The privacy protection algorithm of ciphertext nearest neighbor query based on the single Hilbert curve

Delin Tan[1,2*], and Huajun Wang[2]
[1] Sichuan Normal University, Chengdu, 610068, China
[e-mail:tdltcl@sicnu.edu.cn]
[2] School of Geophysics, Chengdu University of Technology, Chengdu, 610059, China
[e-mail:hjwang@sdut.edu.cn]
*Corresponding author: Delin Tan

Received March 21, 2022; revised May 27, 2022; revised June 28, 2022; accepted July 28, 2022; published September 30, 2022

Abstract

Nearest neighbor query in location-based services has become a popular application. Aiming at the shortcomings of the privacy protection algorithms of traditional ciphertext nearest neighbor query having the high system overhead because of the usage of the double Hilbert curves and having the inaccurate query results in some special circumstances, a privacy protection algorithm of ciphertext nearest neighbor query which is based on the single Hilbert curve has been proposed. This algorithm uses a single Hilbert curve to transform the two-dimensional coordinates of the points of interest into Hilbert values, and then encrypts them by the order preserving encryption scheme to obtain the one-dimensional ciphertext data which can be compared in numerical size. Then stores the points of interest as elements composed of index value and the ciphertext of the other information about the points of interest on the server-side database. When the user needs to use the nearest neighbor query, firstly calls the approximate nearest neighbor query algorithm proposed in this paper to query on the server-side database, and then obtains the approximate nearest neighbor query results. After that, the accurate nearest neighbor query result can be obtained by calling the precision processing algorithm proposed in this paper. The experimental results show that this privacy protection algorithm of ciphertext nearest neighbor query which is based on the single Hilbert curve is not only feasible, but also optimizes the system overhead and the accuracy of ciphertext nearest neighbor query result.

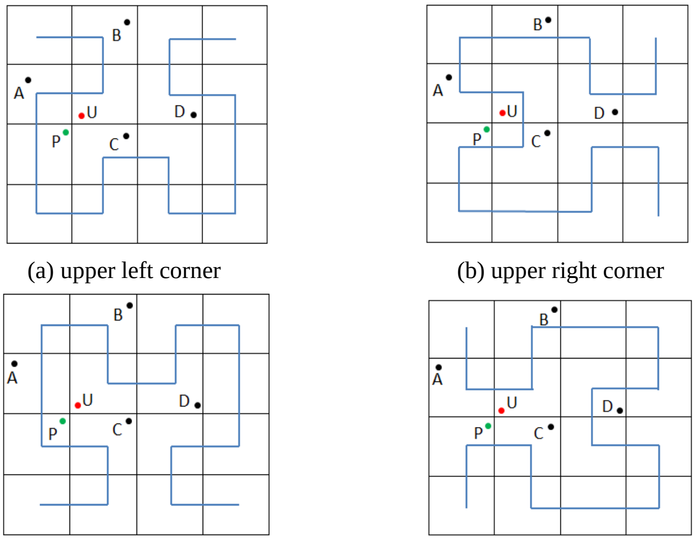
Keywords: Ciphertext, Nearest Neighbor Query, Single Hilbert Curve, System Overhead, Accuracy

research was supported by funding of the Experimental Equipment Division of Sichuan Normal University (No.SYJS2021031)

http://doi.org/10.3837/tiis.2022.09.014 **ISSN : 1976-7277**

1. Introduction

With the rapid development of mobile computing technology, wireless network technology, GPS technology and GIS technology etc., LBS (location-based services, LBS) came into being and are widely used. For example, the NNQ (nearest neighbor query, NNQ) is one of the commonly used applications in location-based services. When LBS users need to use the nearest neighbor query, they are usually forced to send their private information, such as their identity information, real-time location, and query content etc., to the LBS server, otherwise they will not be able to use this service. During the process of using the nearest neighbor query, there are three places where privacy leakages are at risk, namely the mobile terminals held by the LBS users, the wireless communication link and the LBS server. At present, the academic community usually does not conduct in-depth research on the privacy protection of the first two places, and mainly focuses on the privacy protection on the LBS server. If there is no privacy protection for the relevant information of the LBS user stored on the LBS server which is generally considered an untrusted third-party server, it is easily to be attacked by malicious attackers. The losses may lead to the privacy leakage of LBS users in lighter cases, even and the safety of life and property of LBS users may be threatened in severe cases. In addition, the privacy protection of user’s information is a necessary prerequisite for the current application to be launched according to the ‘Data Security Law’ [1]. Therefore, the privacy protection of the nearest neighbor query has received extensive attention and research in the academic community, and related algorithms with different security levels are proposed according to the different requirements. Here, the privacy protection algorithms of ciphertext NNQ have attracted much attention and favor in the academic community because of the high-level privacy protection effect and the ciphertext computational property. However, there are still some shortcomings in this type of privacy protection algorithms: (1) Traditional ciphertext NNQ privacy protection algorithms usually need to use double Hilbert curves to optimize the accuracy of query results, and it is obvious that the system overhead is exponential comparing with the single Hilbert curve. (2) The traditional ciphertext NNQ privacy protection algorithms can’t obtain accurate query results under some special cases.



(a) upper left corner (b) upper right corner

(c) lower left corner (d) lower right corner

Fig. 1. Schematic diagram for special cases without accurate results

Take **Fig. 1** as an example, the truly nearest neighbor POI (point of interest, POI) can't be obtained in all various situations of the traditional ciphertext NNQ privacy protection algorithms. As shown in **Fig. 1**, the truly nearest neighbor POI of LBS user who locates in U is P. However, the approximate nearest neighbor POIs are {A, B} in **Fig. 1** (a), {A, C} in **Fig. 1** (b), {B, D} in **Fig. 1**(c), {A, B} in **Fig. 1** (d). It can be seen that no matter which combination of double Hilbert curves, the traditional ciphertext NNQ privacy protection algorithms can't obtain the truly nearest neighbor POI in the situation shown in **Fig. 1**.

