ATLANTIS PRESS International Journal of Networked and Distributed Computing, Vol. 5, No. 3 (July 2017) 123–132

CSEP: Circular Shifting Encryption Protocols for Location Privacy Protection

Chen Di [1], Zhao Binglin [2], Li Hao [1], Zhou Shilei [1]

[1] Luoyang Electronic Equipment Test Center of China,
No.17 Zhou Shan Street
Luoyang, 471003, China
E-mail: chendi610@126.com, 13939942396@163.com, zhoushilei@163.com

[2] State Key Laboratory of Mathematical Engineering and Advanced Computing,
No.62 Science Avenue
Zhengzhou, 450001, China
E-mail: gzu_zhaobl@126.com

Abstract

Location Based Service (LBS) is gaining popularity. As one fundamental LBS service, range search returns all Point of Interests (POIs) within a user-specified range. However, people leave their location privacy at risks when using range search. How to provide a high-quality range search service while protecting users' location privacy is a challenging problem. Most existing approaches use space-filling curves and cloaked region method to provide privacy-preservation location services, but these methods cannot return the accurate results. In this paper, we propose a set of Circular Shifting Encryption Protocols (CSEP) based on homomorphism and circular shift for location privacy protection of range search. CSEP leverages homomorphism encryption to encrypt users' locations, and LBS servers compute distances on cyphertext. In this way, LBS server can return POIs within the specified range, while learning nothing about the user's real location. To accommodate the different query range and the private protection degree of users, we propose a circular shifting encryption method to reduce the redundancy and increase the degree of privacy protection. We implement a prototype of CSEP, and evaluate it with real POI set of a large-scale production LBS. Experimental results show that CSEP can provide reliable privacy protection and accurate range search, with reasonable compute overhead and communication overhead.

Keywords: location based service; location privacy; range search; circular shifting encryption

1. Introduction

With the widespread use of mobile devices and GPS navigation, Location Based Services (LBS) are becoming indispensable in our daily life. Users can enjoy the convenience provided by LBS by submitting LBS queries. However, as the information about users accumulated on the untrusted LBS servers, user's location information is at risks. Once leaked, the location data could reveal sensitive information of users, such as where they are at which time, what kind of queries they submit, and what they are doing, etc.[1] Thus, high attention should be paid to the location privacy issue.

Nearest neighbor search is one of the most basic services in LBS, and is the basis for other services such as route planning, proximate friends finding and so on. Range search is one of the most important modes of nearest neighbor search, which need to return very accurate results. The problem we are considering is to provide a reliable location privacy preserving range search method that guarantees accurate results.

In recent years, many approaches have been proposed. (1) Most existing approaches use spatial cloaking to hide the real location of a user [2-5]. In these approaches, an anonymization server transforms the location of a user to a cloaked region, and sends this region to the LBS server. Then, the LBS server returns

Copyright © 2017, the Authors. Published by Atlantis Press.
This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

all POIs within the proximity of the cloaked region. However, as the LBS does not know the exact location of the user, it cannot return the exact POIs within say *r* meters from the user. (2) Space transformation method converts the location information into another space representation[6-7], which has a certain transformation relationship with the original one, such as Hilbert space filling curve[6] and Moore curve[7]. However, these space-filling curve methods cannot return the exact nearest neighbors and the result may generate serious deviation in some cases. (3) Cryptographic transformation[6,8] methods provide privacy preserving by encryption, which can provide higher degree of location privacy protection, but needs client to take part in the encrypted distance computing and cannot support multiple neighbors query.

In this paper, we propose Circular Shifting Encryption Protocols (CSEP) for location privacy preserving, which do not need a third-party server. CSEP leverages the spatial cloaking technique and complements it with encryption method for improving precision. There is no third party in CSEP and we do not trust LBS server. To provide different privacy protection level, two protocols are designed in CSEP. Protocol 1 uses spatial cloaking method to hide user's exact location. Moreover, homomorphic encryption is used during the interaction process in Protocol1 to avoid monitoring from adversaries. Protocol 2 is proposed based on Protocol 1 to provide higher degree of privacy protection. Protocol 2 leverages circular shifting encryption for POIs in query results and corresponding POI information in LBS server. Thus Protocol 2 decreases the redundancy of POI information during transmission, and provides higher privacy protection degree as well.

We make the following contributions in this paper:

- We propose CSEP, a set of Circular Shifting Encryption Protocols to achieve reliable location privacy preserving mechanism based on spatial cloaking and homomorphic encryption.
- We implement a prototype of CSEP, and evaluate it using real POI dataset from the aspects of results accuracy, compute overhead and communication overhead etc.

The rest of the paper is organized as follows: Section 2 introduces some technical background of this paper. Section 3 introduces the main idea of CSEP and related definitions. Section 4 presents the design of two protocols in CSEP. Section 5 analyzes the privacy protection degree provided by CSEP. Section 6 evaluates CSEP with experiments and Section 7 concludes this paper.

2. Preliminary

This section first introduces a grid geospatial representation used in CSEP. Then introduces Paillier homomorphic encryption for encryption distance calculation and circular shifting.

2.1. Geospatial representation

In this paper, we adopt the traditional grid geospatial representation: Consider a two-dimensional geographic region, we split the region with 1 meter as a unit by latitude and longitude direction, respectively. We define the square grid unit with latitude index Hi and longitude index V_j as $<H_i, V_j>$ cell.

A POI can be represented by$<POI_{ID}, H, V, POI_{TYPE}, POI_{INFO}>$. Among them, POI_{ID} is the index that can uniquely identify the POI. H is the latitude value, V is the longitude value. POI_{TYPE} is the type information of the POI (e.g. hotel, scenic site, petrol filling station, shopping mall). POI_{INFO} is the associated POI information This fine-grained partitioning method can identify POIs accurately, thus can complete the geospatial calculation with high precision.

