Privacy Preserving Scheme for Location and Content Protection in Location-Based Services

Tao Peng[1], Qin Liu[2], Guojun Wang[3(✉)], and Yang Xiang[4]

[1] School of Information Science and Engineering, Central South University, Changsha 410083, China
[2] School of Information Science and Engineering, Hunan University, Changsha 410082, China
[3] School of Computer Science and Educational Software, Guangzhou University, Guangzhou 510006, China
csgjwang@gmail.com
[4] School of Information Technology, Deakin University, 221 Burwood Highway Burwood, Melbourne, VIC 3125, Australia

Abstract. Location-Based Services (LBSs) have been facilitating and enriching people's daily lives. While users enjoy plenty of conveniences, privacy disclosure in terms of both location information and query contents is common. Most of the existing solutions mainly focus on location privacy and adopt K-anonymity principle to preserve user's privacy. However, these methods are vulnerable to protect user's query content. In this paper, we propose a Privacy Preserving and Content Protection (PPCP) scheme for LBSs users. Unlike most of researches requiring a trusted third party (TTP), our scheme is based on a semi-trusted middle entity, which is unaware of both the exact location information about issuer and query content in the user's requirement. We utilize space filling curve to transform user location and protect user query content based on encryption technology, so that the proposed scheme can provide enhanced location privacy and query privacy protection in both snapshot and continuous LBSs.

Keywords: Location-Based Service (LBS) · Hilbert curve · Location privacy · Query privacy · Continuous query

1 Introduction

The proliferation of location-aware devices and rapid development of wireless communication have fostered various Location-Based Service (LBS) applications. According to a new research report [1] from the analyst firm Berg Insight, the global market for mobile LBSs is forecasted to increase from 10.3 billion Euro in 2014 at a compound annual growth rate (CAGR) of 22.5 % to 34.8 billion Euro in 2020. Searching for points of interests (POIs) based on a user's location is one

G. Wang et al. (Eds.): SpaCCS 2016, LNCS 10066, pp. 26–38, 2016.
DOI: 10.1007/978-3-319-49148-6_3

of the most popular applications in LBSs. Users can enjoy the service by issuing *snapshot* or *continuous* LBS queries [2] to a Location Service Provider (LSP) anytime and anywhere. Typical snapshot LBS requirements include k-Nearest Neighbor *(kNN) query* (e.g., "Get top-5 nearest hotels around me"), and *range query* (e.g., "Find all hospitals within the scope of 1 km"). Continuous query can be like "Continuously send me the nearest restaurant on my road every 5 min", or "Continuously report me real time traffic information on my road". For all these queries, users should submit their current locations and requirement contents (e.g., types of POI) to the remote LSPs to activate the LBSs. While users get great benefits from LBS, they may put the sensitive information in jeopardy. The adversary can collect user data in various ways to infer some privacy information of users, such as user's identity, home location, hobbies, and even health condition and religious affiliation, etc. Generally, privacy concerns in LBSs exist in two aspects [3,4]: *location privacy* and *query privacy*. The former is related to the disclosure and misuse of user's location information, the latter, on the other hand, is related to disclosure of the service content. Although distinct, these two types are closely related. There is possibility that compromising one of them may lead to the failure of the other.

Existing researches mainly focus on location privacy and adopt popular K-anonymity principle [5,6] for privacy protection: A user satisfies K-anonymity if the location information sent to the LSP is made indistinguishable from those of at least other K-1 users. To achieve location K-anonymity, a trusted third party (TTP), called the Anonymizer, is introduced acting as an intermediate tier between the users and the LSPs. The Anonymizer blurs exact location of a user into an anonymizing spatial region (ASR or K-ASR) and then transmits the query to the LSP. Even if the adversary knows there are K users in the region, he cannot learn the exact position of each user with a probability larger than $1/K$. However, the trusted Anonymizer has knowledge about all users' locations, which will lead it to be an attractive attack target. Once it is compromised by the adversary, the privacy of users or even the security of whole system will be under threat. Moreover, in practice, it is a tricky thing to find a third party that can be fully trusted by all users.

