Hindawi
Complexity
Volume 2017, Article ID 1490283, 14 pages
https://doi.org/10.1155/2017/1490283

Research Article

Efficient Privacy-Preserving Protocol for *k*-NN Search over Encrypted Data in Location-Based Service

Huijuan Lian,[1] Weidong Qiu,[1] Di Yan,[2] Zheng Huang,[1] and Jie Guo[1]

[1]*School of Cyber Security, Shanghai Jiao Tong University, Shanghai 200240, China*
[2]*Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China*

Correspondence should be addressed to Weidong Qiu; qiuwd@sjtu.edu.cn

Received 1 September 2017; Revised 13 November 2017; Accepted 20 November 2017; Published 20 December 2017

Academic Editor: Jia Wu

With the development of mobile communication technology, location-based services (LBS) are booming prosperously. Meanwhile privacy protection has become the main obstacle for the further development of LBS. The *k*-nearest neighbor (*k*-NN) search is one of the most common types of LBS. In this paper, we propose an efficient private circular query protocol (EPCQP) with high accuracy rate and low computation and communication cost. We adopt the Moore curve to convert two-dimensional spatial data into one-dimensional sequence and encrypt the points of interest (POIs) information with the Brakerski-Gentry-Vaikuntanathan homomorphic encryption scheme for privacy-preserving. The proposed scheme performs the secret circular shift of the encrypted POIs information to hide the location of the user without a trusted third party. To reduce the computation and communication cost, we dynamically divide the table of the POIs information according to the value of *k*. Experiments show that the proposed scheme provides high accuracy query results while maintaining low computation and communication cost.

1. Introduction

Nowadays, the location-based services are developing rapidly with the wide use of mobile Internet and smart mobile devices. Location-based services use mobile device to learn the current location with the help of built-in positioning devices and get the location information through the mobile network.

In a location-based service, a user obtains the query result by providing accurate locations to the service provider. The user privacy is probably obtained by the adversary through the conjunction of relevant background knowledge and the captured location information. How to utilize the service while protecting the location privacy of the user has become a research topic in location-based services in recent years.

The privacy data of location-based services includes three aspects: identity information, location information, and query content. The privacy of location refers to hiding the accurate location of the user. The privacy of query content refers to hiding the specific description of the request submitted by the user. When the query content is obtained, the characteristics and the behaviors of the user can be deduced.

In the LBS, the core of privacy-preserving is to cut off the relevance of the identity information, location information and query content. The query contents from the user are relevant to each other in the case of continuous querying. Therefore, it not only enhances the difficulty of privacy protection, but also increases the computation cost.

There are two communication modes in the privacy-preserving query in LBS: one is based on the trusted third party (TTP) and the other is without trusted third party (TTP-free). Although TTP-based solutions [1–7] are able to collect enough information to maximally meet the needs of privacy protection, there are two problems: (1) It is difficult to obtain the TTP that fill the bill. (2) Centralized attack on TTP makes it become the bottleneck in the query scheme. TTP-free-based solutions take advantage of the limited information to help maximize privacy protection. However, there

is a lack of methods which are outstanding in all three aspects of query accuracy, efficiency, and privacy-preserving.

Lien et al. [8] have proposed a private circular query protocol (PCQP) without a TTP, which uses Moore curve and Paillier cryptosystem to implement the protection of the location and query content. The scheme contains a large number of homomorphic additions and multiplications, and thus it requires higher computation and communication cost. Utsunomiya et al. [9] made some improvements on the basis of PCQP and proposed a lightweight private circular query protocol (LPCQP) with divided POI-table to effectively reduce the number of homomorphic additions and multiplications. The dividing of the POI-table is performed only once in the initialization process and the number of subtables observably influences the accuracy of the query. In some extreme cases, the scheme cannot return enough k POIs to the user. In addition, when the number of POIs in a subtable is much larger than k, the homomorphic additions and multiplications bring large number of unnecessary computation cost. LPCQP uses the homomorphic encryption scheme proposed by Smart and Vercauteren to ensure the security. The scheme needs a large size of public key and a large amount of computation.

Considering the advantages and disadvantages of the above two schemes, this paper proposes an efficient private circular query protocol (EPCQP) to mitigate the drawbacks of LPCQP without damaging the security. The proposed scheme utilizes the fully homomorphic encryption scheme to address the problem of secure querying over encrypted data in LBS. To omit the redundant homomorphic additions and multiplications, the proposed scheme dynamically divides the encrypted POI-table according to the query requirement of the user. The data security depends on the Brakerski-Gentry-Vaikuntanathan (BGV) homomorphic encryption scheme [10]. In addition, the user utilizes the circular shift and modulo operation to replace the real location which guarantees the location privacy-preserving.

The proposed scheme has the following advantages.

(1) Location Privacy and Data Privacy. The proposed scheme can defend the correlation attack, the background knowledge attack, the offline keyword guessing attack, the inference attack, the man-in-the-middle attack, and the link attack.

(2) Computation Efficiency. The computation cost is reduced by 99% or more when k is from 5 to 50 compared with that of PCQP. When k is smaller than 25, the computation cost of EPCQP is significantly lower than that of LPCQP.

(3) High Accuracy Rate. The accuracy rate of the proposed scheme is higher than 90% even if k is large for the uniform dataset, and it is higher than 84% when k is large for the real-world dataset.

The remainder of the paper is organized as follows. Section 2 introduces the relevant background knowledge. Section 3 describes the proposed protocol in detail and Section 4 discusses the performance of the proposed scheme based on various experiments. Related work is reviewed in Section 5 and the conclusions are drawn in Section 6.

2. Background

In this section, we review the main techniques which are utilized in the proposed protocol.

2.1. Moore Curve. Space-filing curves [11] represent a class of curves which traverse through all points in a two-dimensional region or more generally an n-dimensional hypercube, without crossing themselves. Hilbert [12] demonstrated the general geometrical generating procedure for constructing an entire class of space-filling curves in 1891. Hilbert curve has the capability of superior clustering and partially retaining the neighboring adjacency of the original data [13, 14]. Figure 1 illustrates the Hilbert curves of different orders. An N-th order Hilbert curve can pass through all cells in a $2^N \times 2^N$ square girds. The number on the corner of each cell denotes an index, called *H-value*, from the set $[0, 2^N - 1]$ when the curve traverses the cells.

Moore curve [15] is a variation of the Hilbert curve with end-point-connected property. Figure 2 illustrates the Moore curves of different orders. The POIs in a two-dimensional region can be stringed into a circular structure. Our scheme adopts Moore curve due to the circularly connected property.

2.2. Homomorphic Encryption. Without the trusted third party, our scheme adopts the homomorphic encryption to protect the data privacy. Due to the property of the homomorphic encryption showed in (1), the server can perform addition or multiplication of plaintexts without decryption. In (1), m_1, m_2, pk, and sk denote the two plaintexts, the public key, and the private key, respectively. The encryption and decryption are denoted as $\varepsilon_{\text{pk}}(m)$ and $D_{\text{sk}}(m)$. Here, $+_c$ and $\times_c$ denote the homomorphic addition and multiplication over ciphertexts.

$$\begin{aligned} D_{\text{sk}}\left(\varepsilon_{\text{pk}}(m_1) +_c \varepsilon_{\text{pk}}(m_2)\right) &= m_1 + m_2 \\ D_{\text{sk}}\left(\varepsilon_{\text{pk}}(m_1) \times_c \varepsilon_{\text{pk}}(m_2)\right) &= m_1 \times m_2. \end{aligned} \tag{1}$$

Homomorphic encryption [16] was proposed by Rivest et al. in 1978. Gentry [17] proposed the first fully homomorphic encryption scheme in 2009, which used bootstrapping to construct a fully homomorphic encryption scheme from a somewhat homomorphic encryption scheme. The scheme can be summarized as follows: FHE = SHE + Bootstrapping, and its security depends on certain worst-case problems over ideal lattices. In order to reduce the computational complexity of the decryption circuit, he designed a lattice-based decryption circuit, and its security depends on the assumed hardness of two problems: sparse subset sum problem (SSSP) and the certain worst-case problems over ideal lattices. Gentry's scheme uses matrix operations and vector modular arithmetic, resulting in discontinuities in computation and fast growth in computation cost. In addition, the size of the ciphertext generated by the corresponding 1-bit plaintext is exponentially increasing; thus the size is becoming too long to be realized by programming. Despite all of these shortcomings, Gentry still makes a great contribution to the study of the fully homomorphic encryption. To improve the