2.2. Paillier homomorphic encryption

Homomorphic Encryption (HE) allows direct addition and multiplication on cyphertexts while preserving decrypt ability. We choose Paillier's system[9] to provide homomorphic encryption of user's location which is simple and efficient. Paillier's cryptosystem is composed of three algorithms Key-Generate, Encrypt and Decrypt. The Paillier's cryptosystem satisfies the following homomorphic properties:

$$D(E_{r1}(m_1) \cdot E_{r2}(m_2) \bmod n^2) = m_1 + m_2$$
$$D(E_{r1}(m_1)^{m_2} \bmod n^2) = m_1 \times m_2$$

For the sake of simplicity, we use $E(m)$ instead of $E(m,r)$ in the remaining paper.

Before the implementation of CSEP, we generated encryption key $EK_u = (n, g)$ and decryption key $DK_u = (\lambda, \mu)$ for Paillier's cryptosystem on the client side. The corresponding public key EK_s and private key DK_s are generated and assigned to the server by a certain public key cipher system.

3. CSEP: Main Idea and Definitions

In this section, we introduce the main idea of CSEP and definitions. CSEP is a location preserving method using a circular shifting encryption method. There are two protocols in CSEP. Protocol1 combines spatial cloaking and homomorphism encryption to provide reliable location privacy protection. Protocol2, which provides higher degree of privacy protection, uses a circular shifting encryption method to reduce the redundancy of POI information by offsetting the POI_{ID} of the POI set and corresponding POI_{INFO} on the server side.

3.1. Main Idea of CSEP

CSEP consists of the client side *U* and the server side *S*, assuming the following conditions:

- *U* is credible. It will not disclose user's location, its decryption key and other relevant information initiatively; *U* has general computing ability to complete encryption, decryption and other tasks.
- *S* is not completely trustworthy. *S* has a strong computing ability, to complete database retrieval, homomorphic encryption and other tasks. The POI-table, which stores POI information, is maintained in *S* and contains 5 attributes: <POI_{ID}, *H*, *V*, POI_{TYPE}, POI_{INFO}>, the meaning of each attribute is described in Section 2.1.
- Both *U* and *S* will follow the protocols.

The main idea of CSEP is based on spatial cloaking method, as shown in Fig.1. Suppose that user *U* takes the query point *Q* (H_Q, V_Q) as the center, and queries all the POIs of type *t* in the range of *r* meters. First, *U* specifies a set of parameters to generate cloaked region rectangle R_2 randomly. Secondly, *S* selects all the POIs that satisfy the query requirements according to the cloaked region R_2, calculates distances between POIs in the candidate POI and query point *Q* using homomorphism encryption (on cyphertext), and sends the encrypted distances and corresponding POI information to *U*. Finally, *U* decrypts the cyphertext and filters the POIs that meet the requirement.

In order to control the overhead of CSEP better, and meet the different privacy protection requirements of users, CSEP provides an enhanced protocol. In the enhanced protocol, indexes of POI result set calculated in the client side and POI table maintained in the server side are shifted, and the server *S* can return all the POI information of the POIs that meet the restriction, without knowing the result POI set. Query location and POI information of the user will be better protected on the client side using a circular shifting method, and the transfer redundancy will be reduced as well.

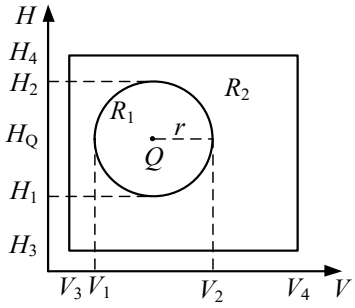


Fig. 1. Query range and Cloaked region

3.2. Definitions

Definition 1: Query range R_1. The circular range R_1 with the query point *Q* as the center, and the user-specified query distance *r* as the radius is defined as Query range R_1, as shown in Fig.1.

Definition 2: Candidate rectangleR_2. The rectangle R_2 (upper right coordinate: (V_4, H_4); lower left coordinate: (V_3, H_3)) containing the query range R_1, is defined as Candidate rectangle R_2, as shown in Fig.1.

4. CSEP: Detailed Design of Protocols

In this section, we introduce the detailed design of two protocols of CSEP. The detailed steps are as follows:

4.1. Protocol 1

Protocol 1 is a baseline protocol as a building block for Protocol 2. The client-server interaction flow of Protocol 1 is shown in Fig.2. The detailed steps are as follows:

①*U*: Determines the boundary of cloaked region rectangle R_2, and encrypts the query position *Q*.

According to the privacy protection requirements and the query range *r*, the client *U* randomly determines H_4-H_3 and V_4-V_3, that is to generate the length and width of cloaked region R_2. Then *U* randomly generates the vertexes coordinates (H_4, H_3, V_4 and V_3) of R_2.The client side *U* calculates $E(1)$, $E(H_Q{}^2 + V_Q{}^2)$, $E(H_Q)$, $E(V_Q)$, then sends R_2 (H_4, H_3, V_4, V_3), $E(1)$, $E(H_Q{}^2 + V_Q{}^2)$, $E(H_Q)$, $E(V_Q)$ and user-specified query POI type *t* to the server *S* together.

②*S*: Determines the candidate POI and performs the encryption distance calculation.

After receiving the boundary value of the cloaked region R_2 and the queried POI type t, the server S retrieves for all the POIs whose belonging grids within or intersect with candidate rectangle R_2, and sorts the POI set that satisfies (POI_{TYPE}=t) as the Candidate POI set (*Candidate_POI*). Suppose there are m POIs in *Candidate_POI*, and the structure of each POI is<POI_{ID}, H, V, POI_{TYPE}, POI_{INFO}>. According to the received cyphertext $E(1)$, $E({H_Q}^2+{V_Q}^2)$, $E(H_Q)$, $E(V_Q)$, S calculates the square of the distances $E(dist_i)$ of each POI_i in *Candidate_POI* to the query point Q on cyphertext:

$$E({dist_i}^2) = E({H_Q}^2+{V_Q}^2)\times E(H_Q)^{-2H_i}\times E(V)^{-2V_i}\times E(1)^{{H_i}^2+{V_i}^2}$$
$$= \mathrm{E}({H_Q}^2+{V_Q}^2-2H_Qh_i-2V_QV_i+{H_i}^2+{V_i}^2)$$
$$= \mathrm{E}((\mathrm{H_Q}-\mathrm{H_i})^2+(\mathrm{V_Q}-\mathrm{V_i})^2)$$

S sends $\{POI_{ID_i}, E({dist_i}^2), POI_{INFO_i}\}$, $(1\le i\le m)$back to U.