It can be seen from the above analysis that if there is a privacy protection algorithm of ciphertext NNQ which only needs a single Hilbert curve, but can obtain the truly query result in any case, it will be very meaningful. So this paper proposes a privacy protection algorithm of ciphertext NNQ based on the single Hilbert curve, namely CNNQBSH (Ciphertext nearest neighbor query based on the single Hilbert curve, CNNQBSH) privacy protection algorithm. This privacy protection algorithm includes many operations, such as Hilbert space transformation, order preserving encryption, symmetric encryption, etc. The work process is consists of two stages, the initialization processing stage and the real-time query processing stage respectively. Here, the intention of the initialization processing stage is to perform Hilbert space transformation and order preserving encryption for the two-dimensional coordinates of the POIs in the designated region, and symmetric encryption of corresponding textual data of the POIs. After the above processing, a one-dimensional ciphertext value that can be compared and can uniquely identify the corresponding two-dimensional coordinate is obtained. Then, the one-dimensional ciphertext value is used as the index value, and the corresponding symmetric ciphertext of POI's textual data are stored as a data node in the database of the LBS server. The real-time query processing stage is to perform Hilbert space transformation, and order preserving encryption on the two-dimensional coordinate of the LBS user, and then generate the corresponding ciphertext NNQ request according to the plaintext request content of the LBS user, and then send it to the LBS server for query processing to obtain approximate nearest neighbor query results. Finally, the precision processing algorithm is called to perform the precision processing on the approximate nearest neighbor query results to obtain the truly nearest neighbor query result required by the LBS user.

The main contributions of this paper are as follows:

(1) This paper proposes a ciphertext nearest neighbor query privacy protect algorithm based on the single Hilbert curve.

(2) This paper proposes a single point query algorithm for ciphertext points of interest in the ELT (Encoded look up table, ELT) table.

(3) This paper proposes a range query algorithm for ciphertext points of interest in the ELT table.

(4) This paper proposes a precision processing algorithm which can refine the approximate nearest neighbor query result set.

(5) Conduct simulation experiments on the above algorithms on simulated data.

2. Related work

Since this paper will use some additional knowledge, such as Hilbert space transformation, OPES (order preserving encryption scheme, OPES) and so on, for ease of reading, this section will give a brief introduction to its concept and related research. Hilbert proposed the concept of space transformation in 1891, meanwhile proposed the space transformation curve to reduce the dimension of high-dimensional space. In honor of his contribution, this type of space transformation curve is called Hilbert curve [2]. Orenstein et al. applied spatial transformation curves to object-oriented database for spatial query processing [3]. Jagadish used Hilbert spatial transformation in spatial database in order to realize the transformation from multi-dimensional index to one-dimensional index [4]. Bader redefined the concept of space transformation curves, and introduces Hilbert space transformation curves, Z space transformation curves, etc., and finally applies them to new applications such as privacy protection [5]. Order preserving encryption is an encryption scheme that can reflect the size order of the plaintext on the ciphertext. Agrawal et al. firstly proposed the concept of order preserving encryption and used it to protect the privacy of digital data [6]. In order to satisfy the range query of encrypted data by the database community, Boldyreva et al. proposed order preserving symmetric encryption based on the existing order preserving encryption scheme [7]. In order to solve the problem that the ROPF (random order preserving function, ROPF) characterizes the leakage of the underlying data, Boldyreva et al. introduced POPE (pseudorandom order preserving function, POPE) and MOPE (modular order preserving encryption, MOPE) respectively, thereby improved the security of the traditional order preserving symmetric encryption scheme [8].

Due to the privacy protection algorithm of ciphertext NNQ can implement the privacy protection effect while also implementing nearest neighbor query operation on the ciphertext POI database without decryption, meanwhile it has the high privacy protection effect, so it has received extensive attention and in-depth research in the academic community. Khoshgozaran et al. proposed the concept of blind query [9], whose motivation is to use a single Hilbert space conversion curve to convert the two-dimensional coordinates of the POIs into the one-dimensional coordinate value which can preserve the proximity of the two-dimensional space, and uniquely identify the two-dimensional coordinates. Since the Hilbert space conversion is performed on the trusted anonymous server, it can be regarded as a simple encryption scheme as long as the relevant parameters, such as the conversion order, starting point and curve direction etc., are used as secret keys. However, the blind query result is less accuracy due to the single Hilbert space transformation curve. Therefore, Khoshgozaran et al. introduced the double Hilbert space transformation curve to furtherly optimize the accuracy of query result. In addition, in order to strengthen the security of the privacy protection algorithm, it also hashes the converted one-dimensional coordinates by using a hash function such as MD5, and stores the hash values in the ELT table which is the encoded look up table and is stored on the LBS server. After optimizing, the privacy protection algorithm is highly secure and is provably secure [10]. In order to solve the problem of wasting storage resources due to a large number of empty grids in the Hilbert space transformation, Tian et al. introduced a decision tree pruning technology to prune the quad storage tree of POIs, and to optimize the system storage space and retrieval efficiency [11]. Zhou et al. proposed a corresponding privacy protection algorithm that supports ciphertext calculation based on the additive homomorphic properties of the Paillier algorithm, combining with PIR technology and K-anonymity algorithm [12]. Although this algorithm can realize other homomorphic operations through addition homomorphic operation, it can't realize efficient homomorphic comparison, homomorphic multiplication, homomorphic square rooting and other ciphertext operations, so its operations efficiency was poor. Shen et al. firstly perform gridding on the region where the LBS users are located, then send the processing results and related parameters to the LBS server [13]. LBS user sends the ciphertext query request to the trusted anonymous server, and the LBS server performs gridding of the region where the LBS user is located. In addition, the LBS server also performs order preserving encryption on all POIs in the region, and returns it to the trusted anonymous server. After the trusted anonymous server obtains the ciphertext POIs, it uses the corresponding data item in the ciphertext query request of the LBS user to compare the ciphertext value, and find out the desired query results. Finally, the trusted anonymous server returns the query results to the LBS user holding the mobile client. This privacy protection algorithm is similar to the one proposed in this paper which also uses the grid processing and order preserving encryption algorithms. The difference between them is that the ciphertext operations of this algorithm are performed on the trusted anonymous server instead of the LBS server, so it still needs to transmit a large number of invalid ciphertext POIs to the trusted anonymous server, so this algorithm are prone to waste of system computing resources and communication overhead.