Another challenging issue to location K-anonymity arises from the *correlation* feature of continuous LBS. When a user sends continuous queries as he moves, a time-series sequence of the corresponding cloaking areas may be tracked and associated to refine the users location, which is called query association attack [7]. For example, assume three users, *a, b,* and *c* are located at different positions. User *a* issues two continuous queries in his trip. The simple K-anonymity (e.g., $K = 2$) approach used to generate an anonymity set (*a, b*) for the first query and an anonymity set (*a, c*) for the second query. The attacker can infer user *a* is the original sender by intersecting these two sets.

In this paper, we propose a Privacy Preserving and Content Protection (PPCP) Scheme for snapshot or continuous LBSs, in which both of location privacy and query privacy are preserved without any fully trusted entities. The key idea is to place a semi-trusted server, called Semi-Anonymizer, between the

user and the LSP. By semi-trusted we mean that the server has no knowledge about a user's real location and query content, while it is honest to respond to all messages and process required operations in the scheme, i.e., it will be able to blur user's exact location and to perform the results matching operations with some transformed and shifted location information of users. The main contributions of proposed scheme are shown as follows:

1. We utilize a space filling curve to perform location transformation on the user and LSP side. The unauthorized entity (includes the Semi-Anonymizer), without the encrypted transforming parameters, is unable to infer any knowledge about a user's real location.
2. We use the public key encryption technique to protect query content so that the query privacy of user is preserved in our scheme.
3. We consider the problem of privacy leakage in the continuous LBSs, and enable the Semi-Anonymizer to cache all of candidate POIs within the whole querying area, so as to reduce the number of queries sent to the LSP. It not only greatly saves the overhead on the Semi-Anonymizer, but also reduces the risk of private information exposure to the LSP or the adversary.
4. Without compromising real locations, the Semi-Anonymizer still has ability to correctly match accurate results for each issuer, hence the user in our scheme can obtain desired answers at low communication, while privacy is preserved.

The remainder of this paper is organized as follows. We introduce technical preliminaries in Sect. 2, and describe the proposed PPCP scheme in Sect. 3. Then, we analyze the performance of our scheme in Sect. 4. Finally, we conclude this paper and present the future work in Sect. 5.

2 Preliminaries

In this section, we first present our system architecture of PPCP scheme, then provide the attacker model and the security requirements. Next, we give an overview for the Hilbert curves and location transformation method, which serve as the technical basis of our work.

2.1 LBS Query

Given a set of static objects $S = (o_1, o_2 ... o_n)$ in 2-dimension (2-D) space, each object has a *type* attribute, $type = TP_{poi}, TP_{poi} \in \{restaurant, hospital, hotel...\}$. A typical LBS user u enjoys the service involving two types of queries:

Definition 1. Snapshot LBS query. The user u with a query location *loc* issues a kNN query trying to find top k POIs from S where $type = TP_{poi}$. The query answer returned by the LPS is set $S' \subset S$ of k objects, where for any object $o \in S'$, and $o' \in S - S'$, $D(o, loc) \leqslant D(o', loc)$, D is the Euclidean distance function.

In case it is a range query (e.g., 1 km), the returned answer is set S', where for any object $o \in S'$, $D(o, loc) \leqslant 1\,\text{km}$. Since k or *range* is a pre-determined parameter, we can represent the query as a 4-tuple $< ID, loc, TP_{poi}, t_i >$, where t_i is the timestamp when the query issues.

Definition 2. Continuous LBS query. A continuous LBS requirement Q includes of a sequence of 4-tuples $q_1 :< ID, loc_1, TP_{poi}, t_1 >, q_2 :< ID, loc_2, TP_{poi}, t_2 > , ...q_n :< ID, loc_n, TP_{poi}, t_n >$, $\forall i \in [1, n-1]$, $t_{i+1} > t_i$.

2.2 System Architecture

Figure 1 illustrates the system architecture of proposed scheme. We employ three roles, the mobile user, the Semi-Anonymizer and the LSP in our system.