③*U*: Performs decryption to obtain the results of the range search query.

U decrypts $E({dist_i}^2)$ in $\{POI_{ID_i}, E({dist_i}^2), POI_{INFO_i}\}$, $(1\le i\le m)$, gets ${dist_i}^2$:

$${dist_i}^2 = D\left(E({dist_i}^2)\right) = (H_Q-H_i)^2+(V_Q-V_i)^2$$

Obviously, ${dist_i}^2 = |Q-p_i|^2$ is the squared value of distance between POI_i∈*Candidate_POI* and the query point Q. If ${dist_i}^2 \le r^2$, POI_i is the POI that U expected to acquire.

Thus, U can get the result set:

$$\{POI_{ID_i}, POI_{INFOi}\}\ (1\le i\le m, {dist_i}^2\le r^2)$$

The time complexity of Protocol 1 is *O(n)*.

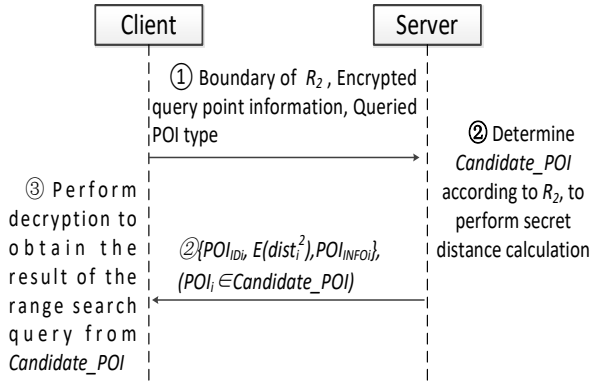


Fig. 2. Interaction of Protocol 1

4.2. Protocol 2

In Protocol 1, the POI set returned to U by S is

$$\{\mathrm{POI_{ID_i}}, \mathrm{E}(\mathrm{dist_i}^2), \mathrm{POI_{INFO_i}}\},\quad (1\le \mathrm{i}\le \mathrm{m})$$

U sorts for the results according to the user-specified query range. However, if the POI_{INFO} of the *Candidate_POI* is intercepted by a third-party attacker during transmission, user's location and query privacy will expose to the attacker. Moreover, when the user needs higher privacy protection degree, the area of the cloaked region will increase, so as the number of POIs in *Candidate_POI*. If the server sends all the POI_{INFO} of POIs in *Candidate_POI*, a lot of redundant information will be introduced during the transmission.

On the basis of Protocol 1, we give another alternative. Using a circular shift encryption method to shift indexes of POI result set calculated in the client side and POI table maintained in the server side, the server S can return all the POI information of the POIs that meet the restriction, without knowing the result POI set.

The client-server interaction flow of Protocol2 is shown in Fig.3, the detailed steps are as follows:

There are 6 steps in Protocol 2, the first 3 steps are essentially the same as the previous steps in the protocol 1, except in step 2: after the server S performs encryption distance calculation, only sends $\{POI_{ID_i}, E(dist_i)\}$, $(1\le i\le m)$ to the user U which gets$\{POI_{ID_i}, dist_i\}$, $(1\le i\le m)$ after decryption and filtering. Then, the server and the client interact again through the following steps to return the POI_{INFO} that satisfies the condition.

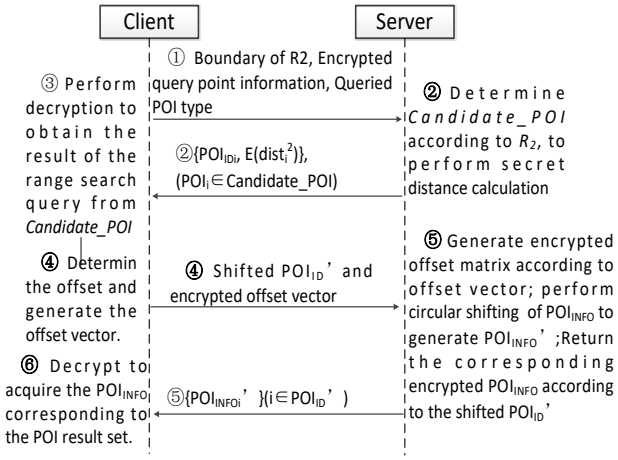


Fig. 3. Interaction of Protocol 2

④*U*: Determine the offset and generate the offset vector.

a. *U* generates an offset *s*, an offset vector *P*[*1*×*m*]. The *s*th element in *P* is *1*, and the other elements are *0*.

b. *U* selects *m* random numbers $\{r_0, r_1, r_2, \cdots, r_{m-1}\}$ to complete encryption of each element in offset vector *P* to generate a *1*×*m* encrypted offset vector *E*(*P*).

$$E(P) = [\mathrm{E}_{\mathrm{r}_0}(0) \quad \mathrm{E}_{\mathrm{r}_1}(0) \quad \cdots \mathrm{E}_{\mathrm{r}_{s-1}}(1) \quad \ldots \quad \mathrm{E}_{\mathrm{r}_{m-1}}(0)]$$

c. According to the result POI set $\{POI_{ID_i}\}$ $(1 \le i \le m, dist_i \le r)$, *U* generates a new set $T = \{POI_{ID_i}'\}(1 \le i \le m, dist_i \le r)$, $POI'_{IDi} = (POI_{ID_i} + s) \bmod m$.

d. *U* sends the encrypted offset vector *E(P)* and the indexes of POIs in *T* corresponding with those in result POI set.