3. Symbols and Concept Definitions in this paper

Some symbols and definitions will be involved in this paper, and this section will introduce and explain them for ease of reading.

3.1 Symbol Description

Table 1. Symbol Description

Symbol	Definition
A_{llx}, A_{lly}	The x and y coordinate of the lower left corner of the given region A
A_{rux}, A_{ruy}	The x and y coordinate of the upper right corner of the given region A
B_{llx}, B_{lly}	The x and y coordinate of the lower left corner of the squared region B
B_{rux}, B_{ruy}	The x and y coordinate of the upper right corner of the squared region B
N	The order of the Hilbert space transformation curve
H	The Hilbert value corresponding to the two-dimensional coordinate
C_H	The order preserving encryption ciphertext of the Hilbert value
C_T	Symmetric ciphertext corresponding to the textal information of POI

3.2 Definition Introduction

Definition 1: Nearest neighbor query is a location-based service application that queries the closest POI to the current location of the LBS user.

Definition 2: The ciphertext approximate nearest neighbor query refers to the ciphertext nearest neighbor query, but its query results can’t guarantee the accuracy. It usually contains two sub-algorithms, namely ciphertext single-point query and ciphertext range query.

Definition 3: The ciphertext single point query refers to a query algorithm that queries the relevant information of the POI corresponding to the given ciphertext value C_H in the database.

Definition 4: The ciphertext range query refers to a query algorithm that queries the relevant information of all POIs corresponding to the given ciphertext value range [C_{Ha}, C_{Hb}] in the database.

Definition 5: Precise processing refers to the process of filtering and refining the approximate nearest neighbor query results obtained by the ciphertext approximate nearest neighbor query algorithm.

4. How this proposed privacy protection algorithm works

In order to introduce the working principle of the CNNQBSH privacy protection algorithm proposed in this paper, this section will introduce it from the aspects of its architecture, workflow, and data structure and so on.

4.1 Architecture

When implementing the CNNQBSH privacy protection algorithm, three-layer architecture commonly used in the privacy protection of location-based service will be adopted, namely, Mobile Intelligent Terminal—Trusted Transit Server—LBS server. Fig. 2 shows the architecture of the CNNQBSH privacy protection algorithm, and the functions of each part which will be introduced below.

MIT (Mobile Intelligent Terminal, MIT): It is a mobile terminal device with GPS function and encryption/decryption function held by LBS users, such as smart phone, tablet computer etc. Its functions are to provide positioning, initiate nearest neighbor query requests, receive and decrypt the returned ciphertext accurate nearest neighbor query result etc. for LBS user who already has used the nearest neighbor query service.

TTS (Trusted Transit Server, TTS): It is an authoritative and trusted transit server for the management of location-based services applications, and it is composed of three parts: cipher machine, space converter and refiner. The functions of the cipher machine are order preserving encryption of the one-dimensional coordinates obtained by Hilbert space transformation, symmetric encryption and decryption of the textual data of points of interest, and symmetric encryption of the accurate nearest neighbor query result that needs to be returned to the mobile intelligent terminal etc. The function of the space converter is to perform operations such as Hilbert space conversion on the two-dimensional coordinates corresponding to the positions of all POIs and LBS users in the designated region. The functions of refiner are to filter and refine the approximate nearest neighbor query result set returned from the LBS server.

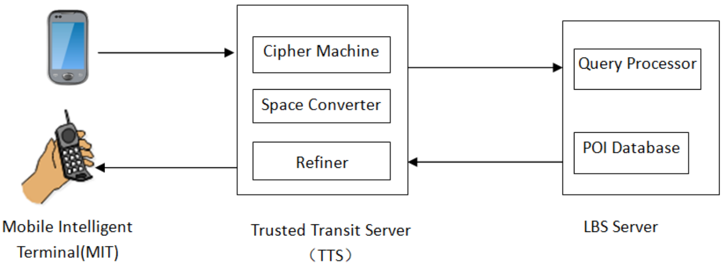


Fig. 2. The architecture of the proposed algorithm

LS (LBS Server, LS)：It is composed of query processor and POI database. The motivation of the query processor is to handle the ciphertext approximate nearest neighbor query request sent from the trusted transit server, and return the query result to the trusted transit server. The function of POI database is to store and provide retrieval of ciphertext points of interest. Here, the ELT table proposed by Khoshgozaran et al. will be used as the POI database.

4.2 The workflow of the proposed algorithm

CNNQBSH algorithm includes two stages, the initialization processing stage and the real-time query processing stage. The workflow of these two processing stages will be described in detail below.

The workflow of the initialization processing stage:

（1）First, square the given region A, and a minimum square region B which surrounds A will be obtained.

（2）Gridding the region B, namely divide region B into M rows and M columns from the horizontal and vertical directions respectively, and M*M grids will be obtained. Each grid is identified by its row number i and column number j, where 0<=i, j<=M-1.

（3）Fill M*M grids with the N-order Hilbert space transformation curve, and make each gird to be visited once by the Hilbert space transformation curve, where $M=2^N$.