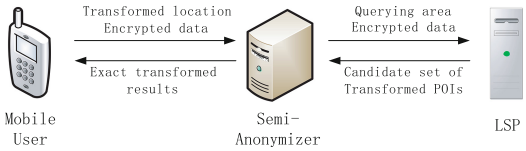


Fig. 1. System architecture of our PPCP scheme

Mobile user: A mobile user carries location-aware (e.g., GPS) devices loaded with LBS applications. The user can determine his current location information by the GPS, and transforms his location and encrypts related information with the pre-process modules of the application. Then, he can enjoy the service by submitting the kNN or range LBS query to a specified LSP by the Semi-Anonymizer, which can be snapshot or continuous queries, for instance "Report me the top-5 nearest hotels", or "Continuously send me the restaurants within 1 mile of my current location every 5 minutes". The user concerns about location privacy and query privacy preserving when he seeks desired information from LSP.

Semi-Anonymizer: It is a semi-trusted party, acting as an intermediate tier between the mobile user and LSP. Semi-trusted in the context means that, on the one hand, it will honestly and correctly carry out all the required operations in the scheme, and will not arbitrarily modify or tamper with any messages, as well as falsifies fake messages; on the other hand, it is curious, and may attempt to locate a query sender and determine his identity based on what it has "see". While, without the transformation parameters and encryption keys, it has no knowledge about the user's real location and service content, as long as it does not collude with LSP. The main jobs of the Semi-Anonymizer in our system are as follows: (i) it processes the queries when receiving users' LBS requests, such as storing user information and forwarding encrypted data to the user-specified LSP. (ii) After getting response from LSP, it conducts result match operations

under the rule of PPCP, pruning false points from the set of candidate POIs, and then returns the exact answers to query issuer.

In practice, the Semi-Anonymizer is analogous to a proxy server maintained by network carriers or other organization. It can be deployed on the network access points or intermediate nodes in different network environments (e.g., base station or gateway) and can be configured based on different policies. For example, in previous research [8,9] the Access Point (AP)-based approach has been used for LBSs in mobile environments. For ease of explanation, in this paper, we only use a single Semi-Anonymizer, but multiple Semi-Anonymizers should be deployed as necessary in reality.

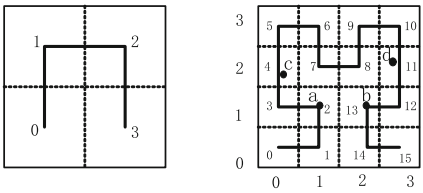
LSP: The online location-based service provider, (e.g., Google Maps or Four Square), employs location-based database servers. They store map resources and the information of POIs (hotel, restaurants and bank, and so on), and other service information as well. As shown in the Fig. 1, LSP does not directly communicate with mobiles users, instead, it provides service via the Semi-Anonymizer. Upon receiving request, LSP searches desired information in its database and returns potential POIs to the Semi-Anonymizer. From the security and privacy aspects, like most researches assumed, LSP is always considered to be an untrusted entity. It has ability to collect all location or content information included in the queries to infer some sensitive data of users, also it may release valuable information to other third parties for monetary reason.

2.3 Threat Model

In our scheme, attackers collect information in various ways, trying to infer the exact location or service content of an LBS issuer. They are assumed to have the following capabilities:

- The location information in the query. This assumption states that either the Semi-Anonymizer is not trusted, or the communication channel between users and the Semi-Anonymiezer is not secure.
- The querying areas of user, and all positions of POIs within this area. This assumption implies that the LSP is untrustworthy. In the worst case the attacker is the LSP itself. The attacker has the knowledge about the map, the POIs in the querying areas, and also has the ability to collect all users' snapshot queries, or keep the history of continuous queries.
- The algorithms or methods that are used to offer privacy in the LBS. This assumption is common in the most security literatures due to the privacy algorithms are usually publicly available.

In our scheme, we also assume that the Semi-Anonymizer will not collude with LSPs. Collusion between the Semi-Anonymizer and some malicious LSP could lead to privacy disclosure. This assumption has also been made in many researches in the field of system security and privacy protection [10,11], in which the server is assumed to not collude with other entity to ensure the security of whole system.



