optimize query performance. Under SecVKQ, a series of secure protocols and a succinct verification strategy are proposed to obtain exact kNN results while preserving data privacy, index privacy, query privacy, and result correctness. Experiment results demonstrate that SecVKQ is efficient and practical on outsourced sensor datasets. However, it is hard for a Voronoi diagram to support queries of higher dimensional sensor data. As part of our future work, we will try to improve the SecVKQ framework by using flexible verifiable data structures.

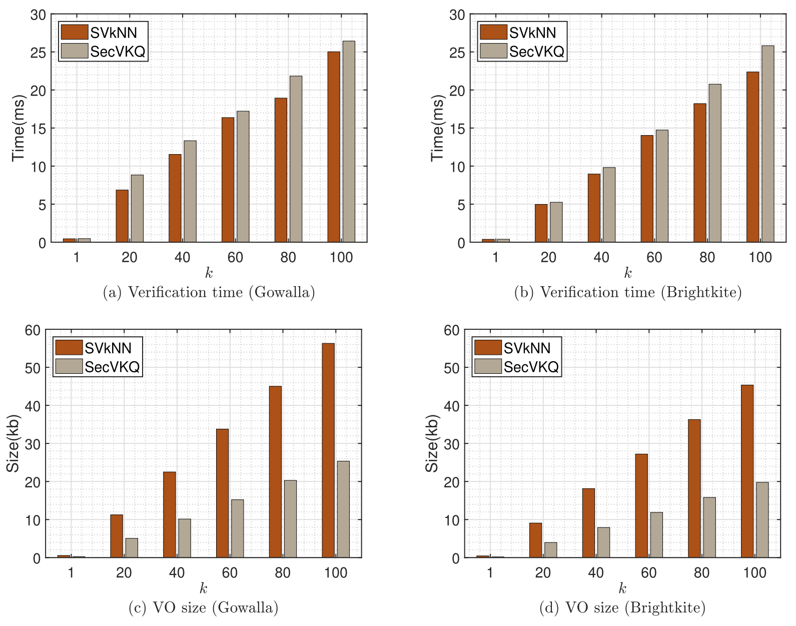


Fig. 11. Comparison of verification cost between SecVKQ and SVkNN on different datasets while k ranges from 1 to 100.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported in part by NSFC, China grants 61632009, 61872133, and 61802076; NSF, USA grants CNS 1824440, CNS 1828363, CNS 1757533, CNS 1629746, and CNS 1651947; the CERNET Innovation Project (NGII20190409); the Guangdong Provincial Natural Science Foundation, China (No. 2017A030308006), and the Hunan Provincial Natural Science Foundation of China (Grant No. 2020JJ3015).