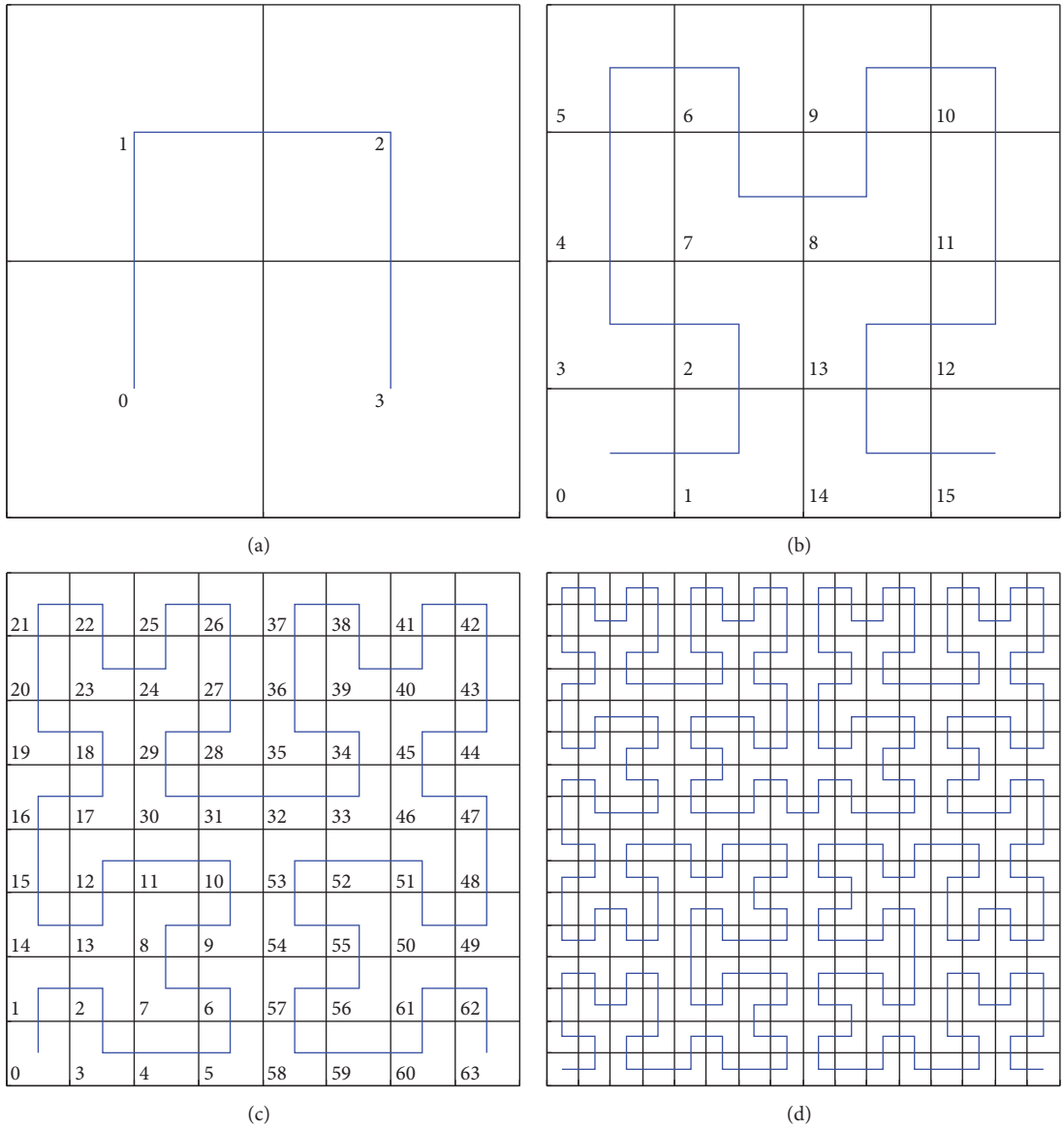

Figure 1: (a) Hilbert curve of order $N = 1$. (b) Hilbert curve of order $N = 2$. (c) Hilbert curve of order $N = 3$. (d) Hilbert curve of order $N = 4$.

poor practicality of Gentry's fully homomorphic encryption scheme, many optimization schemes of the fully homomorphic encryption have emerged since 2009. In 2010, van Dijk et al. [18] applied simple algebraic structure to construct a fully homomorphic encryption scheme, which is based on integer arithmetic, on the basis of Gentry's scheme. This scheme is simpler, but not practical.

In 2010, Smart and Vercauteren [19] proposed a fully homomorphic encryption scheme with smaller key and ciphertext size. In 2011, Gentry and Halevi [20] improved the original fully homomorphic encryption scheme with a new key generation algorithm, and the full polynomial inversion is not required in this scheme. In 2011, Gentry and Halevi [21] put forward some schemes which did not require squashing step and the hardness of SSSP assumption to further optimize the performance, and the practicability was improved. In 2012, Brakershi et al. [10] proposed the Brakerski-Gentry-Vaikuntanathan homomorphic encryption scheme which applied the key switching and modulus switching technique, meanwhile with additional BitDecomp technology to manage the noise. In the same year, Coron et al. [22] proposed a fully homomorphic encryption scheme over the integers which reduces the public key size of the van Dijk et al. scheme.

Most of the recent research results are still achieved by improving the original Gentry's scheme. Halevi and Shoup developed a new fully homomorphic encryption library, namely, HELib [23] in 2013, which is based on the BGV homomorphic encryption scheme, using modulus switching technique to reduce the ciphertext noise. It accomplishes the homomorphic operation of subtraction and shift, on the basis of the original additions and multiplications. In addition to

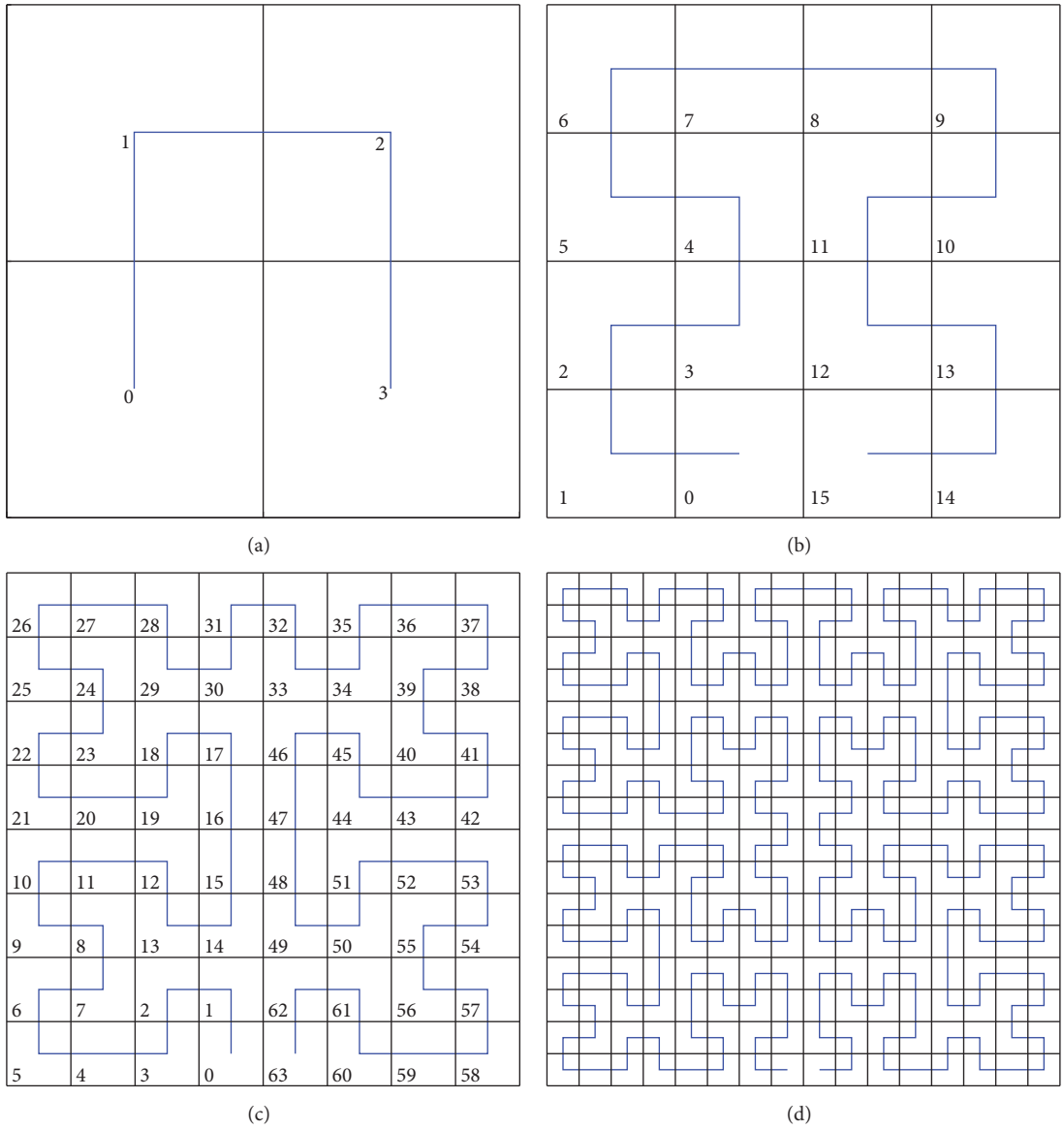

Figure 2: (a) Moore curve of order $N = 1$. (b) Moore curve of order $N = 2$. (c) Moore curve of order $N = 3$. (d) Moore curve of order $N = 4$.

the functional improvement, the performance of HELib is mainly optimized by Smart-Vercauteren ciphertext packing technique [24] and Gentry-Halevi optimization [25] to further improve the efficiency of homomorphic operations. In general, the homomorphic operation of encrypted data is accomplished and it shows better performance.

Taking everything above into consideration, we adopt the homomorphic encryption scheme released in HELib in the proposed protocol.

3. The Proposed Protocol

An efficient private circular query protocol is proposed which can achieve high accuracy and low computation cost without the trusted third party.