（4）Call algorithm 1 to calculate the grid coordinates p(i,j) corresponding to the two-dimensional coordinates p(x,y) of all POIs in the given region, and then call the Hilbert value calculation algorithm proposed by Faloutsos to calculate the Hilbert value corresponding to the grid coordinate p(i,j) [14]. Then, the spatial transformation and dimensionality reduction of the two-dimensional coordinates of POIs in the given region is done, and it is uniquely identified by the one-dimensional Hilbert value.

（5）Call the order preserving encryption algorithm proposed by Boldyreva et al. to encrypt the one-dimensional Hilbert values [8], and obtain the ciphertext value C_H that can preserve the order of the plaintext space. In addition, it is also necessary to encrypt the textual data of the POIs, such as the name, description and other related information, by the AES encryption algorithm, and obtain the ciphertext value C_T.

（6）Store the POIs of the given region on database by type. The ciphertext value C_H and C_T form a node indexing by C_H, and then insert it into the corresponding type ELT table in order to provide query function for LBS Users. Here, these ELT tables are the database.

（7）So far, the initialization processing stage of CNNQBSH algorithm is completed.

The workflow of the real-time query processing stage:

（1）When a LBS user needs to use the nearest neighbor query, the mobile intelligent terminal firstly generates a nearest neighbor query request quad tuple <U_{id}, Loc, Time, Content>, where U_{id} is the identity information of the LBS user, Loc is the real-time location of the LBS user, Time is the time when the query request is sent, and Content is the specific nearest neighbor query content. Then, the mobile intelligent terminal performs link encryption on the nearest neighbor query request quad tuple by using a commonly used link encryptor, and then transmits it to the trusted transit server.

（2）The trusted transit server uses the commonly used link decryptor to decrypt the ciphertext nearest neighbor query request sent from the mobile intelligent terminal, and generates the ciphertext nearest neighbor query request CReq(C_H)=<Type,C_H> that matches the query of the ciphertext POI database of the LBS server according to the query content, and then sent CReq(C_H)=<Type,C_H> to the LBS server for query processing. Here, Type is the type of POI, and C_H is the ciphertext Hilbert value corresponding to the two-dimensional coordinates of the location.

（3）When LBS server receives the ciphertext nearest neighbor query request $CReq(C_H)$ =<Type,C_H>, it calls Algorithm 2 to deal with it, then get the approximate nearest neighbor query result set, and returns it to the trusted transit server.

（4）After receiving the approximate nearest neighbor query result set, the trusted transit server determines the number of POIs in the result set. If there is only one POI in the result set, it is the approximate nearest neighbor query result. If the number of POI in the result set is greater than 1, firstly use the AES decryption algorithm to decrypt the ciphertext data C_T, and calculate the distance between the location of POIs in the approximate nearest neighbor result set and the location of the LBS user according to the location in the textual information. Finally, the closest POI is obtained, and it is also the approximate nearest neighbor query result of the LBS user.

（5）After calling Algorithm 3 to accurately deal with the approximate nearest neighbor query result, the accurate nearest neighbor query result desired by the LBS user can be obtained, and then return it to the mobile intelligent terminal after through link encryption.

（6）After receiving the above accurate ciphertext nearest neighbor query result, the mobile intelligent terminal decrypts it using link decryptor to obtain the accurate query result in plaintext, and returns it to the LBS user.

（7）Then the real-time query processing stage of CNNQBSH algorithm ends.

4.3 data structure

In CNNQBSH algorithm, there is a data structure ELT, which is based on the data table proposed by Khoshgozaran et al.[10], and stores and retrieves the POIs on the LBS server, such as formula (1) shown.

$$ELT=<<Key_i,C_{Ti}>,Prev,Next> \qquad (1)$$

In formula (1), <Key_i, C_{Ti}> is called as the entity located in the grid, and can also called as a member of the ELT table and represented by ELT.elem, where Key_i represents the order preserving encryption ciphertext of the Hilbert value corresponding to the i-th grid, C_{Ti} represents the AES ciphertext of the textual information of POI in the i-th grid. Prev is the pointer which points the previous grid, and Next is the pointer which points the next grid. Also, the number of entities in the ELT is called its length, which is represented by ELT.length.

5. Implementation of the Proposed Algorithm

Some sub-algorithms will be called in CNNQBSH algorithm, such as the algorithm to calculate the grid coordinate p(i,j) of the two-dimensional coordinate p(x,y) in the squared region B, the algorithm to calculate the Hilbert value corresponding to the grid coordinate p(i,j), the algorithm for order preserving encryption of Hilbert value, the algorithm for ciphertext approximate nearest neighbor query in ELT table, the algorithm for accurate processing of approximate nearest neighbor query results, etc. Because the algorithm for calculating the Hilbert value corresponding to the grid coordinate will use the algorithm proposed by Faloutsos [14], and the algorithm for order preserving encryption of the Hilbert value will use the algorithm proposed by Boldyreva et al. [8], they will not be introduced here. In addition, the ciphertext approximate nearest neighbor query algorithm will contain two sub-algorithms, the ciphertext single point query algorithm and the ciphertext range query algorithm respectively, they will be introduced below.