References

[1] T. Wang, Q. Yang, X. Shen, T.R. Gadekallu, W. Wang, K. Dev, A privacy-enhanced retrieval technology for the cloud-assisted internet of things, IEEE Trans. Ind. Inf. (2021) http://dx.doi.org/10.1109/TII.2021.3103547.
[2] X. Wang, L.T. Yang, L. Song, H. Wang, L. Ren, M.J. Deen, A tensor-based multiattributes visual feature recognition method for industrial intelligence, IEEE Trans. Ind. Inf. 17 (3) (2021) 2231–2241.
[3] M. Alazab, S. Khan, S.S.R. Krishan, Q. Pham, M.P.K. Reddy, T.R. Gadekallu, A multidirectional LSTM model for predicting the stability of a smart grid, IEEE Access 8 (2020) 85454–85463.
[4] X. Xie, X. Yang, X. Wang, H. Jin, D. Wang, X. Ke, BFSI-B: An improved k-hop graph reachability queries for cyber–physical systems, Inf. Fusion 38 (2017) 35–42.
[5] S. Zhang, G. Wang, M.Z.A. Bhuiyan, Q. Liu, A dual privacy preserving scheme in continuous location-based services, IEEE Internet Things J. 5 (5) (2018) 4191–4200.
[6] Q. Liu, P. Hou, G. Wang, T. Peng, S. Zhang, Intelligent route planning on large road networks with efficiency and privacy, J. Parallel Distrib. Comput. 133 (2019) 93–106.
[7] T. Wang, Y. Liu, X. Zheng, H. Dai, W. Jia, M. Xie, Edge-based communication optimization for distributed federated learning, IEEE Trans. Netw. Sci. Eng. (2021) http://dx.doi.org/10.1109/TNSE.2021.3083263.
[8] T. Wang, Y. Lu, J. Wang, H. Dai, X. Zheng, W. Jia, EIHDP: edge-intelligent hierarchical dynamic pricing based on cloud–edge-client collaboration for IoT systems, IEEE Trans. Comput. 70 (8) (2021) 1285–1298.
[9] J. Luo, X. Deng, H. Zhang, H. Qi, QoE-Driven computation offloading for edge computing, J. Syst. Archit. 97 (2019) 34–39.
[10] K. Yu, L. Lin, M. Alazab, L. Tan, B. Gu, Deep learning-based traffic safety solution for a mixture of autonomous and manual vehicles in a 5G-enabled intelligent transportation system, IEEE Trans. Intell. Transp. Syst. 22 (7) (2021) 4337–4347.
[11] Q. Liu, Y. Peng, J. Wu, T. Wang, G. Wang, Secure multi-keyword fuzzy searches with enhanced service quality in cloud computing, IEEE Trans. Netw. Serv. Manag. 18 (2) (2021) 2046–2062.
[12] Q. Liu, Y. Peng, S. Pei, J. Wu, T. Peng, G. Wang, Prime inner product encoding for effective wildcard-based multi-keyword fuzzy search, IEEE Trans. Serv. Comput. (2020) http://dx.doi.org/10.1109/TSC.2020.3020688.
[13] X. Zheng, Y. Zhou, Y. Ye, F. Li, A cloud data deduplication scheme based on certificateless proxy re-encryption, J. Syst. Archit. 102 (2020) 101666.
[14] N. Eltayieb, R. Elhabob, A. Hassan, F. Li, An efficient attribute-based online/offline searchable encryption and its application in cloud-based reliable smart grid, J. Syst. Archit. 98 (2019) 165–172.
[15] L. Yuan, S. Zhang, G. Zhu, K. Alinani, Privacy-preserving mechanism for mixed data clustering with local differential privacy, Concurr. Comput.: Pract. Exper. (2021) http://dx.doi.org/10.1002/CPE.6503.
[16] Q. Liu, Y. Tian, J. Wu, T. Peng, G. Wang, Enabling verifiable and dynamic ranked search over outsourced data, IEEE Trans. Serv. Comput. (2019) http://dx.doi.org/10.1109/TSC.2019.2922177.
[17] Q. Liu, G. Wang, F. Li, S. Yang, J. Wu, Preserving privacy with probabilistic indistinguishability in weighted social networks, IEEE Trans. Parallel Distrib. Syst. 28 (5) (2017) 1417–1429.
[18] W. k. Wong, D. W.-l. Cheung, B. Kao, N. Mamoulis, Secure $k$nn computation on encrypted datanases, in: Proc. of SIGMOD, 2009, pp. 139–152.
[19] B. Wang, Y. Hou, M. Li, Practical and secure nearest neighbor search on encrypted large-scale data, in: Proc. of INFOCOM, 2016, pp. 1–9.
[20] H. Hu, J. Xu, C. Ren, B. Choi, Processing private queries over untrusted data cloud through privacy homomorphism, in: Proc. of ICDE, 2011, pp. 601–612.
[21] X. Wang, J. Ma, X. Liu, R. Deng, Y. Miao, D. Zhu, Z. Ma, Search me in the dark: privacy-preserving boolean range query over encrypted spatial data, in: Proc. of INFOCOM, 2020, pp. 2253–2262.
[22] X. Lei, G. Tu, A. Liu, T. Xie, Fast and secure kNN query processing in cloud computing, in: Proc. of CNS, 2020, pp. 1–9.
[23] R. Agrawal, J. Kiernan, R. Srikant, Y. Xu, Order preserving encryption for numerice data, in: Proc. of SIGMOD, 2004, pp. 563–574.
[24] M. Naveed, S. Kamara, C.V. Wright, Inference attacks on property-preserving encrypted databases, in: Proc. of CCS, 2015, pp. 644–655.
[25] B. Yao, F. Li, X. Xiao, Secure nearest neighbor revisited, in: Proc of the ICDE, 2013, pp. 733–744.
[26] Y. Elmehdwi, B.K. Samanthula, W. Jiang, Secure k-nearest neighbor query over encrypted data in outsourced environments, in: Proc of the ICDE, 2014, pp. 664–675.
[27] H.I. Kim, H.J. Kim, J.W. Chang, A secure $k$nn query processing algorithm using homomorphic encryption on outsourced database, Data Knowl. Eng. 123 (2019) 101602.
[28] N. Cui, X. Yang, B. Wang, J. Li, G. Wang, SVkNN: efficient secure and verifiable k-nearest neighbor query on the cloud platform*, in: Proc. of ICDE, 2020, pp. 253–264.
[29] H. Rong, H. Wang, J. Liu, W. Wu, M. Xian, Efficient integrity verification of secure outsourced $k$nn computation in cloud environments, in: Proc. of TrustCom, 2016, pp. 236–243.