3.1. Overview of EPCQP. Figure 3 illustrates the whole architecture of EPCQP.

3.1.1. Initialization Process

Step 1. The server constructs a Moore curve and generates the *H-index* for every registered POI on the target region.

In this step, an LBS server selects the appropriate parameters to construct a Moore curve that covers up the target region and builds the POI-table containing the information of all registered POIs. The POI-table contains the *H-index* and POI-info of each POI, for example, the longitude, the latitude, and the name. Each stored POI in the POI-table is numbered in accordance with the evenly distributed *H-index*

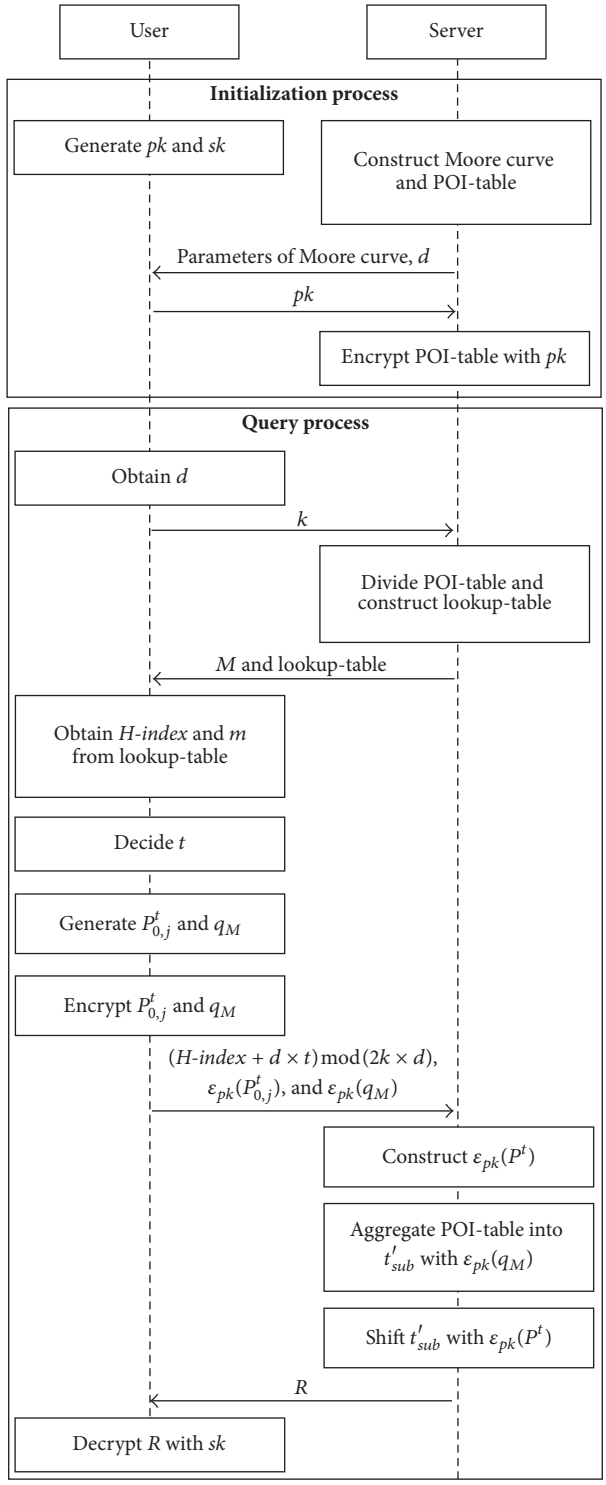

Figure 3: Interaction diagram of EPCQP.

with common difference d instead of the associated *H-value*. The definition of *H-index* will be presented in Section 3.2.1.

Step 2. The user generates the public and private key pairs and sends the public key pk to the server. Then the server encrypts the POI-table with the public key pk.

3.1.2. Query Process

Step 3. The user issues a k-NN query to the server.

Step 4. The server divides the POI-table into M subtables. Note that the *H-index* column of the POI-table will not be encrypted. The number of all entries in each subtable is $2k$. $\mathrm{ID}_{\text{sub-table}}$ denotes an index of the subtable in which a POI is contained. The sever stores the mapping from *H-value* to *H-index* of each POI into a lookup-table. In addition, the lookup-table records the $\mathrm{ID}_{\text{sub-table}}$ in order that the user knows which subtable he should search. Consequently, the lookup-table contains three attributes of each POI: *H-index*, *H-value*, and $\mathrm{ID}_{\text{sub-table}}$. The details of the dividing method will be presented in Section 3.2.2.

Step 5. The sever announces the setting parameters of the Moore curve and the lookup-table to all registered users in public. The user can retrieve the *H-index* of his current location on the target region and the subtable that he should search. The details will be presented in Section 3.2.3.

Step 6. The user chooses an integer t to generate a $2k \times 2k$, t-offset circular shift permutation matrix $P^t_{i,j}$ $(i, j = 0, \ldots, 2k - 1)$ and a vector q_M. In the first row of $P^t_{i,j}$, the $(2k - t + 1)$-th element is the only nonzero element. The m-th element of q_M is the only nonzero element. The definitions of $P^t_{i,j}$ and q_M will be presented in Sections 3.2.3 and 3.2.4.

Step 7. The user calculates the value $(H\text{-}index + d \times t)$ modulo $(2k \times d)$ to generate the shift-H-index. The *H-index* is retrieved from the lookup-table according to the current location. The user encrypts q_M and the first row of $P^t_{i,j}$ by the public key pk generated in Step 2, denoted as $\varepsilon_{\mathrm{pk}}(q_M)$ and $\varepsilon_{\mathrm{pk}}(P^t_{0,j})$. Then the user sends the shift-H-index, $\varepsilon_{\mathrm{pk}}(P^t_{0,j})$, and $\varepsilon_{\mathrm{pk}}(q_M)$ to the server.

Step 8. The server utilizes $\varepsilon_{\mathrm{pk}}(P^t_{0,j})$ to construct a secret circular shift matrix $\varepsilon_{\mathrm{pk}}(P^t)$. The server aggregates all of the subtables with $\varepsilon_{\mathrm{pk}}(q_M)$ into a new table t'_{sub}. The *H-indexes* of the POIs in t'_{sub} are numbered again from d to $(2k \times d)$.

Step 9. The server utilizes $\varepsilon_{\mathrm{pk}}(P^t)$ to perform a secret circular shift on t'_{sub} based on the fully homomorphic property with the public key of the user. The server performs a k-NN search upon the circularly shifted t'_{sub} and then returns the k encrypted results to the user. The detail will be present in Section 3.2.4.

Step 10. The user decrypts the received results with the private key sk selected in Step 2 and obtains the required k-NN POIs.

The *H-index* of the user has been added by t (in Step 7), and the POIs in t'_{sub} have been secretly circularly shifted by $(d \times t)$ (in Step 9). Based on the additive and multiplicative homomorphism, the secret k-NN search results in the shifted

t'_{sub} will be consistent with the results searched in their plaintexts of the original subtable.

3.2. Details of EPCQP

3.2.1. Mapping from H-Value to H-Index. Figure 1 shows that the starting cell is not adjacent to the ending cell in the Hilbert curve. The searching range will be reduced when the user is near to the starting or ending cell of Hilbert curve, and it means that the query accuracy will be reduced. Figure 2 indicates that the start point and the end point of Moore curve are neighbors. Therefore, we adopt Moore curve to transform a two-dimensional space into a sequence of *H-values*. With the capability of Moore curve, all the POIs can be constructed into a circular structure which is important when dividing the POI-table in the subsequent steps. In addition, the results mainly rely on the order of *H-values*, so that altering *H-values* of POIs do not affect the query results.

H-index denotes an evenly distributed sequence numbered in the ascending order of *H-values* with a common difference d. *H-index* of the i-th POI in a Moore curve is calculated as

$$H\text{-}index\,(i) = d \times i, \tag{2}$$

where d is an integer greater than or equal to one and i is the sequencing-order of the POI along with the ascending order of *H-values* in the given Moore curve. The server constructs a POI-table which contains the POI-info and *H-index* of all POIs and records the mapping from *H-value* to *H-index* in the lookup-table. The server should update the POI-table and lookup-table whenever any POI changes and then publicly announces the new lookup-table to all registered users.

3.2.2. Dividing the POI-Table. PCQP requires a huge number of calculations due to the multiplication applied across the entire POI-table. Utsunomiya et al. proposed a lightweight k-NN search protocol according to dividing the POI-table into M subtables. The server divides a POI-table in the initialization process in advance in LPCQP. When the user issues a k-NN query, he selects only one subtable which contains POIs that the user needs. Although LPCQP reduces computation cost observably compared with PCQP, it requires a trade-off between the query accuracy and computation cost on the server. Utsunomiya et al. have illustrated that k/n_p should be 0.5 or less in order to obtain the high query accuracy, where n_p denotes the number of all entries in a subtable. The query accuracy of LPCQP becomes worse as M get larger due to the loss of some POIs from aggregating subtables. Extremely, when $k > n_p$, the server only returns n_p POIs to the user. The dividing of the POI-table is performed only once in advance; therefore, it cannot satisfy the query requirement if M is not appropriate.