⑤*S*: Circular shifted encryption of POI_{INFO}

After receiving the encrypted offset vector *E(P)* and the set *T*, *S* first generates an *m*×*m* offset matrix *M*:

$$\mathrm{M} = \begin{bmatrix} \mathrm{E}_{\mathrm{r}_0}(0) & \mathrm{E}_{\mathrm{r}_1}(0) & \cdots \mathrm{E}_{\mathrm{r}_{s-1}}(1) & \cdots & \mathrm{E}_{\mathrm{r}_{m-1}}(0) \\ \mathrm{E}_{\mathrm{r}_{m-1}}(0) & \mathrm{E}_{\mathrm{r}_0}(0) & \cdots \mathrm{E}_{\mathrm{r}_{s-1}}(1) & \cdots & \mathrm{E}_{\mathrm{r}_{m-2}}(0) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathrm{E}_{\mathrm{r}_1}(0) & \mathrm{E}_{\mathrm{r}_2}(0) & \cdots \mathrm{E}_{\mathrm{r}_{s-1}}(1) & \cdots & \mathrm{E}_{\mathrm{r}_0}(0) \end{bmatrix}$$

S treats the POI_{INFO} column in table *Candidat_POI* as a *m*×*1* vector:

$$POI_{INFO} = [I_0 \quad I_1 \quad \cdots \quad I_{s-l} \cdots \ I_{m-l}]^{\tau}$$

The offset matrix *M* is used to make the circular shifted encryption for the vector POI_{INFO}, the shifted vector is denoted as POI_{INFO}':

$$POI'_{INFO} = M * POI_{INFO}$$

$$= \begin{bmatrix} \mathrm{E}_{\mathrm{r}_0}(0)^{I_0} \times \cdots \times \mathrm{E}_{\mathrm{r}_{s-1}}(1)^{I_{s-1}} \times \cdots \mathrm{E}_{\mathrm{r}_{m-1}}(0)^{I_{m-1}} \\ \mathrm{E}_{\mathrm{r}_{m-1}}(0)^{I_0} \times \cdots \times \mathrm{E}_{\mathrm{r}_{s-1}}(1)^{I_s} \times \cdots \mathrm{E}_{\mathrm{r}_{m-1}}(0)^{I_{m-1}} \\ \cdots \cdots \cdots \\ \mathrm{E}_{\mathrm{r}_1}(0)^{I_0} \times \cdots \times \mathrm{E}_{\mathrm{r}_{s-1}}(1)^{I_{s+m-2}} \times \cdots \mathrm{E}_{\mathrm{r}_{m-1}}(0)^{I_{m-1}} \end{bmatrix}$$

The operator "*" represents the Paillier matrix multiplication. Since *r* is generated randomly, when $k \neq l$, $E_k(a) \neq E_l(a)$, $(a = 0 \text{ or } a = 1)$.Therefore, it is very difficult to get the offset *s*. Moreover, POI_{ID} in set *T* have been shifted, so *S* can not get the results calculated in *U*, and get the corresponding POI_{INFO}' in the result POI set according to the shifted indexes in *T*. The *i*th element in vector POI_{INFO}' is denoted as $POI'_{INFO\,i}$ $(1 \le \mathrm{i} \le \mathrm{m})$.*S* sends the set $\{POI'_{INFO\,i}\}$ $(i \in T)$ back to *U*.

⑥*U*: Decrypts to acquire the POI_{INFO} corresponding to the POI result set.

The indexes that corresponding to the result POIs can be obtained by decrypting the set $\{POI'_{INFO\,i}\}$ $(i \in T)$ on the client s ide.

The time complexity of circular shifted encryption algorithm in Protocol 2 is *O(n)*. In order to better illustrate the interaction progress in Protocol 2, a simple example is given below:

Example: *S* generates a candidate POI set *Candidate_POI* according to the boundary of candidate rectangle R_2, the encrypted query point location information and the query POI type. Then *S* sends the candidate POI_{ID} and the encrypted distances to *U*.

$$Candidate\_POI = [P_0 \quad P_1 \quad P_2 \ P_3 \ P_4]$$

U filters *Candidate_POI* to derive POI indexes of the result set:

$$Result\ POI_{ID} = \{P_1, P_2, P_3\}$$

generates offset vector *P* (offset *s* = 3):

$$P = [0, 0, 1, 0, 0]$$

encrypts the offset vector:

$$E(P) = [\mathrm{E}_{\mathrm{r}_0}(0) \quad \mathrm{E}_{\mathrm{r}_1}(0) \quad \mathrm{E}_{\mathrm{r}_2}(1) \quad \mathrm{E}_{\mathrm{r}_3}(0) \quad \mathrm{E}_{\mathrm{r}_4}(0)]$$

The set of POI indexes of offset candidate POI set:

$$T = [P_3 \quad P_4 \quad P_0 \ P_1 \ P_2]$$

The corresponding indexes of result POI in the shifted *T*:

$$Result\ POI_{ID}' = \{P_4, P_0, P_1\}$$

S generates an offset matrix *M* according to the received encrypted offset vector:

$$\mathrm{M} = \begin{bmatrix} \mathrm{E}_{\mathrm{r}_0}(0) & \mathrm{E}_{\mathrm{r}_1}(0) & \mathrm{E}_{\mathrm{r}2}(1)\mathrm{E}_{\mathrm{r}_3}(0) & \mathrm{E}_{\mathrm{r}_4}(0) \\ \mathrm{E}_{\mathrm{r}_4}(0) & \mathrm{E}_{\mathrm{r}_0}(0) & \mathrm{E}_{\mathrm{r}_1}(0)\mathrm{E}_{\mathrm{r}2}(1) & \mathrm{E}_{\mathrm{r}_3}(0) \\ \mathrm{E}_{\mathrm{r}_3}(0) & \mathrm{E}_{\mathrm{r}_4}(0) & \mathrm{E}_{\mathrm{r}_0}(0)\mathrm{E}_{\mathrm{r}_1}(0) & \mathrm{E}_{\mathrm{r}2}(1) \\ \mathrm{E}_{\mathrm{r}2}(1) & \mathrm{E}_{\mathrm{r}_3}(0) & \mathrm{E}_{\mathrm{r}_4}(0)\mathrm{E}_{\mathrm{r}_0}(0) & \mathrm{E}_{\mathrm{r}_1}(0) \\ \mathrm{E}_{\mathrm{r}_1}(0) & \mathrm{E}_{\mathrm{r}2}(1) & \mathrm{E}_{\mathrm{r}_3}(0)\mathrm{E}_{\mathrm{r}_4}(0) & \mathrm{E}_{\mathrm{r}_0}(0) \end{bmatrix}$$