(a) 1-Order (b) 2-Order

Fig. 2. Hilbert curve in two dimensions. (a) *1*-order. (b) *2*-order

2.4 Hilbert Curve

Hilbert curve [12] is one of the space-filling curves, which traverses through all cells in a two-dimension or multidimensional space once and only once. A N order Hilbert curve in 2-D space is a line which goes through every cell in a square where is separated into $2^N * 2^N$ equal-sized cells. Each cell is assigned by an integer value, named Hilbert value (denoted as H-value), according to the sequence when the curve traverses. Figure 2 shows the first two steps of production of the Hilbert curve in two dimensions. Figure 2(a) is a 1-order curve, where the square is divided into $2^1 * 2^1$ cells and the curve orderly passes through their center points to generate H-values of these cells. The Fig. 2(b) shows a 2-order Hilbert curve, which orderly passes through each center point of $2^2 * 2^2$ cells. In the square space, if we use the grid coordinates to denote very cell as $< i, j >$, the corresponding Hilbert value of each cell based on the space-filling curve order can be determined. This process can be defined as encoding and its inverse operation is decoding.

Definition 1: The H-value of a cell s, $< i_s, j_s >$, in the grid coordinate system can be transformed as

$$H(s) = \dot{f}(< i_s, j_s >) \tag{1}$$

where $0 \leqslant i_s, j_s < 2^N$, $0 \leqslant H(s) < 2^{2N}$, and $\dot{f}$ is the spatial transformation function, which transforms the 2-D grid coordinate into a 1-D H-value by a Hilbert curve. Given a curve setting parameter, the curve is determined, and the H-value mapping to each grid cell is assigned. We term this parameter as spacial transformation parameter (STP), and $STP = \{(X_o, Y_o), N, \Gamma, \Theta\}$, where (X_o, Y_o) is the curve's starting point, N is the curve order, Γ is curve orientation, and Θ is curve scale factor. For example, in the Fig. 2(b), the users *a, b, c, d* have the grid coordinates of $< 1, 1 >, < 2, 1 >, < 0, 2 >, < 3, 2 >$. When the 2-order Hilbert curve orderly passes each cell, the H-values of these users are 2, 13, 4 and 11, respectively. The Hilbert curve is suitable for our scheme due to its important property, that is the spatial transformation $\dot{f}$ is one-way function if the STP is not known [13,14]. The procedure of encoding the 2-D space and generating 1-D H-value by such a one-way function can be regarded as encrypting the elements of the original space, and STP is the key of this encryption. Any malicious attacker, without this key, is computationally

impossible to reverse the transformation and decode the 1-D H-value back to the original space.

2.5 Location Transformation

In our scheme, we use location transformation method to preserve user's location privacy. This process is conducted on user's mobile device before user submits the LBS query. Mobile user determines his current location, *loc*, by location-aware device. We assume it is a point and is identified by two values, for instance, its latitude and longitude. Without loss of generality, we define the coordinate (x, y) as to the spatial position of the mobile node in the 2-D space (i.e., x- and y-axes). User has to first specify a STP of space-filling curve to transform the point, since the PPCP scheme use the Hilbert method to perform transformation. Specifically, referring to the curve scale factor, Θ, user can freely choose an area, e.g., a city or a region. According to the irregular spatial region specified by the user, the system will generate a minimum bounding rectangle to contain it as the transformation square space, and we denote it by the coordinates of left bottom vertex (x_l, y_l) and right top vertex (x_r, y_r). Then, the space scale is divided into $m * m$ equal-sized cells to construct a grid system, here $m = 2^N$, N is the order of Hilbert curve specified by the user. We define the unit length of each cell in this square space as *Unit*, and $Unit = (x_r - x_l)/m$. A user u with a 2-D point coordinate of (x_u, y_u) can be presented by the grid coordinate $< i_u, j_u >$.

$$< i_u, j_u > = < \left\lfloor \frac{x_u - x_l}{Unit} \right\rfloor, \left\lfloor \frac{y_u - y_l}{Unit} \right\rfloor > \tag{2}$$

In the cell user located, if we set the left bottom vertex as the origin of coordinates in this cell space, the user's relative location (relative offset to the origin of coordinate) can be presented by $(x_{u\_o}, y_{u\_o})'$.

$$(x_{cl}, y_{cl}) = (Unit * i_u, Unit * j_u) \tag{3}$$

$$(x_{u\_o}, y_{u\_o})' = (x_u - x_{cl}, y_u - y_{cl}) \tag{4}$$

Thus, a user u with a point coordinate of (x_u, y_u) transforms his location as the grid coordinate $< i_u, j_u >$ and his relative offset location in the corresponding grid, $(x_{u\_o}, y_{u\_o})'$. This process is to prepare for the Hilbert encoding.