Inspired by LPCQP, we propose an efficient k-NN search scheme to mitigate the drawbacks of LPCQP, called EPCQP. In EPCQP, the size of each subtable depends on the query requirement of the user instead of dividing the POI-table only once in the initialization process. We divide the POI-table into M subtables, and each subtable has $2k$ POIs. M is calculated as

$$M = \left\lceil \frac{n}{2k} \right\rceil, \tag{3}$$

where n denotes the number of all entries in the POI-table. The j-th subtable, denoted as t^j_{sub}, is defined as

$$\begin{aligned} t^j_{\text{sub}} &= \left[I^j_i\right]_{1\le i\le 2k} \\ I^j_i &= I_{2k\times(j-1)+i}, \end{aligned} \tag{4}$$

where I_i denotes the i-th entry of the original POI-table and I^j_i denotes the i-th entry of the j-th subtable.

As mentioned in Section 3.2.1, the start point is adjacent to the end point in Moore curve. Geographically, the first POI and the last POI stored in the POI-table are close to each other. Therefore, the first and the last POIs are neighbors in two-dimensional space regardless of the *H-index* distance between them. If the number of all entries in the M-th subtable is less than $2k$, it will be appended using the entries in the original POI-table from the first to the l_p-th in order. l_p is calculated as

$$l_p = 2k \times \left\lceil \frac{n}{2k} \right\rceil - n. \tag{5}$$

As shown in Figure 4, there are nine POIs in the POI-table. If $k = 2$, the POI-table will be divided into three subtables each containing four POIs. The third subtable contains i, a, b, and c.

3.2.3. Aggregating Subtables. The user looks for the *H-index* from the lookup-table according to the current location and then chooses the subtable which contains the nearest POI as the subtable for querying. Next, the user obtains the index of the target subtable by retrieving the $\text{ID}_{\text{sub-table}}$ column of the lookup-table. Without loss of generality, let the m-th subtable be the one that the user selects. The user generates a vector q_M defined as

$$\begin{aligned} q_M &= \left|q_i\right|_{1\le i\le M} \\ q_i &= \begin{cases} 1, & i = m \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{6}$$

The user encrypts q_M with the public key pk and then sends the ciphertext $\varepsilon_{\text{pk}}(q_M)$ to the server. The server multiplies each element of $\varepsilon_{\text{pk}}(q_M)$ by the corresponding subtable. Then the j-th subtable t^j_{sub} becomes t'^j_{sub}.

$$\begin{aligned} t'^j_{\text{sub}} &= \varepsilon_{\text{pk}}\left(q_j\right) \times_c \varepsilon_{\text{pk}}\left(t^j_{\text{sub}}\right) \\ &= \left[\varepsilon_{\text{pk}}\left(q_j\right) \times_c \varepsilon_{\text{pk}}\left(I^j_i\right)\right]_{1\le i\le 2k}. \end{aligned} \tag{7}$$

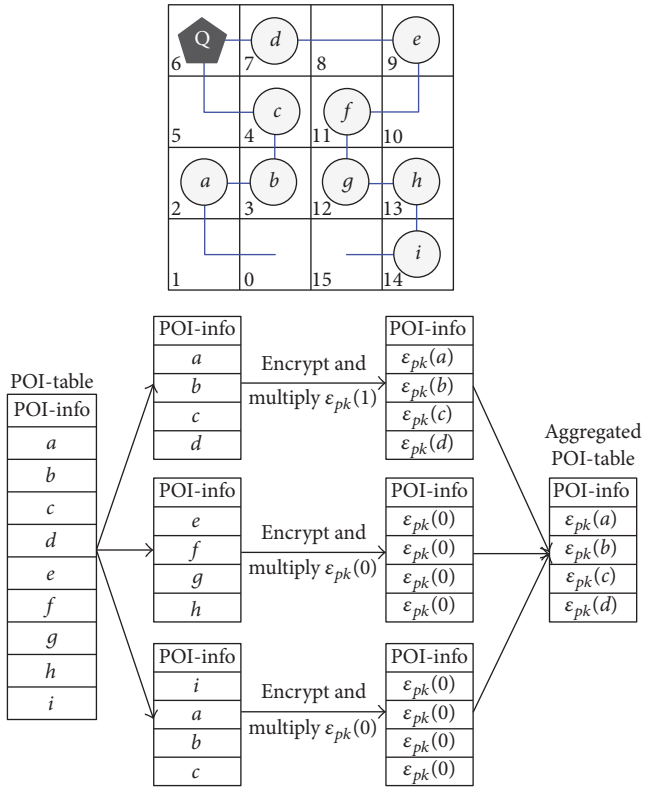

Figure 4: Example of aggregating subtables.

The server aggregates all the subtables into a new table t'_{sub} calculated as

$$t'_{\text{sub}} = t'^{1}_{\text{sub}} +_c t'^{2}_{\text{sub}} +_c \cdots +_c t'^{M}_{\text{sub}} = \left[I'_i\right]_{1\le i\le 2k}$$

$$I'_i = \varepsilon_{\text{pk}}(q_1) \times_c \varepsilon_{\text{pk}}\left(I_i^1\right) +_c \varepsilon_{\text{pk}}(q_2) \times_c \varepsilon_{\text{pk}}\left(I_i^2\right) +_c \cdots +_c \varepsilon_{\text{pk}}(q_M) \times_c \varepsilon_{\text{pk}}\left(I_i^M\right). \quad (8)$$

According to (6) and (8), all entries of the POI-info in the subtables become zero in their plaintext domain except the m-th subtable. Note that the H-*indexes* of the POIs in t'_{sub} are numbered from d to $(2k \times d)$.

Due to the properties of homomorphic encryption, t'_{sub} satisfies

$$D_{\text{sk}}\left(I'_i\right) = I_i^m. \quad (9)$$

Figure 4 indicates the process of aggregating subtables. There are nine POIs and a user (Q) on the map. According to the Moore curve, the nine POIs are stored in the POI-table in the ascending order of H-*indexes*. When the user issues a k-NN search with $k = 2$, the server divides the POI-table into three subtables and each subtable has four POIs. The POI d has the nearest H-*index* to the user; therefore, the user selects the first subtable to query. Then all the subtables are aggregated into one table with the entries of the first subtable.

3.2.4. Secret Circular Shift in the m-th Subtable. In order to keep the original H-*index* secret to the server, the POI-info column of the original POI-table is circularly shifted in PCQP. On the basis of Paillier encryption scheme, an approach for circular shift by the encrypted matrix-vector multiplication is proposed in PCQP, where the POI-info column is in its plaintext domain whereas the permutation matrix is encrypted. Although the circular shift in the entire POI-table maintains the neighboring relationship with POIs, it requires a huge number of calculations due to the multiplication applied across the entire POI-table. Utsunomiya et al. proposed a lightweight scheme (LPCQP) to mitigate the drawbacks of PCQP. Inspired by LPCQP, we utilize the circular shift in t'_{sub} and modulo operation to hide the real location of the user.

After shifting the POI-info column t units downward circularly, the same k-NN search results can be obtained by changing the querying H-*index* to the shift-H-index, which is calculated as

$$\text{shifted-}H\text{-index} = (H\text{-}index + d \times t) \bmod (2k \times d). \quad (10)$$

When t is a negative integer, it represents an upward shifting.

The shift parameter t is decided by the user; hence the server cannot obtain the original H-*index* of the user according to the shifted-H-index. The current location of the user is protected from being disclosed.

Likewise, the circularly shift on the POI-info column on the server side should be kept secretly. In PCQP, the entire POI-info column has been circularly shifted by multiplying an encrypted $n \times n$ matrix. The operator of multiplicative and additive homomorphism will incur a huge overhead, especially for the mobile services. In this paper, only the POI-info of the m-th subtable which the user selects will be circularly shifted by multiplying an encrypted $2k \times 2k$, t-offset matrix $P^t_{i,j}$, which is defined as

$$P^t_{i,j} = \begin{cases} 1, & j = i + (2k - t) \bmod (2k) \\ 0, & \text{otherwise,} \end{cases} \quad (11)$$

where $i, j = 0, 1, \ldots, 2k - 1$.

The user encrypts the permutation matrix $P^t_{i,j}$ with the public key pk to obtain the ciphertext $\varepsilon_{\text{pk}}(P^t)$ and then sends it to the server. The server multiplies $\varepsilon_{\text{pk}}(P^t)$ with the aggregated table t'_{sub} in order to circularly shift the POI-info column of the m-th subtable and keeps the H-*index* column intact.

Note that the pseudorandom numbers are different during each encryption process, it means that the server has no way to distinguish the encrypted '0' and '1'. In order to reduce the computation and communication cost, the user only encrypts the first row of $P^t_{i,j}$ and sends $\varepsilon_{\text{pk}}(P^t_{0,j})$ to the server. Then sever constructs $\varepsilon_{\text{pk}}(P^t)$ by circularly shift on $\varepsilon_{\text{pk}}(P^t_{0,j})$. For ease of understanding, there is an example on how to construct $\varepsilon_{\text{pk}}(P^t)$.