The server multiplies M with the stored candidate POI set:

$$POI'_{INFO} = M * POI_{INFO}$$

$$= \begin{bmatrix} \mathrm{E}_{\mathrm{r}_0}(0)^{I_0} \times \mathrm{E}_{\mathrm{r}_1}(0)^{I_1} \times \mathrm{E}_{\mathrm{r}_2}(1)^{I_2} \times \mathrm{E}_{\mathrm{r}_3}(0)^{I_3} \times \mathrm{E}_{\mathrm{r}_4}(1)^{I_4} \\ \mathrm{E}_{\mathrm{r}_4}(0)^{I_0} \times \mathrm{E}_{\mathrm{r}_0}(0)^{I_1} \times \mathrm{E}_{\mathrm{r}_1}(0)^{I_2} \times \mathrm{E}_{\mathrm{r}_2}(1)^{I_3} \times \mathrm{E}_{\mathrm{r}_3}(0)^{I_4} \\ \mathrm{E}_{\mathrm{r}_3}(0)^{I_0} \times \mathrm{E}_{\mathrm{r}_4}(0)^{I_1} \times \mathrm{E}_{\mathrm{r}_0}(0)^{I_2} \times \mathrm{E}_{\mathrm{r}_1}(0)^{I_3} \times \mathrm{E}_{\mathrm{r}_2}(1)^{I_4} \\ \mathrm{E}_{\mathrm{r}_2}(1)^{I_0} \times \mathrm{E}_{\mathrm{r}_3}(0)^{I_1} \times \mathrm{E}_{\mathrm{r}_4}(1)^{I_2} \times \mathrm{E}_{\mathrm{r}_0}(0)^{I_3} \times \mathrm{E}_{\mathrm{r}_1}(0)^{I_4} \\ \mathrm{E}_{\mathrm{r}_1}(0)^{I_0} \times \mathrm{E}_{\mathrm{r}_2}(1)^{I_1} \times \mathrm{E}_{\mathrm{r}_3}(0)^{I_2} \times \mathrm{E}_{\mathrm{r}_4}(0)^{I_3} \times \mathrm{E}_{\mathrm{r}_0}(0)^{I_4} \end{bmatrix}$$

According to the shifted *Result* POI_{ID} from *U*, the number 0, 1 and 4 elements corresponding to POI'_{INFO} are extracted and sent to *U*. *U* performs decryption according to the corresponding random number, and

finally obtains the corresponding POI_{INFO} of the result set $\{P_1, P_2, P_3\}$.

In practice, the two protocols can be combined into an adaptive protocol, according to the data size of the POI information to select different protocol branches, in order to achieve a balance in performance and the privacy protection degree.

CSEP has the following major advantages over existing methods:

- *High precision of returned results.* As the LBS server calculates the distances from candidate POIs from user's location, the user can filter out POIs for exact results, without distance calculation.
- *Location privacy protection.* LBS operates on the cyphertext of user's location when calculating distances, and indexes corresponding to POI have been shifted. Thus, CSEP is provable secure.

5. Privacy Analysis

In this section, we analyze privacy metric of CSEP. Privacy protection degree of CSEP is controlled by client side, and performs as a controllable variable in experiments.

Location privacy metrics can be measured by disclosure risks to the adversary. Disclosure risk represents the probability that an attacker may know about the user's location and other sensitive information according to the public information and other background knowledge. Typically, the more background knowledge about the public information, the greater the risk of disclosure. We use D to denote public information, D_k to denote disclosing D using background knowledges K, so $r(D, K)$ represents as $r(D, K) = P_r(D_k)$.

In CSEP, both protocols using homomorphic encryption can provide reliable privacy protection against ciphertext-only attack, since all information obtained by third-party attackers are ciphertext. Protocol 2 provides higher degree of privacy protection, uses a circular shifting encryption method. Attackers can hardly acquire valid location information unless crack Parllier's system, that is to crack homomorphic encryption scheme and RSA scheme as well, which is hard to achieve.

Even the attacker cracks Parller's system, or gets the key by some special methods, his probability of getting the exact location of the user is still small. Since the user only sends a cloak region instead of query point, which can be an arbitrary point to the server. Assuming that the protocol is open, that is, the attacker knows the rules of the agreement and gets to know the cloaked region, can only guess the possible location of the query point in the rectangular area shown in Fig.1. The area of the grid is S_0, the area of the candidate rectangle R_2 is S_{R2}. The successful probability of the attacker is:

$$\mathrm{r(D,K)} = P_r(D_k) = \frac{S_0}{S_{R2}} = \frac{1}{(H_4 - H_3 - 2r) \times (V_4 - V_3 - 2r)}$$

Consequently, we can achieve custom settings for user's privacy requirements by controlling the size of cloaked rectangle, which varies as independent variables in our experiment.

6. Evaluation

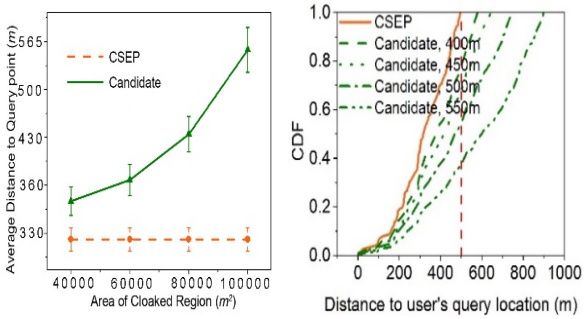
In this section, we evaluate CSEP with experimental results, based on real data from a large-scale production LBS. The metrics we considered include result precision, response time and communication overhead.