[30] S. Yang, S. Tang, X. Zhang, Privacy-preserving k nearest neighbor query with authentication on road networks, J. Parallel Distrib. Comput. 134 (2019) 25–36.

[31] A. Okabe, B. Boots, K. Sugihara, S.N. Chiu, Spatial tessellations: concepts and applications of voronoi diagrams, College Math. J. (2001) http://dx.doi.org/10.2307/2687299.

[32] E. Mykletun, M. Narasimha, G. Tsudik, Authentication and integrity in outsourced databases, ACM Trans. Storage 2 (2) (2006) 107–138.

[33] P. Paillier, Public-key cryptosystems based on composite degree residuosity classes, in: Proc. of EUROCRYPT, 1999, pp. 223–238.

[34] S. Goldwasser, S. Micali, C. Rackoff, The knowledge complexity of interactive proof systems, SIAM J. Comput. 18 (1989) 186–208.

[35] K. Xue, S. Li, J. Hong, Y. Xue, N. Yu, P. Hong, Two-cloud secure database for numeric-related SQL range queries with privacy preserving, IEEE Trans. Inf. Forensics Secur. 12 (7) (2017) 1596–1608.

[36] Q. Liu, S. Wu, S. Pei, J. Wu, T. Peng, G. Wang, Secure and efficient multi-attribute range queryies based on comparable inner product encoding, in: Proc. of CNS, 2018, pp. 1–9.

[37] Y. Manolopoulos, A. Nanopoulos, A. Papadopoulos, Y. Theodoridis, R-Trees: Theory and Applications, Springer, 2005, http://dx.doi.org/10.1007/978-1-84628-293-5.

[38] A. Liu, K. Zhengy, L. Liz, G. Liu, L. Zhao, X. Zhou, Efficient secure similarity computation on encrypted trajectory data, in: Proc. of ICDE, 2015, pp. 66–77.

[39] A.C. Yao, How to generate and exchange secrets, in: Proc. of FOCS, 1986, pp. 162–167.

[40] X. Yi, R. Paulet, E. Bertino, V. Varadharajan, Practical k nearest neighbor queries with location privacy, in: Proc. of ICDE, 2014, pp. 640–651.

[41] R. Li, A.X. Liu, H. Xu, Y. Liu, Ieee transactions on dependable and secure computing, in: Adaptive Secure Nearest Neighbor Query Processing over Encrypted Data, 2020, http://dx.doi.org/10.1109/TDSC.2020.2998039.

[42] X. Lei, A.X. Liu, R. Li, G. Tu, Seceqp: a secure and efficient scheme for $k$nn query problem over encrypted geodata on cloud, in: Proc. of ICDE, 2019, pp. 662–673.

[43] H. Chen, I. Chillotti, Y. Dong, O. Poburinnaya, I. Razenshteyn, M.S. Riazi, SANNS: Scaling up secure approximate k-nearest neighbors search, in Proc. of USENIX Security, 2020, pp. 2111–2128.

[44] R.C. Merkle, A certified digital signature, in: Proc. of CRYPTO, 1989, pp. 218–238.

[45] H. Pang, A. Jain, K. Ramamritham, K. Tan, Verifying completeness of relational query results in data publishing, in: Proc. of SIGMOD, 2005, pp. 407–418.

[46] X. Zhu, J. Wu, W. Chang, M.Z. Alam Bhuiyan, R. Choo, F. Qi, Q. Liu, G. Wang, On authenticated skyline query processing over road networks, Concurr. Comput.: Pract. Exper. (2020) http://dx.doi.org/10.1002/CPE.5747.

[47] C. Xu, C. Zhang, J. Xu, vChain: enabling verifiable boolean range queries over blockchain databases in: Proc. of SIGMOD, 2019, pp. 141–158.

[48] S. Wu, Q. Li, G. Li, D. Yuan, X. Yuan, C. Wang, ServeDB: secure, verifiable, and efficient range queries on outsourced database, in: Proc. of ICDE, 2019, pp. 626–637.

[49] Y. Yang, S. Papadopoulos, D. Papadias, G. Kollios, Authenticated indexing for outsourced spatial databases, VLDB J. 18 (3) (2009) 631–648.

[50] S. Papadopoulos, L. Wang, Y. Yang, D. Papadias, P. Karras, Authenticated multistep nearest neighbor search, IEEE Trans. Knowl. Data Eng. 23 (5) (2011) 641–654.

