Semantic-Aware Obfuscation for Location Privacy at Database Level

Thu Thi Bao Le and Tran Khanh Dang

Faculty of Computer Science & Engineering, HCMUT
Ho Chi Minh City, Vietnam
{thule,khanh}@cse.hcmut.edu.vn

Abstract. Although many techniques have been proposed to deal with location privacy problem, which is one of popular research issues in location based services, some limitations still remain and hence they cannot be applied to the real world. One of the most typical proposed techniques is obfuscation that preserves location privacy by degrading the quality of user's location information. But the less exact information, the more secure and the less effective services can be supported. Thus the goal of obfuscated techniques is balancing between privacy and quality of services. However, most of obfuscated techniques are separated from database level, leading to other security and performance issues. In this paper, we introduce a new approach to protect location privacy at database level, called Semantic B^{ob}-tree, an index structure that is based on B^{dual}-tree and B^{ob}-tree and contains semantic aware information in its nodes. It can achieve high level of privacy and keep up the quality of services.

Keywords: Privacy, Security, Location based services, Obfuscation.

1 Introduction

Location Based Service (LBS) is a concept that denotes applications supplying utilities to users by using their geographic location (i.e., spatial coordinates) [9]. In recent years, with the development of technologies including mobile devices, modern positioning techniques such as Global Positioning System (GPS), internet (like wireless, 3G), Location Based Services have become more and more popular and have brought many utilities for human life.

However, when using these applications, the users will be asked to send their sensitive information to the service provider, such as location, identity, name, etc. Revealing this information may cause some dangerous. The attackers may know the personal information that the users want to keep secret, such as true position at current time, or the daily schedule. Therefore, privacy in LBS is a serious problem and a lot of research has been done in this field. Note that the more accurate information is, the more easily the privacy is revealed. Therefore the main problem is that how to balance between the quality of services and the privacy.

To solve this privacy-preserving problem, many techniques are suggested, such as k-anonymity based approaches [8, 17, 18], policy based approaches [14], etc. Among

K. Mustofa et al. (Eds.): ICT-EurAsia 2013, LNCS 7804, pp. 111–120, 2013.

them, the most popular one is the obfuscation-based approach. The general idea of this technique is to degrade the quality of user's location information but still allow them to use services with acceptable quality [1, 4]. However, this technique has some limitations. Most of the algorithms that belong to the obfuscation techniques are geometry-based. They do not consider the geographic feature inside the obfuscated region. Based on the knowledge about the geography of the region, the adversary can increase the probability of inferring the exact user's location [3]. Another limitation is that these algorithms are separated from the database level. This makes the algorithms go through two phases: First, retrieving the accurate location of user at the database level, and then obfuscating this information at the algorithm level. So the time cost is increased and security is harder to obtain [6, 7, 12, 13].

Motivated by these issues, we will propose a new semantic-aware obfuscation technique, called Semantic B^{ob}-tree, where semantic of regions are taken into each user's account. This technique is embedded into a geo-database in order to reduce the overall cost.

The rest of the paper is organized as follows. We review related work in section 2. Next, in section 3, we propose a new idea for obfuscation, called Semantic B^{ob}-tree. Following them, the evaluation is discussed in section 4 before we make our conclusion and future work in section 5.

2 Related Work

2.1 Location Obfuscation

There are many approaches to location privacy problem. Among the most popular techniques to protect user's location, the obfuscation gained much interest [1, 2, 10, 19]. Location obfuscation is the process of degrading the quality of information about a person's location, with the aim of protecting that person's location privacy [1, 4]. In [1, 11], the authors propose obfuscation techniques by enlarging, shifting, and reducing the area containing real user's location.

However, these obfuscation techniques just deal with geometry of the obfuscated region, not concern about what are included inside. The adversary with geographical knowledge may infer sensitive location information from obfuscated location generated by geometric based techniques. For example, if an adversary knows that a person is in a region just contains a hospital and a lake. But that person cannot be in the lake (assume no one can be in the lake), so the adversary may easily infer that he/she is in the hospital. We call this privacy attack as spatial knowledge attack.