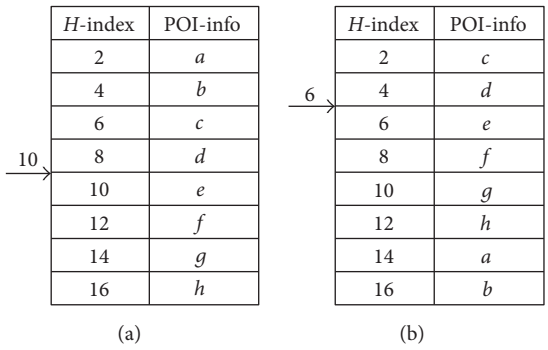

Figure 5: (a) The evenly distributed POIs. (b) After circularly shifting, the shift-H-index = $(10 + 2 \times 6) \bmod 16 = 6$.

Let $(\varepsilon_{pk}(P_{00}^t)\ \ \varepsilon_{pk}(P_{01}^t)\ \ \varepsilon_{pk}(P_{02}^t)\ \ \varepsilon_{pk}(P_{03}^t))$ be the first row of $\varepsilon_{pk}(P^t)$, and $\varepsilon_{pk}(P^t)$ can be constructed as

$$\begin{pmatrix} \varepsilon_{pk}(P_{00}^t) & \varepsilon_{pk}(P_{01}^t) & \varepsilon_{pk}(P_{02}^t) & \varepsilon_{pk}(P_{03}^t) \\ \varepsilon_{pk}(P_{03}^t) & \varepsilon_{pk}(P_{00}^t) & \varepsilon_{pk}(P_{01}^t) & \varepsilon_{pk}(P_{02}^t) \\ \varepsilon_{pk}(P_{02}^t) & \varepsilon_{pk}(P_{03}^t) & \varepsilon_{pk}(P_{00}^t) & \varepsilon_{pk}(P_{01}^t) \\ \varepsilon_{pk}(P_{01}^t) & \varepsilon_{pk}(P_{02}^t) & \varepsilon_{pk}(P_{03}^t) & \varepsilon_{pk}(P_{00}^t) \end{pmatrix}. \quad (12)$$

The server multiplies $\varepsilon_{pk}(P^t)$ with the aggregated table t'_{sub} to obtain the circularly shifted and decrypted POI-info data. After decryption, the results will have the same shifted value t.

As shown in Figure 5(a), there are 8 POIs in the subtable, where $k = 4$ and $d = 2$. The *H-index* of the user is 10, and the results will be c, d, e, and f. The shift-H-index is 6 when $t = 6$. As presented in Figure 5(b), the user will obtain the same results according to the shift-H-index.

Note that the first entry and the last entry of the subtable are not adjacent to each other. Therefore, the neighboring relationship will be changed after circularly shifting the m-th subtable, and this will reduce the search accuracy. This effect is negligible compared with the reduction in computation and communication overhead.

3.2.5. k-NN Search. Lien et al. [8] proposed a cross-like k-NN search algorithm to achieve high accuracy rate and showed that two additional queries started from the central cells of search region are sufficient to achieve the reasonable accuracy rate in most cases. However, it may include duplicated POI especially when the user is close to the boundary. Utsunomiya et al. [9] proposed a group-based query point selection algorithm which achieved a higher accuracy rate. We adopt the methods proposed in LPCQP to improve the accuracy rate.

3.3. Security Analysis. Mainly the most common attacks that can obtain some private information of the user can be listed as the following 6 chances. We divide them into three groups where there are two of them in each group.

The first group consists of correlation attack and background knowledge attack. The adversary utilizes the former one to eavesdrop some input queries and output results through the network. Then combining with some prior knowledge obtained from the latter one about the basic information of the user such as age or job, the adversary can infer the location of the user with a relatively large probability.

The second group includes offline keyword guessing attack as well as inference attack. There is a trapdoor generated from a search word and it does not leak any information of this search word. The first attack guesses the content of the encrypted data by computing trapdoors of some widely used words. Simultaneously, the second attack combines some background knowledge of the data content with some access patterns to identify the trapdoor of some words.

The last group is divided into man-in-the-middle attack (MITM) and link attack. For MITM, the adversary needs to be a third party between users and servers. It needs to guarantee that both two parties believe they are having conversations directly with the other one. When it comes to the link attack which is more commonly used, the adversary combines the inaccurate location information of the user and the data source from the outside to determine the accurate location or the identification of the user.

In this paper, we are aiming at proposing an efficient scheme for k-NN search with perfect privacy-preserving. In the following part, we will illustrate the security analysis of six different kinds of attacks that are mentioned previously.

3.3.1. The Correlation Attack. Correlation attack belongs to the plaintext attacks that utilize a statistical weakness due to a poor choice of the Boolean function.

The correlation attacks can be successfully mounted due to the fact that obvious correlations between the output of a special linear feedback shift register (LFSR) and the outputs of all the LFSRs defined by Boolean functions can be distinguished. Thus combining with part of the keystream knowledge, an adversary can get the key of the special LFSR by brute-forcing.

As for the correlation attack in EPCQP, the user sends $\varepsilon_{pk}(P_{0,j}^t)$ and $\varepsilon_{pk}(q_M)$ which are all encrypted with distinct random numbers, with these random numbers being changed every time. Therefore, the server will have no idea of correlating queries issued by the user. It implies that our scheme is secure under the correlation attack.

3.3.2. The Background Knowledge Attack. This kind of attack exploits the close relationship between several standard common attributes where the sensitive attribute is exactly among them. In this way, the adversary can reduce the cardinality of the possible value set to find this sensitive attribute.

It can be seen that only the user knows the shifts amount because it is encrypted in our scheme. As for the server, it can learn nothing about the location though it can get the query history and the profiles of the user. This arises from the fact that the user can change the shifts amount every time while all the sensitive information transferred is encrypted.

In terms of this attack in our scheme, we only transfer the shifted location chosen by the user during the k-NN search. The shifted location $(H\text{-}index + d \times t) \bmod (2k \times d)$

is not sensitive because the shifts amount is selected by the user independently and randomly. Therefore, no sensitive information about the user will be leaked to the server, and thus our scheme is resistant to this attack.

3.3.3. The Offline Keyword Guessing Attack. Generally speaking, dictionary attack as well as offline guessing attack occurs based on the fact that the so-called weak secrets may have low entropy. This means that it comes from the value set of a small cardinality. Similarly, the keywords also come from a rather smaller set when compared with the weak secrets such as passwords. Basically, low entropy means high probability, so the user is more likely to use the low entropy keywords when querying the table.

In our scheme, as for the offline keyword guessing attack, taking the location of the user, for example, the user rarely uses the low entropy keywords. Particularly, the location data in the server is collected as a lookup-table which is not the sensitive information of the user. Accordingly, by offline keyword guessing attack, no one can guess location data of the user during the query in our scheme.

3.3.4. The Inference Attack. The inference attack aims at gaining knowledge of the user by analyzing the data. If the adversary can obtain the real value of the sensitive information with a relatively high probability, we say the user leaks the information. In the whole process, the adversary cannot directly obtain any data from some trivial information.

When it comes to the inference attack in our scheme, it utilizes the access pattern of the user, such as a document that contains previously queried keywords. However, no sensitive information about the access pattern of the user will be leaked to the server or an adversary because $P^t_{i,j}$ and q_M are encrypted with distinct random numbers, and these random numbers are scrambled by the user each time when the user queries. Consequently, our scheme is robust to this attack.

3.3.5. The Man-in-the-Middle Attack. The man-in-the-middle attack is a widely used attack where there exists a third party called man-in-the-middle. It plays the role of a user when communicating with the server and then it will play the role of a server when facing the user. So it needs to imitate all the information transmitted in the network to make the user and the server believe that they are truly having conversation with the other one.

There are two widely used methods in preventing man-in-the-middle attack. Authentication is used to make sure that the information transmitted during the communication is from a legitimate source. Tamper detection ensures that the information is not tampered during the transmission.

In our scheme, the user will never send any sensitive information or plaintext to the server, nor will the server do. The user only sends k, pk, $(H\text{-}index + d \times t) \bmod (2k \times d)$, $\varepsilon_{\text{pk}}(P^t_{0,j})$, and $\varepsilon_{\text{pk}}(q_M)$. As for the server, it only needs to send the parameters of Moore curve, lookup-table, d, M, and the final result set which is encrypted by the homomorphic encryption scheme. Totally, they are encrypted with distinct random numbers and the numbers are varied every time when the user queries. Meanwhile the information of m which is the location of the subtables is transmitted in the form of $\varepsilon_{\text{pk}}(q_M)$. Thus, the adversary cannot play the role of the server, because he has no idea of m and cannot communicate with the user. In sum, our scheme can defend against this attack.

3.3.6. The Link Attack. When it comes to the link attack which is more commonly used, the adversary combines the inaccurate location information of the user and the data source from the outside to determine the accurate location or identification of the user. However, all of the location information is represented as $(H\text{-}index + d \times t) \bmod (2k \times d)$ in the aggregated subtable. The server can learn nothing from the shift amount, though it may have access to the partial knowledge of the data source from the outside. Therefore, our scheme can also guarantee the security under link attack.

4. Performance Evaluation

In this section, we compare the performance of EPCQP with that of the related two works: PCQP [8] and LPCQP [9]. We adopt the homomorphic encryption scheme released in HELib. The proposed scheme is implemented in JAVA language and performed on a laptop computer with a 1.6 GHz Intel Core i5 CPU and 4 GB RAM.

We use two datasets including a uniform dataset and a real-world dataset. Each dataset contains 10000 POIs. The real-world dataset is extracted from the base stations datasets in China. We randomly select 1000 locations on the map to issue the k-NN search in each experiment and the results are averaged.