With the specified STP, using Eq. 1, we can generate a Hilbert curve, and transform the user's grid coordinate $< i_u, j_u >$ to Hilbert value.

$$H(u) = \dot{f}(< i_u, j_u >) \tag{5}$$

Therefore, the user with a 2-D point of $loc : (x_u, y_u)$ can transform his location to 1-D H-value $H(u)$ and a relative position (x_{u_o}', y_{u_o}'). We denote the transformed location as loc'. The Fig. 2(b) illustrates examples of location transformation. In this figure, we assume the *Unit* of each cell is 10. The users *a, b, c, d* have locations of (15,15), (25,15), (6,23), (34,26). After transforming, they

can be presented as {2,(5,5)′}, {13,(5,5)′}, {4,(6,3)′}, {11,(4,6)′}. Notice that, in our scheme we view the cell user located as his querying area. In case the area is too small to preserve the location privacy, user can designate the smallest size of querying cell with a minimum value of *Unit*, where the user can accept to reveal the fact that he is in somewhere within this area without any concerns.

3 Privacy Preserving and Content Protection Scheme

In this section, we will present the details of PPCP scheme, which mainly consists by five steps: Step 1. query issue; Step 2. request processing; Step 3. POI search; Step 4. results match and Step 5. results transformation.

3.1 Query Issue

In order to preserve his privacy, the user has to perform some pre-process before issuing the LBS query. First of all, the user should specify a STP, and transforms his location (loc) to the encoded (loc') using the Hilbert curve as described in Sect. 2.5, which is

$$loc : (x_u, y_u) \rightarrow loc' : \{H(u), (x_{u\_o}, y_{u\_o})'\} \tag{6}$$

We take the user a in the Fig. 2(b) for example, after transforming, his location is $loc : (15, 15) \rightarrow loc' : \{2, (5, 5)'\}$. Then in order to preserve the query privacy, user a encrypts related information in his query, which includes two parts: the type of POI and the STP:

$$C = E^*_{pk}(TP_{poi}, STP) \tag{7}$$

where $E^*_{pk}(\cdot)$ is an asymmetric encryption algorithm under the public key [15] of the LSP. Along these information, the user sends a requirement to the Semi-Anonymizer. Message from user to the Semi-Anonymizer is

$$MSG_{U2A} = q : \{ID, loc', C, t_i\} \tag{8}$$

where ID refers to the identity of user (the user can also use a pseudonym to hide his real identity), loc' is the transformed location of the user, C is the encrypted information in Eq. 7, t_i is the timestamp when the request issued.

While user roams, there is possibility the movement trajectory of user may go out of the scope of one cell, which means the user may have different H-values included in his continuous queries.

Figure 3 shows the continuous query processing in PPCP scheme. In this figure, the red line shows trajectory of user a, the stars ($L_1 - L_4$) are footprints at times $t_1 - t_4$. If we use the Hilbert curve in the Fig. 2(b) to transform user location, the user traverses two Hilbert cells, for instance, 2 and 13. Thus, the transformation of a can be represented as: $loc_1 : (15, 15) \rightarrow loc'_1 : \{2, (5, 5)'\}$; $loc_2 : (18, 15) \rightarrow loc'_2 : \{2, (8, 5)'\}$; $loc_3 : (24, 15) \rightarrow loc'_3 : \{13, (4, 5)'\}$;

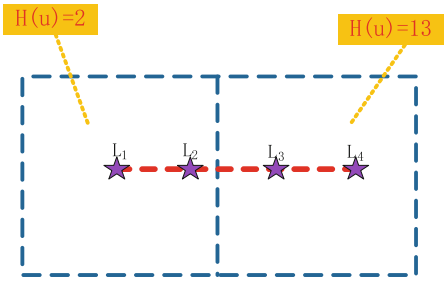


Fig. 3. Example of continuous LBS query processing in PPCP (Color figure online)