Algorithm 1: this algorithm for calculating the grid coordinate p(i,j) in the squared region B corresponding to the two-dimensional coordinate p(x,y) in the given region A.
Input: the lower left corner coordinate (B_{lx}, B_{ly}) of the squared region B， the upper right corner coordinate (B_{rx}, B_{ry}) of the squared region B, two-dimensional coordinate p(x,y) located in the given region A and the order N of Hilbert space transformation curve.
Output: The grid coordinate p(i,j) corresponding to the two-dimensional coordinate p(x,y).

```
i=$\left\lfloor \frac{x \cdot 2^N}{B_{rx} - B_{lx}} \right\rfloor$;        // Calculate the grid coordinate i of the x coordinate
j=$\left\lfloor \frac{y \cdot 2^N}{B_{ry} - B_{ly}} \right\rfloor$;        // Calculate the grid coordinate j for the y coordinate
Return p(i,j)。
```

Algorithm 2: The single point query algorithm for ciphertext points of interest
Input: ELT table、the element C_H that needs to be queried.
Output: the approximate nearest neighbor query result set for C_H.

```
low=1, high=ELT.length;
while（low<=high）{
    mid=(low+high)/2;
    if(EQ($C_H$,ELT.elem[mid].Key))        //EQ stands for equal operator
        return ELT.elem[mid];
    else if(LT($C_H$,ELT.elem[mid].Key))    //LT stands for less than operator
        high=mid-1;
    else low=mid+1;
}
return {ELT.elem[mid], ELT.elem[low-1], ELT.elem[high+1]};
```

Algorithm 3: The precision processing algorithm
Input: the approximate nearest neighbor query result set.
Output: the accurate nearest neighbor query result.
1. Calculate the distance d between the user's location and the location of the approximate nearest neighbor query result of the result set.
2. Calculate the number n of grid covered by the distance d in the squared region B. When the distance is less than the side length of one grid, it is counted as one grid.
3. Take the grid where the LBS user is located as the center, and construct a squared region C with a side length of (2n+1)*(2n+1), and calculate the Hilbert values corresponding to these grids respectively.
4. Group the above discrete Hilbert values into n runs according to their continuity [10]. Here, each run is the grid group with continuous Hilbert value.
5. Execute the ciphertext range query by calling algorithm 4 with the start and end Hilbert values of each run on the ELT table of the LBS server, and return the ciphertext data corresponding to each poi in the run, namely, <Key, C_T>.

6. After summarizing all ciphertext POIs in each run, the ciphertext data of all POIs about the squared region C can be obtained, and then return them to the trusted transit server.
7. The trusted transit server performs AES decryption on each ciphertext data C_{Ti}, and then compares and filters according to the distance between its two-dimensional coordinates and the location of the LBS user to find the accurate nearest neighbor query result.
8. Return the accurate nearest neighbor query result to the mobile intelligent terminal.
9. Then the precision processing algorithm ends.

Algorithm 4: The ciphertext range query algorithm
Input: the index values range (C_{Ha}, C_{Hb}) of query
Output: all elements of the index values range (C_{Ha},C_{Hb})
1. Take C_{Ha} and C_{Hb} as the index values respectively, and then call Algorithm 2 to query in the ELT table.
2. If the index value C_{Ha} is in the ELT table, begins the query from left to right with the index value C_{Ha}, and ends the query when the key of the element is greater than C_{Hb}. If the index value C_{Ha} isn't in the ELT table, begin the query from left to right with the key of the element which is smaller than but closest to C_{Ha}, and ends the query when the key of the element is greater than C_{Hb}.
3. Return all elements in the index values range (C_{Ha}, C_{Hb}) to the trusted transit server.
4. Then the ciphertext range query of the POIs in the ELT table is over.

6. Performance Analysis of Algorithms

In order to analyze the performance of the CNNQBSH privacy protection algorithm proposed in this paper, this section will introduce its correctness, security and time complexity respectively.

6.1 Correctness Analysis of Algorithms

In order to prove the correctness of CNNQBSH algorithm, it can be verified from the following two aspects: (1) the necessity of precision processing about the approximate nearest neighbor query result; (2) whether the result of the precision processing is the actual query result. The above two aspects will be analyzed below.

(1) First of all, verify the necessity of precision processing about the approximate nearest neighbor query result. As shown in **Fig. 3**, when the LBS user sends the nearest neighbor query request in the location Q, and the query result obtained by using the approximate nearest neighbor query algorithm proposed in this paper is A or B, however its actual nearest neighbor POI is C. It can be seen that the query result obtained after Hilbert space transformation may not be the accurate query result. The main reason for this to happen is that there are two shortcomings when the two-dimensional space coordinates are transformed by the N-order Hilbert space curve: ①Because the Hilbert space is a U-shaped space, and this will cause $(2^N-1)*(2^N-1)$ grids to lose one edge, so that some neighbor grids in two-dimensional space will be lost, and the Hilbert values of POIs are quite different. ②The four nearest neighbor directions of the POI in the original two-dimensional space have only two nearest neighbor directions in the Hilbert transformation space, so the nearest neighbor query result in the Hilbert transformation space is affected. It can be seen that it is necessary to perform corresponding precision processing in order to obtain the accurate nearest

neighbor query result.

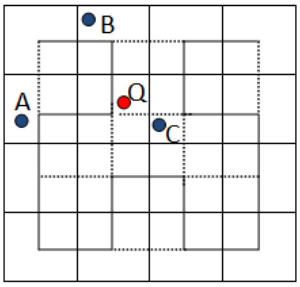


Fig. 3. Schematic diagram of inaccurate approximate result

（2）The query result obtained after the precision processing is the accurate nearest neighbor query result desired by the LBS user. As shown in **Fig. 4**, when the LBS user is located at Q, the query result obtained by calling the approximate nearest neighbor query algorithm proposed in this paper is A, but the accurate query result is C in fact.

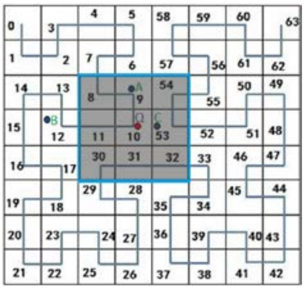


Fig. 4. The necessary of precision processing

In order to obtain accurate nearest neighbor query result, the precision processing algorithm proposed in this paper will be called. That is, firstly calculate the distance d between A and Q, and calculate the number n of grids covered by d, and then create a square region containing (2n+1)*(2n+1) grids with the Q as the center, as shown in the shaded region in **Fig. 4**. Obviously, if there is another POI D which is closer to Q than A, the shaded region will definitely contain D. As shown in **Fig. 4**, the POI C which is the accurate nearest neighbor POI of the LBS user, but not the query result of the approximate nearest neighbor query algorithm. It can be seen that the precision processing of the approximate nearest neighbor query result is necessary and correct.