2.2 Semantic-Aware Obfuscation

Because geometry-based techniques cannot protect location privacy if the adversary has geographical knowledge about obfuscated region, the semantic-aware obfuscation technique has been proposed in [3]. This technique considers sensitive feature types inside the obfuscated region. However, this technique does not concern about how big the area of the obfuscated region is. In some LBS applications, the requirement is that

the area of the obfuscated region must be big enough to protect user's location privacy and must be small enough to ensure the quality of services.

2.3 B^{ob}-Tree

B^{ob}-tree [6, 7, 12] is based on B^{dual}-tree [5] and contains geographic-aware information on its node. The process of calculating the obfuscated region can be done in only one phase: traversing the index structure to retrieve the appropriate obfuscated region that contains user's location. This one-phase process can reduce the processing time. However, in this approach, the region is divided into just two geographic features: approachable and unapproachable parts. The criteria of this division are completely based on geographic features. For example the lakes, the mountains are unapproachable parts, while the parks, the schools are approachable parts. But the user may need more semantic for his/her particular case, such as if the user is a common citizen, the military zone may be unapproachable, but if the user is a soldier, it is an approachable part. In the other words, the adversary may infer sensitive location information by using information about user and geographical knowledge. Moreover, a region can be sensitive for someone but non sensitive for another, or a place is in high sensitivity for someone but is in low sensitivity for another.

Motivated by this, we combine two ideas of semantic aware obfuscation and B^{ob}-tree to make use of their advantages, concerning both the area of the region and the geographic feature inside the regions with more semantic for the users.

3 Semantic B^{ob}-Tree

Much of the research has been done in spatial obfuscation [1, 2, 10, 19], but all of these obfuscation techniques just deal with the geometry of the obfuscated regions. In other words, these obfuscation techniques concern only the area of the regions, but do not care about the geographic feature inside the obfuscated regions. Besides, each semantic obfuscation techniques has its disadvantage, technique in [3] does not concern the area; B^{ob}-tree has not enough semantic information attached for various expectation of different users. So, in this work, we propose a new semantic-aware obfuscation technique which concerns both the area of the regions and the geographic features inside the regions. This new technique not only ensures the same quality of service as others in [1] (because the obfuscated regions produced by these techniques have the same area), but also has higher user's location privacy.

3.1 Concepts

In our proposed technique, the regions are divided into three geographic features: sensitive, non-sensitive and unreachable. A place is sensitive when the user does not want to reveal to be in that area. A place is unreachable when the user because of various reasons, cannot enter in. Otherwise a place is non-sensitive. Like B^{ob}-tree, the requirement is that the obfuscated regions generated by this technique contain only reachable places, and satisfied the privacy of each user's expectation. Next is some concepts proposed in [3] which be used in our new technique.

Sensitive Level. Sensitive level defines the degree of sensitivity of a region for a user. It depends on the extent and nature of the objects located in the region as well as the privacy concerns of the user. It means that if a user is a doctor, a hospital where he/she works in not has a high sensitive level. But for another one, hospital has a high sensitive level because he/she wants to keep the health status to be secret. Sensitive level is in the range [0, 1]. Value 0 means that region is not sensitive or unreachable, we can publish the location of user is that region while value 1 means that region has the highest sensitivity and we have to hide that location.

Sensitive Threshold Value. The sensitive threshold value quantifies the maximum acceptable sensitivity of a location for the user. Its value ranges in [0, 1]. Value 1 means that the user does not care of location privacy, everyone can know his/her true position. Value is closer to 0 means higher degree of location privacy that user wants. A region r is location privacy preserving when its sensitive level is equal or smaller than the threshold value.

Feature Type. Any place belongs to a unique feature type. Users can specify the feature type that they consider sensitive, non sensitive and unreachable. A feature type is sensitive when it denotes a set of sensitive places. For example if hospital is a sensitive feature type, then Hospital A, an instance of Hospital, is a sensitive place. Instead a feature type is unreachable when it denotes a set of places which for various reasons, the user cannot enter in it. For example, the feature type Military Zone may be unreachable if the user is a common citizen. Otherwise, a place is non sensitive. The score of a feature type ft, score(ft), is used to specify how much sensitive ft is for the user. It is in the range [0, 1] and has the same meaning with sensitive level.