$loc_4 : (27, 15) \rightarrow loc'_4 : \{13, (7, 5)'\}$. In the continuous LBS query, messages from the user to the Semi-Anonymizer are sequence requests:

$$\begin{aligned} MSG_{U2A} = \; & q_1 : \{ID, loc'_1, C_1, t_1\}... \\ & q_4 : \{ID, loc'_4, C_4, t_4\} \end{aligned} \tag{9}$$

3.2 Request Processing

Upon receiving the message from user, the Semi-Anonymizer extracts the ID and transformed location, loc'. According to different conditions, it will process the request under following 3 cases:

Case 1: It is an initial (or a snapshot) query, for example, the request, q_1, issued from user a when he is located on L_1, the Semi-Anonymizer stores the ID and location information, and forwards the encrypted query to the specified LSP.

Case 2: It is not an initial LBS query. While the H-value included in this request is just as the same as previous one, which means the user roams in the same cell. For example, the request q_2 and q_4 issued from user a when he is located on L_2 and L_4. Because the Semi-Anonymizer already cached the POIs within the whole cell from the previous answer, it does not need to contact LSP to get the update request results. Instead, the Semi-Anonymizer can skip Step 3, directly executes the results match in Step 4.

Case 3: It is not an initial LBS query. While the H-value in this request is as different as previous one, which means the user moves to other cells. For example, the request q_3 issued from user a when he is located on L_3. The Semi-Anonymizer stores new location information of user, and enlists LSP for help to find a update query answer. In case 1 and case 3, message from the Semi-Anonymizer to the LSP is

$$MSG_{A2L} = \{C, H(u)\} \tag{10}$$

3.3 POI Search

LSP decrypts the message with its private key and retrieves query which includes: the POI-type, the STP and H-value generated by the user in Step 1. With

STP and *H(u)*, LSP can decode the cell user located, and then computes the coordinate values of this cell, which can be presented by left bottom vertex (x_{cl}, y_{cl}) and right top vertex (x_{cr}, y_{cr}). LSP finds all of POIs, which match the required TP*poi* in the cell from its database. For each selected POI P_i with a location (x_i, y_i) from the set of Pe', LSP transforms its real location into the relative offset coordinate, $(x_i, y_i)'$, to the left bottom vertex of the cell by

$$(x_i, y_i)' = (x_i - x_{cl}, y_i - y_{cl}) \tag{11}$$

Finally, LSP returns the set of transformed POIs, Pe' to the Semi-Anonymizer.

$$MSG_{L2A} = \{Pe'\} \tag{12}$$

3.4 Results Match

The Anonymizer obtains candidate POIs, which may potentially be the answers for the range of the entire querying cell. The Semi-Anonymizer caches these all POIs in order to handle the request in the case 2 of Step 2. Then, it finds out the exact results for the query sender. Since the position of user and corresponding candidate POIs are transformed by the same STP, they have the same H-value, which means that their relative offset locations are shifted by the same origin of coordinates (x_{cl}, y_{cl}). Hence, the Semi-Anonymizer can obtain distances between user and each POI, and exactly find out the desired result set, Re', from all candidate POIs according to the requirement of user. Then, the Semi-Anonymizer will return the accurate results to query sender. The message from the Semi-Anonymizer to user is:

$$MSG_{A2U} = \{Re', Re' \subset Pe'\} \tag{13}$$

3.5 Results Transformation

The user obtains exact POIs from the Semi-Anonymizer, but the locations of them are transformed ones. The task of the results transformation is to compute the real locations of the POIs. Notice that when user conducts the location transformation in Step 1, he already has got the origin of coordinates of the cell (x_{cl}, y_{cl}) by Eq. 3, and the POIs got from the Semi-Anonymizer are shifted based on the origin of the cell user located. Therefore, the user can easily transform POIs to the real location by Eq. 11 and finally gets the accurate answers for his LBS query.