[51] Y. Jing, L. Hu, W. Ku, C. Shahabi, Authentication of k nearest neighbor query on road networks, IEEE Trans. Knowl. Data Eng. 26 (6) (2014) 1494–1506.

[52] R. Zhang, J. Sun, Y. Zhang, C. Zhang, Secure spatial top-k query processing via untrusted location-based service providers, IEEE Trans. Dependable Secure Comput. 12 (1) (2015) 111–124.

[53] S. Jiang, X. Zhu, L. Guo, J. Liu, Publicly verifiable boolean query over outsourced encrypted data, IEEE Trans. Cloud Comput. 7 (3) (2019) 799–813.

Qin Liu received her B.Sc. in Computer Science in 2004 from Hunan Normal University, China, received her M.Sc. in Computer Science in 2007, and received her Ph.D. in Computer Science in 2012 from Central South University, China. She has been a Visiting Student at Temple University, USA. Her research interests include security and privacy issues in cloud computing. Now, she is an Associate Professor in the College of Computer Science and Electronic Engineering at Hunan University, China.

Zhengzheng Hao received his B.Sc. in Computer Science in 2019 from East China Jiaotong University, China. Currently, he is pursuing the M.Sc. degree in the College of Computer Science and Electronic Engineering at Hunan University, China. His research interests include security and privacy issues in cloud computing.

Yu Peng received his B.Sc. in Software Engineering in 2018 from China West Normal University, China. Currently, he is pursuing the Ph.D. degree in the College of Computer Science and Electronic Engineering at Hunan University, China. His research interests include security and privacy issues in cloud computing.

Hongbo Jiang received the Ph.D. degree from Case Western Reserve University, in 2008. After that, he joined the faculty of the Huazhong University of Science and Technology as a full professor and the dean of the Department of Communication Engineering. Now, he is a full professor with the College of Computer Science and Electronic Engineering, Hunan University. His research concerns computer networking, especially algorithms and protocols for wireless and mobile networks. He is serving as an editor for the IEEE/ACM Transactions on Networking, associate editor for the IEEE Transactions on Mobile Computing, and associate technical editor for the IEEE Communications Magazine.

Jie Wu is the Chair and a Laura H. Carnell Professor in the Department of Computer and Information Sciences at Temple University, Philadelphia, PA, USA. Prior to joining Temple University, he was a Program Director at the National Science Foundation and a Distinguished Professor at Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. Dr. Wu has regularly published in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE TRANSACTIONS ON SERVICE COMPUTING, and Journal of Parallel and Distributed Computing. Dr. Wu was general co-chair/chair for IEEE MASS 2006, IEEE IPDPS 2008, IEEE ICDCS 2013, and ACM MobiHoc 2014, as well as program co-chair for IEEE INFOCOM 2011 and CCF CNCC 2013. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a CCF Distinguished Speaker and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.

Tao Peng received the B.Sc. in Computer Science from Xiangtan University, China, in 2004, the M.Sc. in Circuits and Systems from Hunan Normal University, China, in 2007, and the Ph.D. in Computer Science from Central South University, China, in 2017. Now, she is an Associate Professor of School of Computer Science and Cyber Engineering, Guangzhou University, China. Her research interests include network and information security issues.

Guojun Wang received B.Sc. degree in Geophysics, M.Sc. degree in Computer Science, and Ph.D. degree in Computer Science, at Central South University, China, in 1992, 1996, 2002, respectively. He is a Pearl River Scholarship Distinguished Professor of Higher Education in Guangdong Province, a Doctoral Supervisor and Vice Dean of School of Computer Science and Cyber Engineering, Guangzhou University, China, and the Director of Institute of Computer Networks at Guangzhou University. He has been listed in Chinese Most Cited Researchers (Computer Science) by Elsevier in the past six consecutive years (2014–2019). His research interests include artificial intelligence, big data, cloud computing, Internet of Things (IoT), blockchain, trustworthy/dependable computing, network security, privacy preserving, recommendation systems, and smart cities. He is a Distinguished Member of CCF, a Member of IEEE, ACM and IEICE.

Shaobo Zhang received the B.Sc. and M.Sc. degree in computer science both from Hunan University of Science and Technology, Xiangtan, China, in 2003 and 2009 respectively, and the Ph.D. degree in computer science from Central South University, Changsha, China, in 2017. He is currently an Associate Professor at School of Computer Science and Engineering of the Hunan University of Science and Technology, China. His research interests include privacy and security issues in social networks and cloud computing.