Privacy Profile. Users specify which feature types they consider sensitive and score of the sensitivity as well as the threshold value in a privacy profile. Every user has a particular privacy profile (or a group of users has one profile). We use this privacy profile to compute obfuscated regions that satisfy the privacy users expected.

Computation of Sensitivity Level. The sensitivity level (SL) of a region r, written by $SL_{Reg}(r)$ is defined in [3]. It is the sum of ratios of weighted sensitive area to the relevant area in the region. The weighted sensitive area is the surface in r occupied by sensitive feature weighted with respect to the sensitivity score of each feature type. The relevant area ($Area_{Rel}(r)$) of r is the portion of the region not occupied by unreachable feature. In the other words, the sensitivity level of a region r is defined by:

$$SL_{Reg}(r) = \Sigma\ Score(ft_{sens}) * (Area_{Fea}\ (r, ft_{sens}) / Area_{Rel}(r)), \text{ where: } Area_{Rel}(r) = Area(r) - \Sigma\ Area_{Fea}\ (r, ft_{UnReach}) \quad (1)$$

If r only contain unreachable features, we define $SL_{Reg}(r) = 0$.

For example, consider a region r which area is 350 (Area(r) = 350). The score of each feature type and its area in r are as figure 1.

We have: Area $(r, ft_1) = 50$, Area $(r, ft_2) = 100$, Area $(r, ft_3) = 200$, Area $(r, ft_4) = 0$. Apply the formula (1), the sensitive level of region r in figure 1 is:

$Area_{Rel}(r) = 100 + 200 + 0 = 300.$

$SL_{Reg}(r) = 0.9*100/300 + 0.5*200/300 = 0.633$

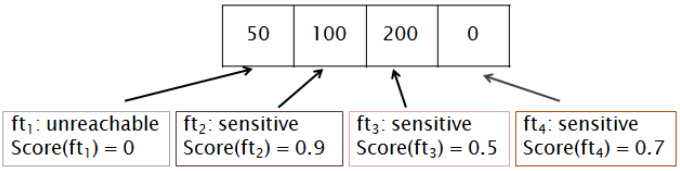

Fig. 1. Example of Computation of Sesitive level

Note that, the sensitive level of a region is less or equal to any sub region in it. For more details and proofs, please refer to [3]. It means that when we merge a region with another region, the sensitive level of the result region is always less or equal to the starting regions.

3.2 Index Structure

The structure of the Semantic B^{ob}-tree is based on B^{+}-tree which indexes one-dimensional values. Similar to B^{dual}-tree [5] and B^{ob}-tree [6, 7, 12], let o be a moving point with a reference timestamp $o.t_{ref}$, coordinates o[1], o[2],…,o[d], and velocities o.v[1], o.v[2],…,o.v[d], its dual is a 2d-dimensional vector as follows expression:

$o^{dual} = (o[1](T_{ref}),\ldots, o[d](T_{ref}), o.v[1],\ldots, o.v[d])$, where $o[i](T_{ref})$ is the i-th coordinate of o at time T_{ref} and is given by: $o[i](T_{ref}) = o[i] + o.v[i] * (T_{ref} - o.t_{ref})$ (2)

First, we apply the Hilbert curve to transform n-dimensional points to one-dimensional values. And then index these values into the structure of Semantic B^{ob}-tree like that of B^{+}-tree. However, each node of the tree has more information to assure semantic-aware privacy preserving. Beside area of reachable regions corresponding to the Hilbert range like B^{ob}-tree, its internal nodes (excluding the leaves) must contains the sensitive level of these regions.