6.1. Setup

In the experiment, we have two physical hosts, one as the client and one as the server. The server is configured with Intel i7 CPU (3.40GHz) and 16GB DDR3 memory, running Windows 8 64bit OS. The client is configured with Intel i5 CPU (3.1GHz) and 4GB DDR3 memory, also running Windows 8 64bit OS. The length of Paillier's public key is 128 bits; Our test data set was collected from a large scale production LBS (from May to August 2015, containing 3,241,177 entries from more than 2,529,445 users).

6.2. Precision of results

The standard to measure precision is whether the returned POI set contains all POIs within the query range. Spatial cloaking method [2–5] uses cloaking region as input to request LBS to get approximate nearest neighbors (approximate NN for short), which cannot guarantee the accuracy of the results. The size of cloaking region influences the returned approximate nearest results to a large extent. In CSEP, candidate POIs are generated only according to cloak region by the server, just similar with other approximate NN method, so we regard candidate POI results as reference object for the final results.



(a) (b)

Fig. 4. Comparison between CSEP results and candidate results.(a) Average distance to query location; (b) CDF of distance to query location

We set query range as *500m* and vary the area of candidate rectangle $R_2$from $4 \times 10^4\, m^2$ to $1 \times 10^5\, m^2$, analyze cumulated distribution and calculate the average of distance between user's query point Q and each POI in the results. As shown in Fig. 4, the average distances between Q and each POI in CSEP results remain *316m* as the area of R_2 grows according to Fig. 4(a), all of results are within *500m* query range according to Fig. 4(b). In Fig. 4(a), average distances between Q and each POI in candidate results increases from *357m* to *562m* as the area of R_2 increases from $4 \times 10^4\, m^2$ to $1 \times 10^5 m^2$, and distributed broader as shown in Fig. 4(b).

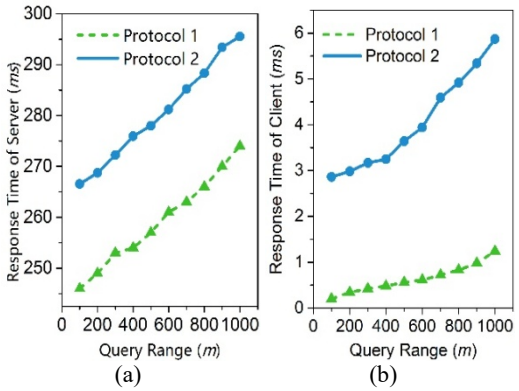
The result generated according to cloak region always contains redundant POIs outside the query range, and the scope of results becomes broader as the cloak region grows. However, the change of cloak region cannot influence the results of CSEP, since the result set are selected as all POIs within query range in CSEP, which does not contain any omissions and outliers.

6.3. Response Time

We evaluate the response time of 2 protocols for CSEP in this subsection, and study the correlation between computational overhead and two adjustable variables, the size of candidate rectangle and query range.

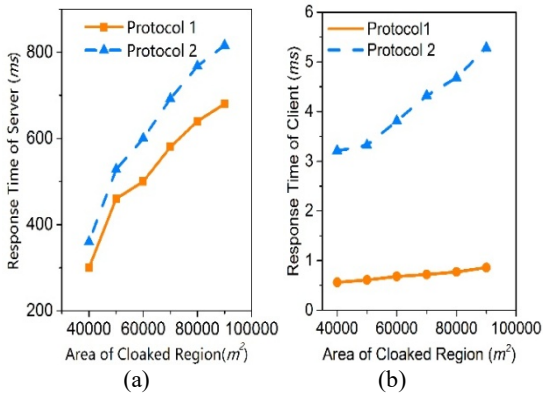
The response time for different query ranges is shown in Fig. 5, where we fix the area of rectangle R_2 as $4 \times 10^4\, m^2$, and vary the query range from 100*m* to 1000*m*. Fig. 5(a) shows the comparison of response time on the server side between protocol 1 and protocol 2. As the radius of query range increases from 100*m* to 1000*m*, response time of protocol 1 increases from 242*ms* to 283*ms* and response time of protocol 2 increases from 268*ms* to 298*ms*. On the client side, as shown in Fig. 5(b), computational overhead of both protocols is less than 10*ms*, which follows the similar positive correlation between query range and response time. Since more POIs need to be calculated and searched as the size of query range becomes larger.

The response time of different cloak regions R_2 is shown in Fig. 6, where we fix the query range as *1000m*, and vary the area of R_2 from $4 \times 10^4 m^2$ to $1 \times 10^5 m^2$. Fig. 6(a) shows the comparison of response time on the server side between Protocol 1 and Protocol 2. As the area of R_2 increases from $4 \times 10^4 m^2$ to $1 \times 10^5 m^2$, the response time of Protocol 1 increases from *315ms* to *643ms* and the response time of Protocol 2 increases from *365ms* to *812ms*. On the client side, as shown in Fig. 6(b), computational overhead of both protocols is less than *10ms*, which follows the same positive correlation between query range and the response time with the server side. Since when cloaked region becomes larger, more POIs are appended into candidate



(a) (b)

Fig. 5. The server- and client-side response time, variable query range, $S_{R2}= 4 \times 10^4 m^2$ (a) Server; (b) Client.



(a) (b)

Fig. 6. The server- and client-side processing time, variable cloaking region, r = 1000. (a) Server; (b) Client.

POI set, and more distance calculation operations need to be handled consequently. As another encryption and decryption operation is carried out in Protocol 2, the cost of computing of Protocol 2 is greater than Protocol 1.

