ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)

EOST-An Access Method for Obfuscating Spatio-Temporal Data in LBS

[1]Gavali A. B, [2]Limkar Suresh, [3] D. S. Patil
[1,2]GHRCEM, Wagholi, Pune, Maharashtra, India
[3]DYPCOE, Ambi, Pune, Maharashtra, India

Abstract
The pervasive diffusion of mobile communication devices and the technical improvements of location techniques are fostering the development of new applications that use the physical position of users to offer location-based services for business, social, or informational purposes. In such a context, privacy concerns are increasing and call for sophisticated solutions able to guarantee different levels of location privacy to the users. Since the development of location-based services, privacy-preserving has gained special attention and many algorithms aiming at protecting user's privacy have been created such as obfuscation or k-anonymity.The OST-tree capable of obfuscating the spatio-temporal data of users. The OST-tree requires more storage space and update overhead although it achieves the lower querying cost and higher privacy protection comparing to the TPR-tree. Also it is easy for the adversary to infer a user's exact position in the obfuscated area if the probability distribution of user's position is uniformly distributed in case of OST- tree. So this problem can be addressed by proposing EOST-tree (AOST-Tree), a structure that embeds the user's privacy policy in its node and obfuscates the spatiotemporal data for the probability distribution of user's position belongs to a region is not uniformly distributed. Because, in real life, the region where a user belongs to depends on many factors related to geography.

Keywords
LBS, Obfuscation, Privacy-Preserving, Spatio-Temporal Indexing

I. Introduction

With the rapid development of Global Positioning System (GPS), there are more than 6.5 billion mobile users by the year 2011 and the number is expected to increase more. Among the various services for mobile phone, the location-based service (LBS) is the most promising one since it supplies users with many value-added services. In order to benefit from these services, users, however, have to reveal their sensitive information such as their current locations. Such novel services pose many challenges because users are not willing to reveal their sensitive information but still want to benefit from these useful services.

To solve this privacy-preserving problem, a variety of algorithms are suggested to hide personal information of users but still allow them to use services with acceptable quality. The general idea of these algorithms is to obfuscate the user's position [3-4], or to anonymize location information [1-2]. There are two limitations of these algorithms. First, all of them deal with only spatial obfuscation, but not temporal one. Second, these algorithms are separated from the database level. This separation makes the algorithms go through two phases: retrieving the exact location of user on the database level first, and then obfuscating this information on the algorithm level. This two-phase process is time-consuming due to the number of disk access required to retrieve user's exact position. Motivated by this, in this work, we propose a new spatiotemporal index structure based on TPR-tree [5], that can contain privacy-preserving information on it. This index structure deals with not only spatial but also temporal obfuscation, so we term it OST-tree, where OST stands for Obfuscating Spatio-Temporal data. Furthermore, because this index structure embeds privacy information on its node, the process of calculating the obfuscated data can be done in only one phase: traversing the index structure to retrieve the appropriately obfuscated data. This one-phase process can reduce the processing time considerably comparing to the two-phase process mentioned above.

In location-based services environment, each user uses many services from various service providers. For each service provider, user just wants to reveal his location in a specific degree of accuracy depending on the user's trust on service providers. For example, users are willing to reveal their exact location to emergency services, but only reveal their obfuscated locations to advertising services. Therefore, classification of location-based service providers according to user's trust is very necessary in order not to violate privacy policy of users. Towards this goal, EOST-tree is designed to feature service provider classification by letting users specify authorizations and put the privacy information on the tree nodes.

The crucial contributions of this paper are to define the new concept of temporal obfuscation, and to embed user's privacy policy into TPR-tree. This new index is capable of obfuscating spatio-temporal data and provides an improvement over the algorithm separated from the database level for both querying costs and user's privacy protection.

III. Overview of Proposed System - Extended OST-Tree

Extend the probability distribution of user's position so that the probability that a user's position (x, y) belongs to a region is not uniformly distributed. Because, in real life, the region where a user belongs to depends on many factors related to geography, it is easy for the adversary to infer a user's exact position in the obfuscated area if the probability distribution of user's position is uniformly distributed.