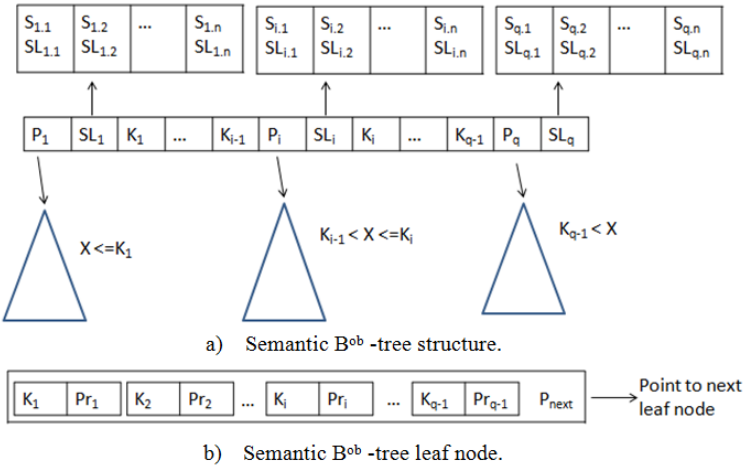

Fig. 2. Semantic B^{ob}-tree index structure

Figure 2 illustrates the structure of the Semantic B^{ob}-tree. Each internal node has the form $<P_1, SL_1, K_1, P_2, SL_2, K_2, \ldots, P_{q-1}, SL_{q-1}, K_{q-1}, P_q, SL_{q-1}>$ where P_i is the tree pointer, SL_i is pointer to an array contains S_i: the area of the reachable region associated with Hilbert interval $[K_{i-1}, K_i]$ and sensitive level (SL_i) of that region for each user (the number of element in this array is equal to number of users) and K_i is the search key value. Note that area of reachable regions S is also the relevant area concept in computation of sensitive level mentioned in equation (1).

Each leaf node has the form $<<K_1, Pr_1>, <K_2, Pr_2>, \ldots, <K_{q-1}, Pr_{q-1}>, P_{next}>$ where Pr_i is data pointer and P_{next} points to the next leaf node of the Semantic B^{ob}-tree.

The sensitive level of a region associated with each internal node (SL_i) will be calculated by applying the equation (1). The area of a region associated with each internal node (S_i) can be calculated by multiplying the total number of cells of each internal node with the area of the projection of each cell into coordinate space [6, 7, 12].

Figure 3 is a simple example of Semantic B^{ob}-tree structure with just one user.

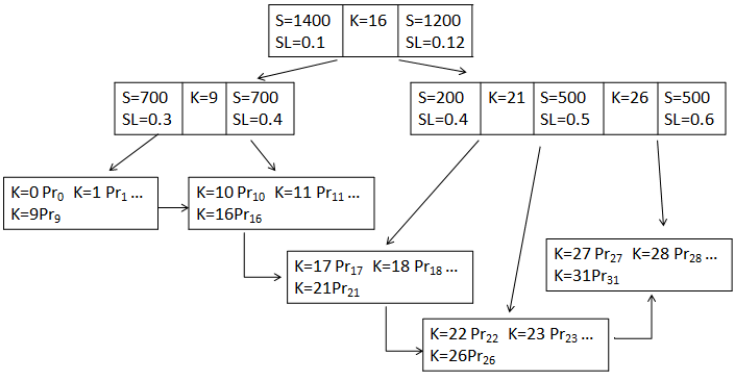

Fig. 3. An example of Semantic B^{ob}-tree with one user

The area of an obfuscated region associated with each internal node in a Semantic B^{ob}-tree is smaller and sensitive level is greater when traversing from the root to the leaf nodes, while the accuracy of user position increases and vice versa. Based on this property, the search process can stop at some internal nodes close to the root if the user wants the very high location privacy.