6.4. Communication Overhead

In this subsection, we evaluate the communication overhead incurred by our two protocols, in terms of bytes transmitted per query. The mainly communication overhead in a query using Protocol 1 consists of $\{POI_{ID_i}, E(dist_i{}^2), POI_{INFO_i}\}$ in candidate POI set. In Protocol 2, $\{POI_{ID_i}, E(dist_i)\}$in candidate POI set and the result set $\{POI'_{INFO_i}\}$ $(i \in T)$ selected from the circular shifted matrix POI_{INFO}' compose the communication overhead per query. First, we fix the area of rectangle cloaking region as $6 \times 10^4\, m^2$, and vary the length of queried range. As shown in Fig.7(a), that both protocols have a larger communication overhead when the length of queried range increases. Since Protocol 2 do not need to transmit redundant candidate POI information, only need to return encrypted POI information of the result set, the communication cost of Protocol 2 is less than that of Protocol 1. Then, we fix the query range as 1000m, and vary the area of cloaked region. Fig. 7(b) shows that both protocols have a larger communication overhead when the length of queried range increases, and still Protocol 2 has a smaller bandwidth.

6.5. Summary

From the experimental results above, we conclude that:

(1) The results contain all POIs within query range, without any outlier and omission. The computation overhead of the two protocols are in millisecond level on a commodity server, and their communication overhead are around 2 KB, for queried range 1000m and cloaking size $6 \times 10^4 m^2$.

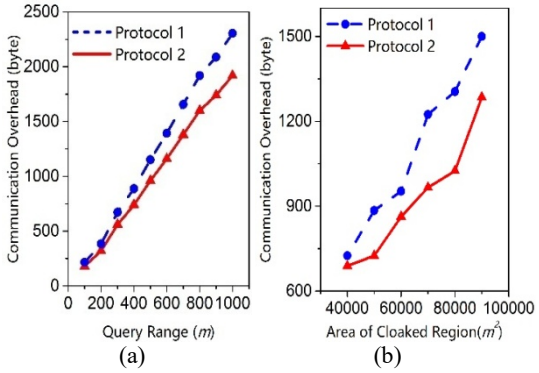


Fig. 7. Communication overhead of baseline and enhanced protocol. (a) Communication overhead, variable query range,. S_{R2}= $6 \times 10^4\, m^2$; (b) Communication overhead, variable cloaking region size, r = 1000m.

(2) The overhead of both protocols increases for larger queried range and cloaking size, echoing a natural tradeoff between performance and privacy.

(3) Since Protocol 2 performs the encryption and decryption for 2 times when filtering the POI information in the result set, the computational cost of Protocol 2 is slightly larger than that of Protocol 1. However, Protocol 2 does not need to transmit redundant POI information during interaction, thus the communication cost of Protocol 2 is less than that of Protocol 2. Moreover, Protocol 2 can provide higher degree of privacy protection.

7. Related Work

The widespread adoption of location-based services (LBS) raises increasing concern about location privacy. A lot of techniques are proposed to prevent personal location data from being exposed and misused.

Existing approaches for user query anonymization in LBS can be roughly grouped into the following five classes: (1) Cloaking[2–5, 11-13], (2) Obfuscation[14, 15] (3) False Locations[4, 23] (4) Space Transformation[6-7], and (5) Dummies[16-18].

7.1. Cloaking

User's query is regarded as the nearest-neighbor (NN) search problem. The cloaking approach was aimed to provide location k-anonymity[19], a variant of classic *k*-anonymity[20]. It requires that any query sent by a user can not be distinguishable from another (at least) *k–1* users. Sadikin[10] provides a pruning rules for searching the result of a NN query. Nearest neighbor relation can also be used for dimension reduction of data[21]. Cloaking has two different flavors, namely spatial cloaking and temporal cloaking. In spatial cloaking, when a user sends her location query, an anonymizaion server replaces the location by a cloaked region enclosing the user and another *k–1* users; In temporal cloaking, the anonymization server deliberately delays a user's query for a specific period so that at least *k –1* users have visited the cloaked region. Later, Casper[11] and CliqueCloak[22] were proposed, and they enabled users to personalize their requirements on location precision and query delay.

A problem with cloaking is that users must trust a centralized anonymizer. To address this issue, some P2P-based cloaking approaches[12-13] are proposed. However, just as the centralized cloaking approaches, P2P-based cloaking still faces the following problem: most LBS Apps expect users to provide exact locations, while few of them (if any) accept inputs of "cloaked regions". This would greatly prevent the adoption of cloaking techniques on mobile devices.

7.2. Obfuscation

Ardagna et.al.[15] proposed obfuscation operators to transform locations into circular areas, so that LBS cannot identify the exact location of a user is. The obfuscation operators include "Enlarge"/"Reduce" the radius of the circle, and "Shift" the center of the circle. In contrast to the above obfuscation which are performed on the Cartesian plane, Duckham et.al.[14] studied the obfuscation of road networks, where locations are modeled as vertices, and proximity of location are modeled as edges. Just as the cloaking approaches, obfuscation also assumes that LBS can process the input of areas or regions, which is not the case for real LBS Apps.

7.3. False Locations

This approach sends false locations (in contrast to real location where the user is) to the LBS, and then tries to construct the right answer based on the results returned by LBS[4, 16]. In SpaceTwist [4], users first sends a false location (termed as "anchor") and a value for distance. LBS returns all POIs within the distance from the anchor. Based on the results, the user then increases the distance and requests the server again. The process continues until the exact result (the nearest POI) is obtained. Similarly, Cover Location[23] also enables a user to retrieve all nearby POIs without revealing her true location. The idea of Cover Location is that the user sends some fake locations near the real location to the LBS, which returns all POIs within the circular areas centered at these location. Then the user's device processes these POIs locally to construct the nearby POIs corresponding to the real location. The above false location approaches can be made transparent to LBS, thereby integrated with existing LBS Apps. However, to construct the true results, users often need to continuously interact with or send multiple queries to location servers, thus resulting in a large latency.

7.4. Space Transformation

In space transformation approaches, users transform the original space into another encoded one, and constructs queries in the encoded space. Then, the location server evaluate k-Nearest Neighbours (KNN) in encoded space, and the user decodes to obtain the locations of KNN in the original space. For example, Khoshgozaran and Shahabi[7] used Hilbert space filling as the one-way transform function. Ghinita et al.[6] proposed another approach based on Private Information Retrieval (PIR). PIR allows a user to retrieve a piece of information without revealing which one she has requested. The authors applied PIR on top of Hilbert space fill for approximate KNN, and Voronoi diagram space representation for exact KNN. As noted above, space transformation approaches are power tools for location query anonymization, but are only limited in functionalities, as it only provides KNN querying services.