6.2 Security Analysis of Algorithms

In order to solve the shortcomings of the traditional ciphertext nearest neighbor query privacy protection algorithm about the usage of the double Hilbert curves, namely the high system overhead and inaccurate query result in some special cases, the CNNQBSH algorithm is proposed. When the attacker attacks this algorithm, its biggest advantage $\text{Adv}_{\text{CNNQBSH}}$(Attacker) is as follows:

$$Adv_{CNNQBSH}(Attacker) = Pr[Attacker^{CNNQBSH} = 1] - Pr[Attacker^{OPES} = 1] \qquad (2)$$

Here, Pr[Attacker^{OPES}=1] is the probability of attacker successfully breaking the order preserving encryption scheme proposed by Boldyreva et al., which has been proved to have POPF-CCA security. Since the security of CNNQBSH algorithm mainly relies on the LF pseudo-random function which is achieved through hypergeometric distribution sampling [15], so it can be seen that the security of the CNNQBSH algorithm is at least POPF-CCA. The CNNQBSH algorithm also uses the Hilbert space transformation operation except using the OPES scheme, the higher security can be achieved as long as the parameters such as its starting point, curve direction, and transformation order being kept secret. So $Adv_{CNNQBSH}(Attacker)$ which is the advantage of an attacker successfully attacking the CNNQBSH algorithm can be completely ignored comparing to the POPF-CCA. By inductive proof, it can be seen that the CNNQBSH algorithm can achieve the highest IND-OCPA security.

Proof: From the security of the OPES scheme, it can at least achieve POPF-CCA security. The reason why it can’t have higher security is that it still has one shortcoming, namely, it will discloses the equality and the distance of data in the continuous plaintext space. Based on this, Boldyreva et al. solved the above shortcoming through MMPH (monotone minimal perfect hashing, MMPH) [16] and modular OPES, and achieved IND-OCPA security. In the CNNQBSH algorithm, because its plaintext space is a two-dimensional space that can’t be sorted, and its ciphertext is a one-dimensional value converted from the two-dimensional coordinates through Hilbert space transformation and then encrypted by OPES. Although it can maintain the lexicographical order of the plaintext numbers, but can’t reflect the equality and distance of the two-dimensional space because the value obtained in the Hilbert transformation space is one-dimensional value. So the CNNQBSH algorithm can also achieve the same security level as the monotonic minimum perfect hashing or the modular OPES, namely, the IND-OCPA security.

6.3 Complexity Analysis of Algorithms

This paper proposes an LBS privacy protection algorithm that supports ciphertext computation, namely CNNQBSH algorithm. It can be seen from the above analysis that CNNQBSH algorithm uses the order preserving encryption algorithm to encrypt the points of interest transformed by Hilbert curve, and then stores the encrypted points of interest in the ELT table in order. Therefore, when perform the single point query algorithm and the range query algorithm on the ELT table, it actually uses a halved query which’s time complexity is $O(\log_2 n)$. In addition, the precision processing algorithm is called in the CNNQBSH privacy protection algorithm, and its time complexity is $O(1)$. It can be seen that the time complexity of the approximate nearest neighbor query algorithm of the CNNQBSH algorithm is $O(\log_2 n)$, and the time complexity of getting the final exact nearest neighbor query result is $O(\log_2{}^2 n)$.

7. Experimental results and analysis

Since the difference in system overhead between single Hilbert curve and double Hilbert curve is apparent, it will no longer be verified here. In order to verify the accuracy and efficiency of the query results about CNNQBSH algorithm, this section will use the currently commonly used SHCBE (Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy, here it is called Single Hilbert ciphertext blind evaluation, SHCBE) privacy protection algorithm [9] and BEBLT (Blind evaluation of location based queries using space transformation to preserve location privacy, BEBLT) privacy protection algorithm [10] to compare with the CNNQBSH algorithm proposed in this paper by 5 experiments which run on POI database containing 5000 points of interest. These 5 experiments here are 5 random experiments, namely, 5 nearest neighbor query requests are sent to find the nearest neighbor hospital nearby the current location of the LBS user in the given region. The accuracy of their query results are shown in **Table 2**, the comparison of the approximate nearest neighbor query response time of those three algorithms are shown in **Fig. 5**, and the comparison of the response time of getting the final result in the three algorithms are shown in **Fig. 6**.

Table 2. The accuracy comparison of their query results

Algorithm	Exp1	Exp2	Exp3	Exp4	Exp5
SHCBE algorithm	Yes	No	No	Yes	No
BEBLT algorithm	No	Yes	Yes	Yes	Yes
CNNQBSH algorithm	Yes	Yes	Yes	Yes	Yes

It can be seen from **Table 2** that the accuracy of the query results about CNNQBSH algorithm is the best in the three ciphertext nearest neighbor query privacy protection algorithms, and the results all are accurate in five experiments. Here, the accuracy of SHCBE privacy protection algorithm is the worst. The main reason for this situation is that the SHCBE algorithm adopts a single Hilbert curve, so it can’t consider the nearest neighbor points of interest of the LBS user in the four directions of up, down, left, and right respectively. Although the BEBLT algorithm has considered the nearest neighbor points of interest of LBS users in four directions, it can't obtain accurate nearest neighbor query results in some special cases, as shown in **Fig. 1**. Therefore, the query results of its multiple times nearest neighbor queries may have the inaccurate condition. Although CNNQBSH algorithm adopts a single Hilbert curve, it performs precision processing on the approximate query results. If none of the points of interest in the approximate nearest neighbor result set is its nearest neighbor points of interest, there will definitely exist points of interest closer to the LBS user than the approximate nearest neighbor query result in the shaded region as shown in **Fig. 4**. Therefore, the nearest neighbor point of interest can be found through the precise processing algorithm proposed in this paper. When the nearest neighbor point of interest exists in the approximate nearest neighbor query result set, the nearest neighbor point of interest can also be found by using the precise processing algorithm proposed in this paper. Therefore, the final query results of the CNNQBSH algorithm are certainly accurate.