4.1. Query Accuracy. The accuracy criteria have been well used in data mining and machine learning areas [26–30]. Similarly, query accuracy is used for validation of the experiment in this paper. The value of k and M is varied from 5 to 50 and from 10 to 500, respectively. Let R and G denote the returned result set and the k-NN ground-truth result set, and the accuracy rate is defined as

$$r_{\text{acc}} = \frac{|R \cap G|}{|G|}. \tag{13}$$

In LPCQP, only the m-th subtable is selected by the user for searching. The number of the subtables will directly affect the query accuracy. When k is greater than n_p, the server returns less than k POIs to the user. Although the computation cost is reduced by dividing the POI-table only once in the initialization process, the query accuracy rate decreases in some cases. After dividing the POI-table into M subtables, $(n - n_p)$ POIs are lost due to the aggregating subtables in (8). Hence, the quality of the results gets even worse as the number of the subtables gets larger. In contrast, the server can construct the subtables with a large unit when M is small, which guarantees that the number of results satisfies the querying requirement. Nevertheless, when n_p is far greater than k, as mentioned in (7) and (8), the computation cost incurred by the homomorphic additions

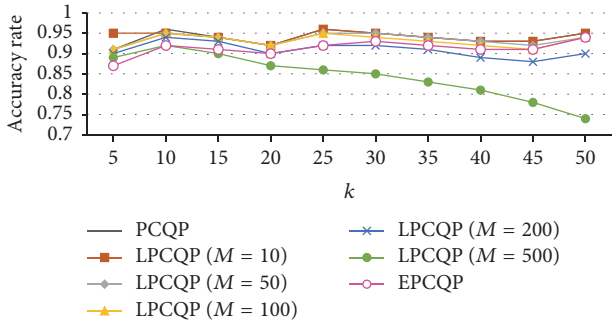

Figure 6: Query accuracy rate versus k of the uniform dataset.

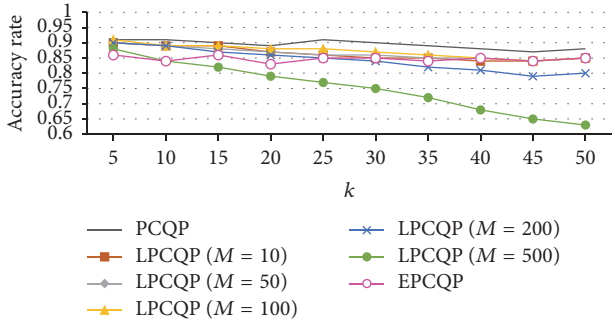

Figure 7: Query accuracy rate versus k of the real-world dataset.

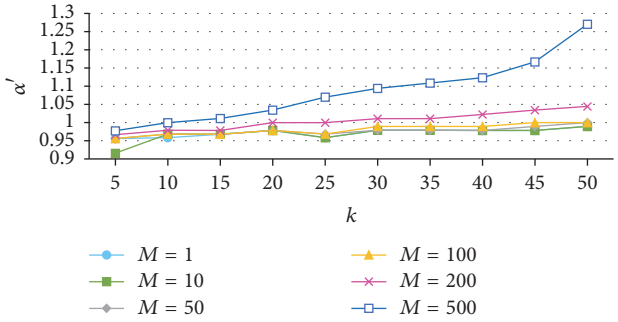

Figure 8: α' versus k of the uniform dataset.

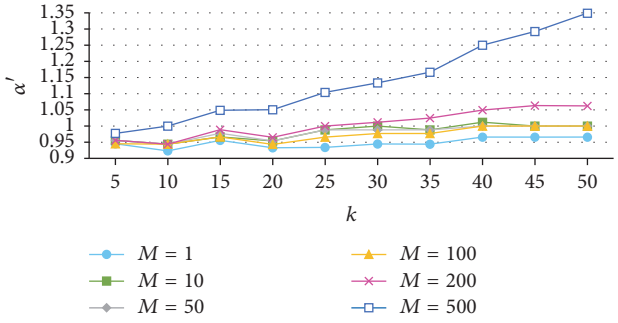

Figure 9: α' versus k of the real-world dataset.

and multiplications will be too wasteful to find at most k POIs. In order to find the maximum M that satisfies the high accuracy and significantly reduces the computation cost, Utsunomiya et al. [9] defined the ratio α as

$$\alpha = \frac{r_{\text{LPCQP}}}{r_{\text{PCQP}}}, \tag{14}$$

where r_{LPCQP} and r_{PCQP} denote the accuracy rate of LPCQP and PCQP. The performance results showed that k/n_p should be 0.5 or less in order to achieve $\alpha \geq 0.95$. It means that when $n_p \geq 2k$, the scheme can keep a considerable high accuracy and reduce the computation cost.

When $k > n_p/2$, the entries lost from aggregating the subtables are too much. The situation will get worse when $k > n_p$, and the user will receive less than k results in this case.

In our scheme, $n_p = 2k$. As mentioned above, EPCQP achieves a high query accuracy rate and reduces the computation cost which is presented in Section 4.2. The advantage of EPCQP is obvious compared with the case that $k > n_p$ in LPCQP.

Figures 6 and 7 represent the accuracy rate versus k for the different datasets. It shows that the accuracy rate of EPCQP is higher than that of LPCQP when M is 500. Although the accuracy rate of EPCQP is slightly lower than that of PCQP and LPCQP when $M \leq 100$, it is still higher than that of LPCQP when $k > 25$ and $M = 200$. The accuracy rate of EPCQP is higher than 90% even if k is large for the uniform dataset, and it is higher than 84% when k is large for the real-world dataset. We define α' as

$$\alpha' = \frac{r_{\text{EPCQP}}}{r_{\text{LPCQP}}}, \tag{15}$$

where r_{EPCQP} denotes the accuracy rate of EPCQP. Figures 8 and 9 indicate the ratio α' versus k for the two datasets. When $M = 1$, the value of α' denotes the ratio of r_{EPCQP} to r_{PCQP}. Note that the ratio α' is kept to 0.9 or above regardless of the various k for the two datasets. Particularly, for the cases of $M = 500$, EPCQP achieves $\alpha' \geq 1$ when $k \geq 10$. As shown in Figures 8 and 9, EPCQP keeps high accuracy rate of LPCQP when $M < 500$ and improves the accuracy rate when $M = 500$. Besides, the advantage of EPCQP will be more significant when M is larger than 500.

4.2. Computation Cost. In the query process of EPCQP, the server first divides the POI-table into M subtables. Compared with calculation of ciphertexts and homomorphic encryption/decryption, the cost of dividing table is negligible. As presented in Section 3.2.3, the server multiplies each element of $\varepsilon_{\text{pk}}(q_M)$ by the POI-info column of the corresponding subtable and then aggregates all the subtables into a table t'_{sub}. The process of aggregating requires n multiplication and $2k(M-1)$ additions according to (8). Finally, the server multiplies $\varepsilon_{\text{pk}}(P^t)$ with the aggregated table t'_{sub} to obtain the circularly shifted and decrypted POI-info data which requires $k(2k-1)$ additions and $(k \times 2k)$ multiplications.

Table 1 represents the computation cost of PCQP, LPCQP, and EPCQP. Compared with the cost of shifting process, the cost on aggregating subtables can be negligible when k is

Table 1: Comparison of computation cost on the server.

Scheme	Addition	Multiplication
PCQP	$k(n-1)$	kn
LPCQP	$n_p(M-1)+k(n_p-1)$	$n+kn_p$
EPCQP	$2k(M-1)+k(2k-1)$	$n+2k\times k$

n and n_p denote the number of all entries in the POI-table and a subtable, respectively.

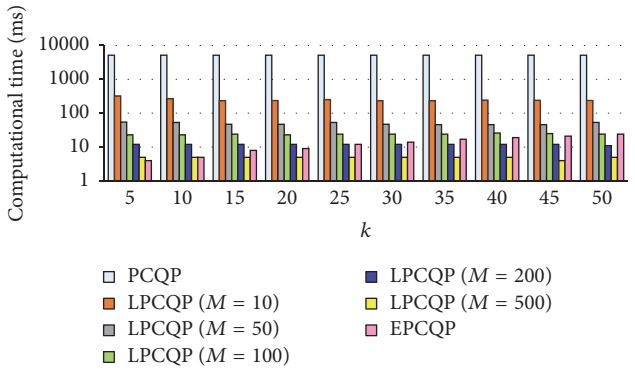

Figure 10: Computational time for encrypting the circular shift matrix.

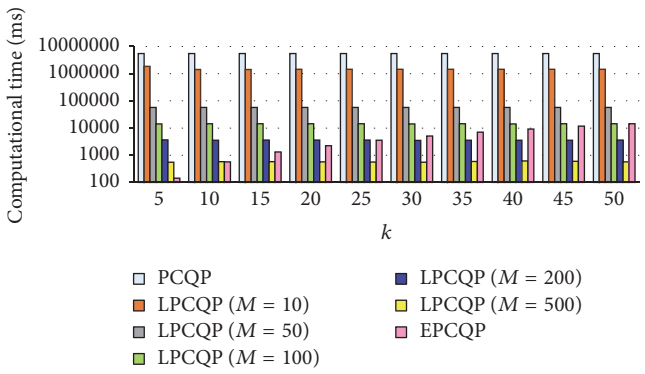

Figure 11: Computational time for the shifting process.

large. As mentioned in Section 4.1, in order to achieve a high accuracy in LPCQP, n_p should be greater than or equal to $2k$. As shown in Table 1, the cost of multiplication in EPCQP is lower than that in LPCQP when $n_p > 2k$. Besides, the cost of addition in the shifting process is reduced obviously. Therefore, our proposed scheme has a high accuracy with the lower computation cost.