3.3 Search, Insert, Delete and Update Algorithms

Algorithm: Search
Input: a dual vector of a moving point o^{dual}, area of an obfuscated region S, sensitive thresh hold value θ.
Output: the region that its area equal to S and has sensitive level smaller than θ and contains a moving point with a dual vector o^{dual}.
Transform o^{dual} into Hilbert value h
while (n is not a leaf node) do
Search node n for an entry i such that $K_{i-1} < h <= K_i$

```
   if θ >= SL_i  then
        if S_i  = S then
           return the region corresponding to the Hilbert
interval [K_i-1, K_i].
        else if S > S_i  then
           return ExtendCell (K_i-1, K_i, S).
        else  // if S < S_i
          n ← n.P_i   // the i-th tree pointer in node n
    else
        return no solution.
search leaf node n for an entry (K_i, Pr_i) with h = K_i.
if found then retrieve the user's exact location.
else the search value h is not in database.
```

The algorithm ExtendCell (K_i, K_j, S) extends the region corresponding to the Hilbert interval [K_i, K_j] by adding more adjacent reachable cells until the area of the extended region equals to S. The sensitive level of the new region is equal or smaller than the old ones, thus it is still equal or smaller than the threshold value. This ensures that the obfuscated region generated by this technique has the same area as the geometry-based techniques and achieves better location privacy protection, because its concerns about the semantic of all objects in the region.

In this search algorithm, if the sensitive threshold value is small and the area S is big (e.g. user want the high degree of location privacy), the search process can stop at the internal node near the root node, we don't have to traverse to the leaf node to find the exact user's position as two phases techniques. Only in cases that users are willing to reveal their exact location, the search process must traverse to the leaf node.

The insert, update and delete algorithm of Semantic B^{ob}-tree are similar to those of B^{+}-tree. However, we have to recalculate S and SL of the region associated with each internal node. Due to the space limitation, we do not present the details here.

3.4 Simple Scenario

The map will be divided into cells. Beside the Hilbert values, each cell also contains sensitive information for each user. The users specify the feature types, which feature types they consider sensitive, unreachable, the sensitive score of each feature type and the threshold value in a privacy profile at the first time they register in the LBS system. They can update this profile later. Based on that information and users location, we generate and update Semantic B^{ob}- tree for everyone can request the services.

When the user wants to request services, he/she issues an authorization in form <id_{sp}, id_{user}, Δs> where id_{sp} is the identity of the service provider, id_{user} is the user's identity, and Δs is the area of the reachable regions. The result returns to the service provider, if any, is a reachable region with has the area of Δs, sensitive level of this region is equal or smaller than the threshold value and contains the user's position. If no solution is found because of cannot satisfy both area and sensitive threshold value, user may input another suitable area or change information in the privacy profile.

Besides, it can be interacted with the LBS middleware in [16] and queries processing techniques in [15] to operate completely.

4 Evaluation

4.1 Privacy Analysis

In [1], the authors have introduced the concept of relevance and accuracy degradation for location privacy measurement, as following:

$$\lambda = (A_i \cap A_f)^2 / (A_i \,.\, A_f) \tag{3}$$

In above formula, A_i is the area of location measurement returned by sensing technology based on cellular phones and A_f is obfuscated area that satisfied user's privacy. If the adversary may increase the value of λ from A_f , the privacy is decreased. In our context, assume that the natural degradation due to the intrinsic measurement error (the relevance associated with A_i) is small and A_i is included in A_f, so:

$$\lambda = A_i/A_f \tag{4}$$

Because the obfuscated area A_f of our technique just contains reachable regions and cannot reduce to any small area, the accuracy degradation λ of Semantic B^{ob}-tree is the same as equation (4).

In the geometry-based techniques, because the adversary can remove the unreachable region in A_f, we call the obfuscated area after the removing is A_{fg} ($A_{fg} <= A_f$), so:

$$\lambda_g = A_i/A_{fg} \tag{5}$$

Because $A_{fg} <= A_f$, from (4) and (5) we have $\lambda_g >= \lambda$, means that privacy of the Semantic B^{ob}-tree is higher than geometry-based techniques.

Similarly, in B^{ob}-tree, since the criteria of approachable and unapproachable parts division are completely based on geographic features, the adversary may reduce A_f to A_{fb} ($A_{fb} <= A_f$), by removing the unreachable regions because of other reasons except for geographic reasons (such as personal reasons). We have:

$$\lambda_b = A_i/A_{fb} \tag{6}$$

Since $A_{fb} <= A_f$, from (4) and (6) we have $\lambda_b >= \lambda$, means that the privacy of the Semantic B^{ob}-tree is also higher than B^{ob}-tree.