It can be seen from **Fig. 5** that the response time of the approximate nearest neighbor query result of CNNQBSH algorithm is optimal, the SHCBE algorithm and the BEBLT algorithm have roughly the same response time, but a little worse than the CNNQBSH algorithm.

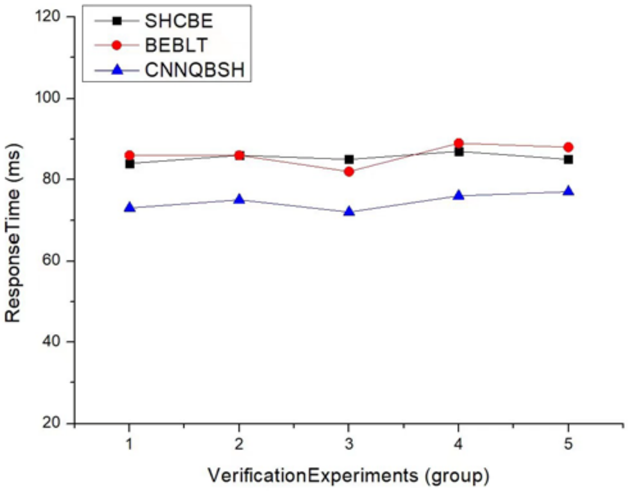


Fig. 5. Comparison of the response time of the approximate result

The main reason of the above situation is that the time complexity of the approximate nearest neighbor query algorithm of the SHCBE algorithm and the BEBLT algorithm are both $O(n)$, but the time complexity of the approximate nearest neighbor query algorithm of CNNQBSH algorithm is $O(\log_2 n)$. So the response time of the approximate nearest neighbor query result of the SHCBE algorithm and the BEBLT algorithm is slower than the CNNQBSH algorithm, and the above experimental results are basically consistent with the actual situation.

From the above analysis and related literature, It is obviously that the time complexity of getting the final query result in the CNNQBSH algorithm is $O(\log_2{}^2 n)$, the time complexity of getting the final query result in the BEBLT algorithm is $O(n \cdot \log \mathrm{n})$, and the time complexity of getting the final query result in the SHCBE algorithm is $O(n)$. However, it can be seen from **Fig. 6** that the response time of the CNNQBSH algorithm to obtain the final query result is slower than that of the BEBLT algorithm. The main reason for this is that the precision processing in the CNNQBSH algorithm is a complex dynamic process, and there are some dynamic operations in the whole process, such as AES decryption, calculating the distance, generating runs and so on. Whether these dynamic operations are performed or not will vary depending on the approximate nearest neighbor query result set, so the time complexity cannot be accurately estimated, but only a rough estimate can be made, so the response time to obtain the final query result is slower than BEBLT algorithm here. In addition, the response time of the BEBLT algorithm and the SHCBE algorithm is completely in line with its time complexity.

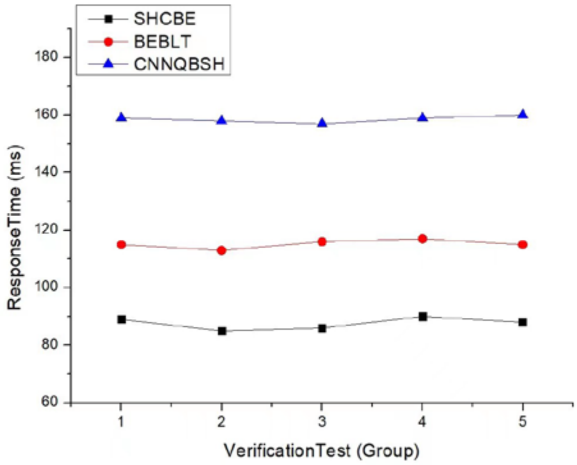


Fig. 6. Comparison of the response time of the final result

In this paper, the CNNQBSH algorithm, the BEBLT algorithm, and the SHCBE algorithm are applied to the nearest neighbor query request to find the nearest hospital to the current LBS user's location, and randomly repeat 5 groups of experiments. In summary, CNNQBSH algorithm optimizes the system overhead of the traditional ciphertext nearest neighbor query privacy protection algorithm by using a single Hilbert curve, and it improves the accuracy of the nearest neighbor query results by the precise processing algorithm proposed in this paper. However, the response time for obtaining the final result is related to the specific data in the approximate nearest neighbor query result set, and different approximate nearest neighbor query result sets needs different operations. So the response time of obtaining the final query result in the CNNQBSH algorithm is slower than that of traditional privacy-preserving algorithms which have poor time complexity. Nevertheless, the response time of the CNNQBSH privacy protection algorithm to obtain the final result is acceptable, so the CNNQBSH privacy protection algorithm is feasible.