The proposed scheme has the approximate computational time for aggregating subtables compared with that in LPCQP. Figures 10 and 11 represent the computational time for encrypting the circular shift matrix and the shifting process, respectively. As shown in Figure 10, the computation cost on encrypting the circular shift matrix in EPCQP becomes one-thousandth of that in PCQP or below when $k \leq 20$, and it is about one-tenth of that in LPCQP regardless of the various k when $M = 10$. Figure 11 shows that the computation cost of the shifting process in EPCQP becomes one-thousandth of

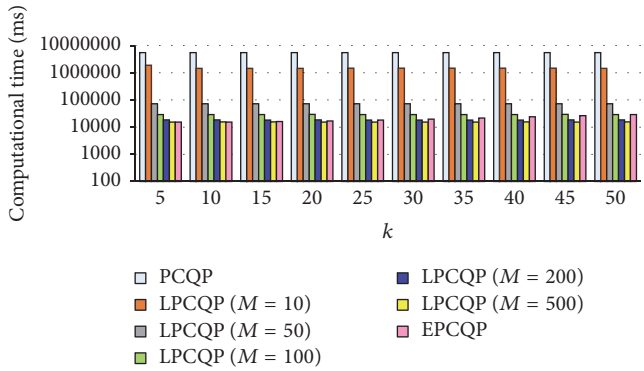

Figure 12: Total computation cost.

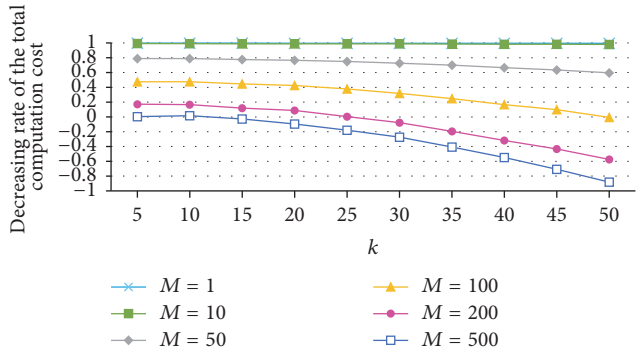

Figure 13: Decreasing rate of the total computation cost.

that in PCQP or below when $k \leq 30$, and it is one percent of that in LPCQP regardless of the various k when $M = 10$. Figures 10 and 11 show that the computation cost of EPCQP is lower than that in LPCQP when $M \leq 100$. Although the computation cost of EPCQP is slightly higher than that in LPCQP when $M = 200$ and $k > 25$, the accuracy rate of EPCQP is higher than that of LPCQP. Figure 12 represents the total computation cost of the PCQP, LPCQP, and EPCQP. As shown in Figure 12, the computation cost of our scheme is lower than that of LPCQP regardless of k when $M \leq 100$. The decreasing rate of the total computation cost is represented in Figure 13. When $M = 1$, it denotes the decreasing rate of the total computation cost compared with that of PCQP. From these results, the computation cost of our proposed scheme is reduced by 99% or more compared with that of PCQP. When $k \leq 25$ and $M \leq 200$, the decreasing rates are positive number, and it means that the computation cost of EPCQP is lower than that of LPCQP.

Based on the above result, we can say that EPCQP keeps high accuracy rate of LPCQP while reducing the computation cost. There is a trade-off between the accuracy rate and the computation cost of EPCQP.

4.3. Communication Cost. In this section, we discuss the communication cost of our scheme. Table 2 shows the communication cost of PCQP, LPCQP, and EPCQP. As described in Section 3.2.4, the encrypted matrix is constructed from the first row. Let l denote the bit-length of an encrypted POI-info.

Table 2: Comparison of communication cost.

Scheme	Uplink	Downlink
PCQP	$l \times n$	$l \times k$
LPCQP	$l\left(n_p + M\right)$	$l \times k$
EPCQP	$l\,(2k + M)$	$l \times k$

n and n_p denote the number of all entries in the POI-table and a subtable, respectively.

The user sends $\varepsilon_{\mathrm{pk}}(P^t_{0,j})$ as $(l \times n)$ bits to the server in PCQP. In LPCQP, the user sends $\varepsilon_{\mathrm{pk}}(P^t_{0,j})$ and $\varepsilon_{\mathrm{pk}}(q_M)$ as $l(n_p + M)$ bits. The communication cost of our scheme is $l(2k + M)$ bits. After searching, the server returns k results to the user.

As showed in Table 2, the downlink communication cost of EPCQP is the same as PCQP and LPCQP. In the uplink process, the communication cost of EPCQP is lower than that of PCQP. In summary, the communication cost of our scheme is in the same degree compared to LPCQP; meanwhile it achieves high accuracy.

5. Related Work

The existing methods for location privacy protection mainly fall into three categories.

5.1. Spatial Cloaking. Spatial cloaking [31–39] methods generate a cloaking region to the location server and the server returns the query results to users or a trusted third party. K-anonymity is the most common model [40], which was firstly implemented in LBS by Gruteser and Grunwald [1]. Chow et al. [5] maintains the location information of the user by using the R-tree structure, which is the classic method for protecting the location of users by using the k-anonymity model. It proposes a scheme that accurately searches k-nearest neighbors in the rectangular anonymous region.

5.2. Location Obstruction. The user continues to submit queries with a specific fake location to the location-based server, and the server iteratively returns the results based on the fake location until the user obtains the result that satisfies the privacy and security requirements. The classic algorithm of location obstruction is SpaceTwist [41], which requires multiple rounds of communication, and the communication cost required for each complete query is large.

5.3. Spatial Transformation. The principle of spatial transformation for protecting location privacy is to convert the location information from conventional data space to another. The scheme [42] utilizes homomorphic encryption to accomplish data interaction between users and servers. Although it achieves strong privacy protection, it is difficult to be adapted to the application environment of continuous queries and real-time responses with extremely expensive computation cost. In order to reduce the computation cost, a scheme [6] based on Hilbert curve is proposed. The scheme, which effectively reduces the computation cost of encryption, transforms all POIs in two-dimensional space into a sequence of integers in one-dimensional sequence and maintains the original neighborhood relationship approximately. The drawback of this scheme is that the query accuracy is not high. A k-anonymous spatial region construction mechanism [7] is proposed for distributed systems. It combines the user location with Hilbert-order to form a spatial area with other peer nodes and then sends the area and the query requirement to the server. HilAnchor scheme [43] is based on SpaceTwist and Hilbert curve. With only two rounds of communication, the user can get the exact k-nearest neighbor POIs without the leakage of the location. The MobiCrowd algorithm [44] utilizes the buffer to guarantee that queries can be accomplished locally in order to reduce the communication cost.

Query privacy is as important as location privacy. The scheme [45] generates dummy queries so that the server and the attacker cannot obtain the preference information of the user by summarizing the rules of the query contexts. An attack mode [46] marks a query according to the query context, the location, and the querying time. Reference [47] proposed the k-Approximate Beyond Suspicion scheme, which first utilizes a clustering algorithm (such as the K-means) to cluster the users who have the similar location and issue the similar queries and then calculates the anonymous areas according to k-anonymous to protect the privacy of the location and query.

6. Conclusion

In this paper, we propose a privacy-preserving circular query protocol with high accuracy and low complexity, which can be utilized in the location-based k-NN search. With the circular shift and the homomorphic encryption, the proposed scheme accomplishes the efficient querying and the privacy protection simultaneously. We adopt the method that the server dynamically divides the encrypted POI-table according to the query of the user. Our scheme mitigates the drawbacks of PCQP and LPCQP without impairing the advantages of them. The computation cost is reduced by 99% or more compared with that of PCQP by allowing a 6% reduction in the accuracy rate regardless of k. Comparing with LPCQP when $M \leq 100$, the computation cost is reduced by up to 47.5% with a 5% reduction in the accuracy rate. With the rapid development of the spatial crowdsourcing, the location privacy has attracted more and more attention. We expect our scheme will inspire the research of location privacy protection and encrypted data computing in spatial crowdsourcing.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Key Research and Development Program of China under Grants

2017YFB0802202 and 2017YFB0802704 and Program of Shanghai Technology Research Leader under Grant 16XD1424400.

References

[1] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, pp. 31–42, ACM, San Francisco, Calif, USA, 2003.

[2] M. F. Mokbel, "Towards Privacy-Aware Location-Based database servers," in *Proceedings of the 22nd International Conference on Data Engineering Workshops, ICDEW 2006*, USA, April 2006.

[3] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preventing location-based identity inference in anonymous spatial queries," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 12, pp. 1719–1733, 2007.

[4] D. Papadias, D. Zhang, and G. Kollios, *Advances in Spatial and Temporal Databases*, vol. 4605, Springer, Berlin, Germany, 2007.