7.5. Dummies

Sending a true location query with indistinguishable dummies is a natural way to provide location privacy[17]. Lee et.al.[16] proposed to use dummy locations to hide the real navigational queries. In their approach, a user querying a route from s to t sends two sets S and T satisfying $s \in S$ and $t \in T$ to the LBS. The LBS returns the route for every pair of start $s' \in S$ and destination $t' \in T$. In SybilQuery[18], each time a user sends a location-based query, $k-1$ dummy queries are sent as cover. These dummy queries are constructed in the following steps: (1) the user specifies her start and destination before trip; (2) an endpoint generator consults a database recoding the history of regional traffics, and generates $k-1$ synthetic endpoints; (3) an path generator consults publicly available navigational systems for a sequence of waypoints; (4) when the user needs to query the LBS, the query generator simulates the $k-1$ synthetic paths and generates k−1 dummy locations for query.

This paper only discussed a small subset of these techniques due to limited space. Interested readers can refer to Wernke's survey[24] for a broader coverage.

8. Conclusions

In this paper, we propose Circular Shifting Encryption Protocol (CSEP) based on homomorphic encryption and spatial cloaking for location privacy

preserving, which can protect the privacy of users' location in the query process under the premise of ensuring the accuracy of results. CSEP can guarantee the accuracy of the results, provide strict privacy protection via encryption method, and also reduce the transmission redundancy with a circular shifting encryption method and improve the privacy protection degree at the same time. Experimental results show that CSEP can provide both privacy protection and the accuracy of results under reasonable computational and communication overheads. In the future, we will further improve the performance and scalability of CSEP to support large-scale user queries.

References

1. S.B.Wicker,"The loss of location privacy in the cellularage,"Commun.ACM, vol. 55, no. 8, pp. 60–68, Aug.2012.[Online].Available:http://doi.acm.org/10.1145/2240236.2240255
2. Niu B, Li Q, Zhu X, et al. Achieving k-anonymity in privacy-aware location-based services[C]// IEEE INFOCOM 2014 - IEEE Conference on Computer Communications. IEEE, 2014:754-762.
3. Zhangwei H, Mingjun X. A Distributed Spatial Cloaking Protocol for Location Privacy[C]// Second International Conference on Networks Security Wireless Communications and Trusted Computing. IEEE, 2010:468-471.
4. 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 ICDE, 2008.
5. Bamba B, Liu L, Pesti P, et al. Supporting Anonymous Location Queries in Mobile Environments with PrivacyGrid[C]// International Conference on World Wide Web. ACM, 2008:237-246.
6. G.Ghinita, P.Kalnis, A.Khoshgozaran, C.Shahabi, and K.-L.Tan, "Private queries in location based services: anonymizers are not necessary." In International Comference on Management of Data, 2008
7. A. Khoshgozaran and C. Shahabi, "Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy," in Advances in Spatial and Temporal Databases, 2007.
8. Ghinita G. Private Queries and Trajectory Anonymization: a Dual Perspective on Location Privacy[J]. Transactions on Data Privacy, 2009, 2(1):3-19.
9. P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in Advances in Cryptology EUROCRYPT 99, ser. Lecture Notes in Computer Science, J. Stern, Ed. Springer Berlin Heidelberg, 1999, vol. 1592, pp.
10. Sadikin M F, Kyas M. Efficient Security and Privacy Protection for Emerging Smart RFID Communications[J]. International Journal of Networked & Distributed Computing, 2014, 2(3):156.
11. Mokbel M F, Chow C Y, Aref W G. The new Casper: query processing for location services without compromising privacy[C]// International Conference on Very Large Data Bases, Seoul, Korea, September. DBLP, 2006:763-774.
12. G. Ghinita, P. Kalnis, and S. Skiadopoulos, "Mobihide: a mobilea peerto-peer system for anonymous location-based queries," in Advances in Spatial and Temporal Databases, 2007.
13. Chow C Yet.al, "Spatial cloaking for anonymous location-based services in mobile peer-to-peer environments," GeoInformatica,vol. 15, no. 2, pp. 351–380, 2011.
14. Duckham M, Kulik L. A Formal Model of Obfuscation and Negotiation for Location Privacy[J]. Lecture Notes in Computer Science, 2005, 3468:152-170.
15. C. A. Ardagna, M. Cremonini, E. Damiani, S. D. C. Di Vimercati, and P. Samarati, "Location privacy protection through obfuscation-based techniques," in Data and Applications Security XXI, 2007.
16. K. C. Lee, W.-C. Lee, H. V. Leong, and B. Zheng, "Navigational path privacy protection: navigational path privacy protection," in CIKM, 2009.
17. H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in International Conference on Pervasive Services, 2005.
18. P. Shankar et.al, "Privately querying location based services with sybilquery," in International Conference on Ubiquitous Computing, 2009.
19. M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in MobiSys, 2003.
20. L. Sweeney, "k-anonymity: A model for protecting privacy," International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 10, no. 05, pp. 557–570, 2002.
21. Ishii N, Torii I, Nakashima T, et al. Generation and Mapping of Multi-Reducts Based on Nearest Neighbor Relation[J].International Journal of Networked & Distributed Computing, 2014, 2(1):1-10.
22. B. Gedik and L. Liu, "Protecting location privacy with personalized kanonymity: Architecture and algorithms," IEEE Transactions on Mobile Computing, vol. 7, no. 1, pp. 1–18, 2008.
23. S. T. Peddinti, et.al., "Cover locations: availing location-based services without revealing the location," in WPES, 2011.
24. M. Wernke, et.al., "A classification of location privacy attacks and approaches," Personal and Ubiquitous Computing, vol. 18, no. 1, pp. 163–175, 2014.