8. Conclusion

In order to find the nearest neighbor hospital to the current location of LBS user and to solve the problems existing in traditional algorithms, this paper proposes the CNNQBSH privacy protection algorithm. These algorithms are suitable for all practical applications that require nearest neighbor query and privacy protection. The CNNQBSH algorithm is based on a single Hilbert curve and an ELT table, and it contains three sub-algorithms, namely the ciphertext single point query algorithm, the ciphertext range query algorithm and the precision processing algorithm. The algorithm also uses Hilbert space transformation, order preserving encryption scheme and AES encryption algorithm and so on. Here, the Hilbert space transformation is mainly used to convert the two-dimensional coordinates in the original region into the one-dimensional coordinates in the Hilbert transformation space, so as to preserve the proximity of the two-dimensional and reduce dimensionality space. The order preserving encryption scheme is embedded in the CNNQBSH algorithm to encrypt the one-dimensional coordinates obtained by Hilbert space transformation, so as to support LBS users to perform ciphertext queries on the ciphertext database of the LBS server. The AES encryption algorithm is mainly used to encrypt textual data that does not support ciphertext operations, such as the identity information of the LBS user, the time when the query was initiated, the query content, and other textual descriptions. Since this paper solves the problem that the final nearest neighbor query result may be inaccurate in some special cases, it can be seen from the experimental results that the ciphertext nearest neighbor query results of the CNNQBSH algorithm are completely accurate, and it will optimize the system overhead because only a single Hilbert curve is used. However, since the CNNQBSH algorithm may perform some unpredictable operations which depend on its approximate nearest neighbor query result set, such as AES decryption, calculating the distance, generating runs and so on, its response time in obtaining the final query result is slower than that of the traditional privacy protection algorithms whose time complexity is actually worse. However, it is still acceptable, so the CNNQBSH algorithm has good theoretical and practical value.

References

[1] Cyberspace administration of china, “To protect data security, the network security review method is coming,” China Youth Jan. 05, 2022. Network [Online]. Available: https://feeds-drcn.cloud.huawei.com.cn/landingpage/latest?docid=10510674803129c3862f3f4219d9efbbc44ee86&to_app=hwbrowser&dy_scenario=sticky&tn=cc9e75d798880aacd17f00fe36ccb7ac3369f16421ad696952a51b55a9bcb3f5&share_to=link&channel=HW_TRENDING&ctype=news&appid=hwbrowser&cpid=666&r=CN

[2] D. Hilbert, “Uber die Stetige Abbildung einer Linie auf ein Flachenstuck,” *Mathematische Annalen*, vol.38, no.1, pp.459-460, Sep. 1891. Article(CrossRef Link)

[3] J.A. Orenstein, “Spatial query processing in an object-oriented database system,” *ACM SIGMOD Record*, vol.15, no.2, pp.326-336, Jun. 1986. Article(CrossRefLink)

[4] H.V. Jagadish, “Linear clustering of objects with multiple attributes,” in *Proc. of 1990 ACM SIGMOD International Conference on Management of Data*, Atlantic City, NJ, USA, pp. 332-342, 1990. Article (CrossRef Link)

[5] M. Bader, *Space-filling Curves*, NewYork, USA: Springer, 2012

[6] R. Agrawal, J. Kiernan and R. Srikant, “Order-preserving encryption for numeric data,” in *Proc. of 2004 ACM SIGMOD international conference on Management of data*, pp. 563-574, 2004. Article (CrossRef Link)

[7] A. Boldyreva, N. Chenette and Y. Lee, “Order-Preserving Symmetric Encryption,” in *Proc. of Advances in Cryptology-EUROCRYPT 2009*, vol.5479, no.1, pp. 224-241, Otc. 2009. Article(CrossRefLink)

[8] A. Boldyreva, N. Chenette, and A. O'Neill, “Order-Preserving Encryption Revisited: Improved Security Analysis and Alternative Solutions,” in *Proc. of the 31st Annual Cryptology Conference*, pp. 578-595, 2011. Article (CrossRef Link)

[9] A. Khoshgozaran and C. Shahabi, “Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy,” in *Proc. of Advances in Spatial and Temporal Databases, SSTD 2007*, vol.4605, no.1, pp.239-257, Jul. 2007. Article(CrossRefLink)

[10] A. Khoshgozaran, H. Shirani-Mehr and C. Shahabi, “Blind evaluation of location based queries using space transformation to preserve location privacy,” *Geoinformatica*, vol. 17, no.4, pp.599−634, Jan. 2013. Article(CrossRefLink)

[11] F. Tian, X. L. Gui and X. J. Zhang, “Privacy-preserving approach for outsourced spatial data based on POI distribution,” *Journal of Computer*, vol. 37, no.1, pp. 123-138, Jan. 2014.

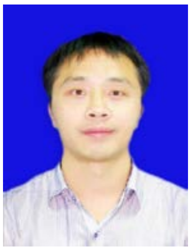
[12] C. L. Zhou, Y. H. Chen and H. Tian, “Location privacy and query privacy preserving method for K-nearest neighbor query in road networks,” *Journal of Software*, vol. 31, no.2, pp. 471−492, Feb. 2020.

[13] N. Shen, C. F. Jia and S. Liang, “Approach of location privacy protection based on order preserving encryption of the grid,” *Journal of Communications*, vol. 38, no.7, pp. 78-88, Jul. 2017. Articl(CrossRefLink)

[14] C. Faloutsos and S. Roseman, “Fractals for secondary key retrieval,” in *Proc. of the 8th Symposium on Principles of Database Systems*, 247–252, 1989. Article (CrossRef Link)

[15] D. P. Hu, *Decimal block encryption based on negative hypergeometric distribution*, Beijing, China: Tsinghua University Press, 2018

[16] D, Belazzougui, P, Boldi, R, Pagh, “Monotone minimal perfect hashing: searching a sorted table with o(1) accesses,” in *Proc. of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 785-794, 2009. Article (CrossRef Link)



Delin Tan is an associate professor of SiChuan Normal University, China. His research interests include the research of privacy protection, fully homomorphic encryption, cloud computing etc.

Huajun Wang is a professor of Chengdu University of Technology, China. His main research interests include the research of sensor networks, network security, embedded and so on.