[5] C.-Y. Chow, M. F. Mokbel, and W. G. Aref, "Casper*: query processing for location services without compromising privacy," *ACM Transactions on Database Systems (TODS)*, vol. 34, no. 4, article 24, 2009.

[6] J.-H. Um, H.-D. Kim, and J.-W. Chang, "An advanced cloaking algorithm using Hilbert curves for anonymous location based service," in *Proceedings of the 2nd International Conference on Social Computing*, pp. 1093–1098, Minneapolis, MN, USA, August 2010.

[7] A.-A. Hossain, A. Hossain, H.-K. Yoo, and J.-W. Chang, "H-star: Hilbert-order based star network expansion cloaking algorithm in road networks," in *Proceedings of the 14th IEEE Int. Conf. on Computational Science and Engineering, CSE 2011, the 11th International Symposium on Pervasive Systems, Algorithms, and Networks, I-SPAN 2011, and the 10th IEEE Int. Conf. on Ubiquitous Computing and Communications, IUCC 2011*, pp. 81–88, chn, August 2011.

[8] I.-T. Lien, Y.-H. Lin, J.-R. Shieh, and J.-L. Wu, "A novel privacy preserving location-based service protocol with secret circular shift for k-NN search," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 6, pp. 863–873, 2013.

[9] Y. Utsunomiya, K. Toyoda, and I. Sasase, "LPCQP: Lightweight private circular query protocol with divided POI-table and somewhat homomorphic encryption for privacy-preserving k-NN search," *Journal of Information Processing*, vol. 24, no. 1, pp. 109–122, 2016.

[10] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," in *Proceedings of the 3rd Conference on Innovations in Theoretical Computer Science, ITCS 2012*, pp. 309–325, usa, January 2012.

[11] H. Sagan, *Space-Filling Curves*, Springer, New York, NY, USA, 1994.

[12] D. Hilbert, "Ueber die stetige abbildung einer linie auf ein flächenstück," *Mathematische Annalen*, vol. 38, no. 3, pp. 459-460, 1891.

[13] B. Moon, H. V. Jagadish, C. Faloutsos, and J. H. Saltz, "Analysis of the clustering properties of the Hilbert space-filling curve," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 1, pp. 124–141, 2001.

[14] H. V. Jagadish, "Linear clustering of objects with multiple attributes," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 332–342, Atlantic City, NJ, USA, May 1990.

[15] E. H. Moore, "On certain crinkly curves," *Transactions of the American Mathematical Society*, vol. 1, no. 1, pp. 72–90, 1900.

[16] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," *Foundations of Secure Computation*, vol. 4, no. 11, pp. 169–180, 1978.

[17] C. Gentry, "Fully Homomorphic Encryption Using Ideal Lattices," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC '09*, pp. 169–178, usa, June 2009.

[18] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2010)*, Lecture Notes in Comput. Sci., pp. 24–43, Springer.

[19] N. P. Smart and F. Vercauteren, "Fully homomorphic encryption with relatively small key and ciphertext sizes," in *Proceedings of the International Workshop on Public Key Cryptography*, vol. 6056 of *Lecture Notes in Comput. Sci.*, pp. 420–443, Springer.

[20] C. Gentry and S. Halevi, "Implementing Gentry's fully—homomorphic encryption scheme," in *Advances in cryptology—EUROCRYPT 2011*, vol. 6632 of *Lecture Notes in Computer Science*, pp. 129–148, Springer, Heidelberg, Germany, 2011.

[21] C. Gentry and S. Halevi, "Fully homomorphic encryption without squashing using depth-3 arithmetic circuits," in *Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011*, pp. 107–116, 2011.

[22] J.-S. Coron, D. Naccache, and M. Tibouchi, "Public key compression and modulus switching for fully homomorphic encryption over the integers," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2012)*, vol. 7237 of *Lecture Notes in Comput. Sci.*, pp. 446–464, Springer.

[23] S. Halevi and V. Shoup, "Design and implementation of a homomorphic-encryption libirary," *IBM Research*, pp. 12–15, 2013.

[24] N. P. Smart and F. Vercauteren, "Fully homomorphic SIMD operations," *Designs, Codes and Cryptography*, vol. 71, no. 1, pp. 57–81, 2014.

[25] C. Gentry, S. Halevi, and N. P. Smart, "Homomorphic evaluation of the AES circuit," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 7417, pp. 850–867, 2012.

[26] J. Wu, S. Pan, X. Zhu, and Z. Cai, "Boosting for multi-graph classification," *IEEE Transactions on Cybernetics*, vol. 45, no. 3, pp. 430–443, 2015.

[27] J. Wu, X. Zhu, C. Zhang, and P. S. Yu, "Bag Constrained Structure Pattern Mining for Multi-Graph Classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 10, pp. 2382–2396, 2014.

[28] J. Wu, S. Pan, X. Zhu, C. Zhang, and X. Wu, "Positive and Unlabeled Multi-Graph Learning," *IEEE Transactions on Cybernetics*, vol. 47, no. 4, pp. 818–829, 2016.

[29] J. Wu, S. Pan, X. Zhu, C. Zhang, and X. Wu, "Multi-instance learning with discriminative bag mapping," *IEEE Transactions on Knowledge and Data Engineering*, no. 99, p. 16, 2018.

[30] J. Wu, S. Pan, X. Zhu, C. Zhang, and P. S. Yu, "Multiple Structure-View Learning for Graph Classification," *IEEE Transactions on Neural Networks and Learning Systems*, no. 99, pp. 1–16, 2017.

[31] C. Bettini, S. Mascetti, X. S. Wang, D. Freni, and S. Jajodia, "Anonymity and historical-anonymity in location-based services," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 5599, pp. 1–30, 2009.

[32] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "ℓ-diversity: Privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, Article ID 1217302, 2007.

[33] M. F. Mokbel, C. Y. Chow, and W. G. Aref, "The new Casper: query processing for location services without compromising privacy," in *Procceedings of the 32nd International Conference on Very Large Data Bases*, pp. 763–774, 2006.

[34] R. Dewri, I. Ray, I. Ray, and D. Whitley, "On the formation of historically k-anonymous anonymity sets in a continuous LBS," *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, vol. 50, pp. 71–88, 2010.

[35] A. Hossain, A.-A. Hossain, S.-J. Jang, Y.-S. Shin, and J.-W. Chang, "K-anonymous cloaking algorithm based on weighted adjacency graph for preserving location privacy," in *Proceedings of the 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom-2012*, pp. 358–365, UK, June 2012.

[36] P. Shankar, Y.-W. Huang, P. Castro, B. Nath, and L. Iftode, "Crowds replace experts: Building better location-based services using mobile social network interactions," in *Proceedings of the 10th IEEE International Conference on Pervasive Computing and Communications, PerCom 2012*, pp. 20–29, Switzerland, March 2012.

[37] Y. Wang, J. Peng, L.-P. He, T.-T. Zhang, and H.-Z. Li, "LBSs privacy preserving for continuous query based on semi-honest third parties," in *Proceedings of the IEEE 31st International Performance Computing and Communications Conference (IPCCC '12)*, pp. 384–391, IEEE, December 2012.

[38] Y. Wang, H. Zhou, Y. Wu, and L. Sun, "Preserving location privacy for location-based services with continuous queries on road network," in *Proceedings of the 2012 7th International Conference on Computer Science and Education, ICCSE 2012*, pp. 822–827, Australia, July 2012.

[39] H. Hu, J. Xu, Q. Chen, and Z. Yang, "Authenticating location-based services without compromising location privacy," in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, SIGMOD '12*, pp. 301–312, May 2012.

[40] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[41] M. L. Yiu, C. S. Jensen, X. Huang, and H. Lu, "SpaceTwist: managing the trade-offs among location privacy, query performance, and query accuracy in mobile services," in *Proceedings of the IEEE 24th International Conference on Data Engineering (ICDE '08)*, pp. 366–375, IEEE Press, Cancun, Mexico, April 2008.

[42] G. Ghinita, P. Kalnis, M. Kantarcioglu, and E. Bertino, "Approximate and exact hybrid algorithms for private nearest-neighbor queries with database protection," *GeoInformatica*, vol. 15, no. 4, pp. 699–726, 2011.

[43] W. Ni, J. Zheng, and Z. Chong, "HilAnchor: location privacy protection in the presence of users' preferences," *Journal of Computer Science and Technology*, vol. 27, no. 2, pp. 413–427, 2012.

[44] R. Shokri, P. Papadimitratos, G. Theodorakopoulos, and J.-P. Hubaux, "Collaborative location privacy," in *Proceedings of the 8th IEEE International Conference on Mobile Ad-hoc and Sensor Systems, MASS 2011*, pp. 500–509, Spain, October 2011.

[45] A. Pingley, N. Zhang, and X. Fu, "Protection of query privacy for continuous location based services," in *Proceedings of the INFOCOM*, pp. 1710–1718, IEEE, 2011.

[46] X. Chen and J. Pang, "Exploring dependency for query privacy protection in location-based services," in *Proceedings of the 3rd ACM Conference on Data and Application Security and Privacy, CODASPY 2013*, pp. 37–47, USA, February 2013.

[47] X. Chen and J. Pang, "Measuring query privacy in location-based services," in *Proceedings of 2nd ACM International Conference on Data and Application Security and Privacy*, pp. 49–60, San Antonio, Tex, USA, Feburary 2012.