4 Performance Analysis

In this part, we analysis the performance of our proposed PPCP scheme regarding computation and communication costs on the user side, the Semi-Anonymizer side and the LSP side respectively.

Computational Cost: First of all, we consider the computational overhead on the mobile user, who conducts *Query issue* (Step 1) and *Results transformation* (Step 5) of the PPCP scheme. The running time of *Query issue* is mainly on the pre-process of query, which includes two parts: location transformation and content encryption. The job of former is to transform the 2-D spatial point into a encoded location by a Hilbert curve, whose computational complexity is $O(N^2)$ [16], where N is the order of Hilbert curve. Generally it is a small constant less than 16. To clarify, the location transformation requires N exponentiations and several multiplications. Here, we only consider the cost of exponential operations due to the its computational overhead is 1000 times that of multiplications. The task of latter is to encrypt the data in Eq. 7 with asymmetric cryptographic algorithm (e.g., RSA). Its computational complexity is *O(1)*, since the encrypted information, type of POIs and STP, have a fixed small size. In terms of the *Results transformation*, its main task is to inverse the transformed POIs to the real places. The expression of transformed POIs are relative offset from the origin of coordinates of the cell, (x_{cl}, y_{cl}), which are already got when the user conducts location transformation in Step 1. Thus user can easily perform *Results transformation* only by several addition operations, and the computational time of it can be neglectful.

Next, we consider the computational overhead on the Semi-Anonymizer, who conducts *Request processing* (Step 2) and *Results match* (Step 4) of the PPCP scheme. The running time of *Request processing* is negligible, since it only needs to simply execute operations of data storage and message forwarding. The *Results match* is to compute distance between each candidate POI to the user, and to select requested results from the all candidate POIs within the whole query area. It depends on the number of POIs, d, and the computational complexity is $O(d)$.

Finally, we consider the computational overhead on the LSP, who conducts *POI search* (Step 3) of the PPCP scheme. The running time of *POI search* is mainly on the following three parts: (1) decode the H-value to the cell user located; (2) search for POIs in the area; and (3) transform the coordinate of candidate POIs. Computational cost of the first part is similar to the procedure of location transformation on the user side, which is $O(N^2)$, and N is a small constant. The running time of the second part is mainly on the number of POIs in the cell, the computational overhead is $O(d)$. The running time of the third part can be ignored, due to this part can be accomplished by several addition operations. Therefore, the computational overhead on the LSP is $O(N^2 + d)$

Communication Cost: We first consider the communication cost between the user and the Semi-Anonymizer. The transfer-out message on the user side is LBS requirement, presented by MSG_{U2A} in Eq. 8, which has s small constant size. The message returned from Semi-Anonymizer is precise answers of his LBS requirement, presented by MSG_{A2U} in Eq. 13. If we consider pre-determined parameters in the requirement specified by the user (e.g., scope in the range query or top-k in the kNN query) as a fix constant, the communication cost between the user and the Semi-Anonymizer is *O(1)*.

Table 1. Performance analysis

Entity	Computational cost	Communication cost
User	$O(N^2)$	$O(1)$
Semi-Anonymizer	$O(d)$	$O(d)$
LSP	$O(N^2 + d)$	$O(d)$

Next, we consider the communication cost between the Anonymizer and LSP. The transfer-in message of LSP is encrypted data, presented by MSG_{A2L} in Eq. 11, whose size is a small constant. The message Semi-Anonymizer got from LSP, presented by MSG_{L2A} in Eq. 12, is the set of candidate POIs in the whole cell. Its size varies with the number of POIs, d, and the communication cost is $O(d)$. We summarize the computational and the communication overhead at the user, the Semi-Anonymizer, and the LSP, respectively, as shown in Table 1.