4.2 Performance

Intuitively, our new technique requires higher storage cost than B^{ob}-tree because it has to store sensitive information for each user in each internal node. Each internal node of Semantic B^{ob}-tree need more (16 bytes * number of users) to store area and sensitive level information (assume we use long and double data type for S and SL).

In the experiment, both B^{ob}-tree and Semantic B^{ob}-tree is all implemented in Java. The user's position, user's privacy profiles are randomly generated. Figure 4 shows the insert, search, update and delete cost (in milisecond) of B^{ob}-tree and Semantic B^{ob}-tree. We can see that the search time of two techniques is approximate. But the cost for inserting, updating and deleting items in Semantic B^{ob} -tree climbs steadily with increasing number of users.

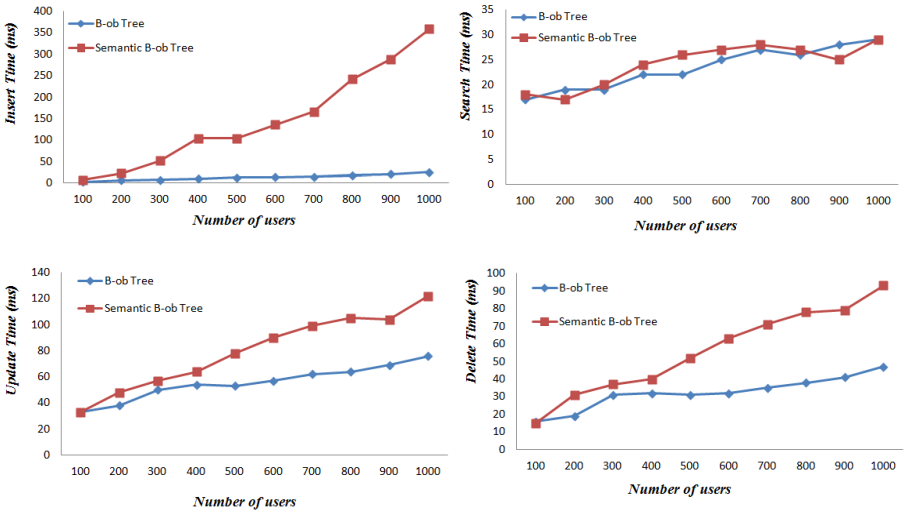

Fig. 4. Insert, search, update and delete time

5 Conclusion and Future Work

In this work, we have introduced the Semantic B^{ob}-tree, a new obfuscated technique for location privacy at database level. Theoretical analyses and discussions have shown that the newly proposed semantic-aware index structure can address the user location privacy more effectively than the B^{ob}-tree [6, 7, 12], which is the first index structure for obfuscation at database level. It is more concretely, more flexible and higher privacy.

In the future, we will intensively evaluate this technique using a variety of datasets to make a comparison with other techniques and optimize the index structure to increase the performance. Quality of services is also a big problem to consider next. Another research direction is to estimate the area and sensitive level of regions into one value for easily computation.

References

1. Ardagna, C.A., Cremonini, M., Vimercati, S.D.C., Samarati, P.: An Obfuscation-based Approach for Protecting Location Privacy. IEEE Transactions on Dependable and Secure Computing (2009)