5 Conclusion

In this paper, we proposed Privacy Preserving and Content Protection (PPCP) scheme to protect user privacy in snapshot and continuous LBS. The main merit of our scheme is that both of location privacy and query privacy are preserved, which is rarely considered in other related approaches. Our scheme does not require any fully-trusted third party (TTP), instead, user privacy is preserved by technologies of location transformation and content encryption. The middle entity, Semi-Anonymizer, is semi-trusted, which has no knowledge of both query content and location information about mobile user. While it still has the ability to match results from the candidate POIs returned by the specified LSP, so that accurate answers can be forwarded to each issuer. Specifically, in continuous LBSs, the Semi-Anonymizer can contact LSP only once and cache all candidate POIs in the whole query area. In the following query processing, it may locally search for answers from the cached POIs and directly replies query user without the participation of LSP. In this way, the communication and computation overhead on the Semi-Anonymizer can greatly reduced. Moreover, the spatio-temporal correlation of continuous queries on the LSP side can be broken, which can enhance the security of whole scheme.

Acknowledgments. This work is supported in part by the National Natural Science Foundation of China under Grant Numbers 61632009, 61472451, 61272151 and 61502163, High Level Talents Program of Higher Education in Guangdong Province under Funding Support Number 2016ZJ01, and Hunan Provincial Natural Science Foundation of China under Grant Number 2016JJ3051.

References

1. Malm, A.: Mobile location-based services. Berg Insights LBS Research Series (2016). http://www.berginsight.com/ReportPDF/ProductSheet/bi-lbs9-ps.pdf
2. Hwang, R.H., Hsueh, Y.L., Chung, H.W.: A novel time-obfuscated algorithm for trajectory privacy protection. IEEE Trans. Serv. Comput. **7**(2), 126–139 (2014)
3. Shin, K.G., Ju, X., Chen, Z., Hu, X.: Privacy protection for users of location-based services. IEEE Wirel. Commun. **19**(1), 30–39 (2012)
4. Pingley, A., Zhang, N., Fu, X., Choi, H.A., Subramaniam, S., Zhao, W.: Protection of query privacy for continuous location based services. In: 2011 IEEE Proceedings of INFOCOM, pp. 1710–1718. IEEE (2011)
5. Gruteser, M., Grunwald, D.: Anonymous usage of location-based services through spatial and temporal cloaking. In: Proceedings of the 1st International Conference on Mobile Systems, Applications and Services, pp. 31–42. ACM (2003)
6. Pan, X., Xu, J., Meng, X.: Protecting location privacy against location-dependent attacks in mobile services. IEEE Trans. Knowl. Data Eng. **24**(8), 1506–1519 (2012)
7. Dewri, R., Ray, I., Whitley, D.: Query m-invariance: preventing query disclosures in continuous location-based services. In: 2010 Eleventh International Conference on Mobile Data Management (MDM), pp. 95–104. IEEE (2010)
8. Luo, W., Hengartner, U.: Veriplace: a privacy-aware location proof architecture. In: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 23–32. ACM (2010)
9. Niu, B., Li, Q., Zhu, X., Cao, G., Li, H.: Achieving k-anonymity in privacy-aware location-based services. In: 2014-IEEE Conference on Computer Communications, IEEE INFOCOM, pp. 754–762. IEEE (2014)
10. Liu, Q., Tan, C.C., Wu, J., Wang, G.: Cooperative private searching in clouds. J. Parallel Distrib. Comput. **72**(8), 1019–1031 (2012)
11. Xiao, M., Wu, J., Huang, L.: Home-based zero-knowledge multi-copy routing in mobile social networks. IEEE Trans. Parallel Distrib. Syst. **26**(5), 1238–1250 (2015)
12. Hilbert, D.: Ueber die stetige abbildung einer line auf ein flächenstück. Math. Ann. **38**(3), 459–460 (1891)
13. Khoshgozaran, A., Shahabi, C.: Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy. In: Papadias, D., Zhang, D., Kollios, G. (eds.) SSTD 2007. LNCS, vol. 4605, pp. 239–257. Springer, Heidelberg (2007). doi:10.1007/978-3-540-73540-3_14
14. Peng, T., Liu, Q., Wang, G.: Enhanced location privacy preserving scheme in location-based services. IEEE Syst. J. **3**(2), 1–12 (2014)
15. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 10–18. Springer, Heidelberg (1985). doi:10.1007/3-540-39568-7_2
16. Liu, X., Schrack, G.: Encoding and decoding the Hilbert order. Softw. Pract. Experience **26**(12), 1335–1346 (1996)