2. Mohamed, F.M.: Privacy in Location-based Services: State-of-the-art and Research Directions. In: 8th International Conference on Mobile Data Management, Germany (2007)
3. Damiani, M.L., Bertino, E., Silvestri, C.: Protecting Location Privacy through Semantics-aware Obfuscation Techniques. In: IFIP Int. Federation for Information Processing (2008)
4. Duckham, M., Kulik, L.: A Formal Model of Obfuscation and Negotiation for Location Privacy. In: Gellersen, H.-W., Want, R., Schmidt, A. (eds.) PERVASIVE 2005. LNCS, vol. 3468, pp. 152–170. Springer, Heidelberg (2005)
5. Yiu, M.L., Tao, Y., Mamoulis, N.: The B^{dual}-Tree: Indexing Moving Objects by Space-Filling Curves in the Dual Space. VLDB Journal 17(3), 379–400 (2008)
6. To, Q.C., Dang, T.K., Küng, J.: B^{ob}-Tree: An Efficient B^+-Tree Based Index Structure for Geographic-Aware Obfuscation. In: Nguyen, N.T., Kim, C.-G., Janiak, A. (eds.) ACIIDS 2011, Part I. LNCS, vol. 6591, pp. 109–118. Springer, Heidelberg (2011)
7. Dang, T.K., Thoai, N., To, Q.C., Phan, T.N.: A database-centric approach to privacy protection in location-based applications. In: RCICT, Lao PDR, pp. 65–71 (March 2011)
8. Gedik, B., Liu, L.: Protecting Location Privacy with Personalized k- Anonymity. IEEE Transactions on Mobile Computing 7(1), 1–18 (2008)
9. Kupper, A.: Location-based Services-Fundamentals and Operation (2005)
10. Truong, A.T., Truong, Q.C., Dang, T.K.: An Adaptive Grid-Based Approach to Location Privacy Preservation. In: Nguyen, N.T., Katarzyniak, R., Chen, S.-M. (eds.) Adv. in Intelligent Inform. and Database Systems. SCI, vol. 283, pp. 133–144. Springer, Heidelberg (2010)
11. Ardagna, C.A., Cremonini, M., Vimercati, S.D.C., Samarati, P.: Privacy-enhanced Location-based Access Control. In: Handbook of Database Security-Applications and Trends, pp. 531–552. Springer (2008)
12. To, Q.C., Dang, T.K., Küng, J.: A Hilbert-based Framework for Preserving Privacy in Location-based Services. Int. Journal of Intelligent Information and Database Systems (2013) ISSN 1751-5858
13. Phan, T.N., Dang, T.K.: A Novel Trajectory Privacy-Preserving Future Time Index Structure in Moving Object Databases. In: Nguyen, N.-T., Hoang, K., Jędrzejowicz, P. (eds.) ICCCI 2012, Part I. LNCS (LNAI), vol. 7653, pp. 124–134. Springer, Heidelberg (2012)
14. Verykios, V.S., Damiani, M.L., GkoulalasDivanis, A.: Privacy and Security in Spatiotemporal Data and Trajectories. In: Mobility, Data Mining and Privacy: Geographic Knowledge Discovery, ch. 8, pp. 213–240. Springer (2008)
15. Ngo, C.N., Dang, T.K.: On Efficient Processing of Complicated Cloaked Region for Location Privacy Aware Nearest-Neighbor Queries. In: Mustofa, K., Neunold, E., Tjoa, A M., Weippl, E., You, I. (eds.) ICT-EurAsia 2013. LNCS, vol. 7804, pp. 101–110. Springer, Heidelberg (2013)
16. Dang, T.K., Ngo, C.N., Phan, T.N., Ngo, N.N.M.: An Open Design Privacy-enhancing Platform Supporting Location-based Applications. In: ACM ICUIMC 2012. Springer, Kuala Lumpur (2012)
17. Truong, A.T., Dang, T.K., Küng, J.: On Guaranteeing k-Anonymity in Location Databases. In: Hameurlain, A., Liddle, S.W., Schewe, K.-D., Zhou, X. (eds.) DEXA 2011, Part I. LNCS, vol. 6860, pp. 280–287. Springer, Heidelberg (2011)
18. Dang, T.K., Truong, A.T.: Anonymizing but Deteriorating Location Databases. Int. Research Journal of Computer Science and Computer Engineering with Applications (POLIBITS) (46), 73–81 (2012) ISSN 1870-9044
19. Truong, Q.C., Truong, A.T., Dang, T.K.: The Memorizing Algorithm: Protecting User Privacy in Location-Based Services using Historical Services Information. Int. Journal of Mobile Computing and Multimedia Communications (a slected paper of MoMM 2009) 2(4), 65–86 (2010)