II. Literature Survey

Paper	Author/Publication/ year	Problem Described	Solution	Future work
OST-Tree: An Access Method for Obfuscating Spatio-Temporal Data in Location Based Services.	Quoc Cuong TO, Tran Khanh DANG, Josef KÜNG, IEEE, 2011.	Since the development of location-based services, privacy-preserving has gained special attention and many algorithms aiming at protecting user's privacy have been created such as obfuscation or k-anonymity. However, all of these researches separate the algorithms from the database level. Thus, the querying process has two phases, querying the database to retrieve the accurate positions of users and then modifying them to decrease the quality of location information. This two-phase process is time-consuming due to the number of disk accesses required to retrieve the user's exact position.	A structure that embeds the user's privacy policy in its node and obfuscates the spatiotemporal data. Experiments show that OST-tree provides an improvement over the algorithm separated from the database level for both querying costs and user's privacy protection. The OST-tree capable of obfuscating the spatio-temporal data of users and it achieves the lower querying cost and higher privacy protection comparing to the TPR-tree.	Future work will extend the probability distribution of user's position so that the probability that a user's position (x, y) belongs to a region is not uniformly distributed. Because, in real life, the region where a user belongs to depends on many factors related to geography, it is easy for the adversary to infer a user's exact position in the obfuscated area if the probability distribution of user's position is uniformly distributed.
An Obfuscation-Based Approach for Protecting Location Privacy	Claudio A. Ardagna, Marco Cremonini, Sabrina De Capitani di Vimercati, and Pierangela Samarati . IEEE Jan-Feb 2011	The physical location of users is rapidly becoming easily available as a class of personal information that can be processed for providing new online and mobile services, generally called Location-Based Services (LBSs) .The pervasive diffusion of mobile communication devices and the technical improvements of location techniques are fostering the development of new applications that use the physical position of users to offer location-based services for business, social, or informational purposes. In such a context, privacy concerns are increasing and call for sophisticated solutions able to guarantee different levels of location privacy to the users.	A solution based on different obfuscation operators that, when used individually or in combination, protect the privacy of the location information of users.	The analysis of solution assuming Gaussian-like distributions and complex location measurement shapes; the introduction of map constraints in the computation of obfuscated areas; the definition of additional techniques for degrading the temporal accuracy of location measurements; the extension of solution to protect the path privacy of the users; and the actual integration and extensive test of given solution in a real scenario.
An Extensible and Pragmatic Hybrid Indexing Scheme for MAC-based LBS Privacy-Preserving in Commercial DBMSs	Tran Khanh Dang, Quoc Cuong To 2009.	The problem of privacy-preserving in LBS.	This paper addressed the problem of privacy-preserving in LBS by proposing a new and extensible framework integrating the most two typical classes of algorithms, obfuscation and k-anonymity. Two commonly used index structures, B+-tree and R-tree, are utilized to create a new and extensible index structure to manage user profile and accelerate the querying process. In addition, applying the MAC to manage the profile and context information of users and classify the service provider accordingly. Finally, the architecture is integrated into commercial DBMSs.	The current framework could be further extended with respect to the following aspects: -new index structure is created to index temporal data for both spatial and non-spatial; -Extend architecture to support more algorithms; -Evaluate the framework in real world applications.

Location Privacy Protection Through Obfuscation-based Techniques	C.A. Ardagna, M. Cremonini, E. Damiani, S. De Capitani di Vimercati, and P. Samarati 2007	The widespread adoption of mobile communication devices combined with technical improvements of location technologies are fostering the development of a new wave of applications that manage physical positions of individuals to offer location-based services for business, social or informational purposes. As an effect of such innovative services, however, privacy concerns are increasing, calling for more sophisticated solutions for providing users with different and manageable levels of privacy.	Privacy-enhanced techniques that protect user privacy based on spatial obfuscation, a solution that both considers the accuracy of location measurements, which is an important feature of location information, and the need of privacy of users. In addition to several obfuscation techniques for privacy preservation, also present and define a formal and intuitive way to express users privacy preferences, and a formal metric for location accuracy.	Issues to be investigated include the analysis of our solution assuming Gaussian-like distributions, the evaluation of obfuscation techniques robustness against de-obfuscation attacks, and the possibility to manage different privacy preferences expressed by users.
B^{ob}-Tree: An Efficient B^{+}-Tree Based Index Structure for Geographic-Aware Obfuscation	Quoc Cuong To1, Tran Khanh Dang1, and Josef Küng2 Springer-2011	The privacy protection of personal location information increasingly gains special attention in the field of location-based services, and obfuscation is the most popular technique aiming at protecting this sensitive information. However, all of the conventional obfuscation techniques are geometry-based and separated from the database level. Thus, the query processing has two time consuming phases due to the number of disk accesses required to retrieve the user's exact location, and the location obfuscation. Also, since these techniques are geometry-based, they cannot assure location privacy when the adversary has knowledge about the geography of the obfuscated region.	These problems are addressed by proposing B^{ob}-tree, an index structure that is based on B^{dual}-tree and contains geographic-aware information on its nodes. Experiments show that B^{ob}-tree provides a significant improvement over the algorithm separated from the database level for query processing time and location privacy protection. The B^{ob}-tree, that is capable of obfuscating the geographic-aware regions. This novel index structure is much more efficient than both OST-tree and TPR-tree in terms of storage space, query processing time, update and privacy protection.	In the future, there will be consideration of the use of the Bob-tree to support other privacy preserving techniques in LBS (e.g., k-anonymity), and will address the quality of LBS problem due to the different shapes of the returned regions wrt. the B^{ob}-tree and other access methods. Another research problem of interest is to extend authorization from <idsp, iduser, Δs> to <idsp, iduser, Δs, Δt, Δp>, where Δt and Δp represent the accuracy degree of time and profile, in order to support temporal and profile obfuscation.
A DATABASE-CENTRIC APPROACH TO PRIVACY PROTECTION IN LOCATION- BASED APPLICATIONS	Tran Khanh DANG, Nam THOAI, Quoc Cuong TO, Trong Nhan PHAN 2011	Privacy preserving in location based services (LBS) has been emerging as a measure for the quality of both LBS providers' services and mobile users' need. A lot of research already done on it can be used to assure user privacy while the quality of services (QoS) must be kept up. However, all of the conventional obfuscation techniques are geometry-based and separated from the database level. Unlike other techniques, the approach presented in this paper will dig privacy protection potential from the database level and exploit its power in an effort of balancing the quality on privacy protection and services.	Addressing the above problems, there is a proposal of OST-tree, an index structure that gets user privacy policy embedded into its nodes and obfuscates the spatiotemporal data; and B^{ob}-tree, another index structure that is based on B^{dual}-tree and contains geographic-aware information in its nodes. Both of them provide a significant improvement over the algorithms separated from the database level in terms of query processing time and user privacy preserving.	In the future, we will address the quality of LBS problem due to the different shapes of the returned regions with respect to the B^{ob}-tree and other access methods.

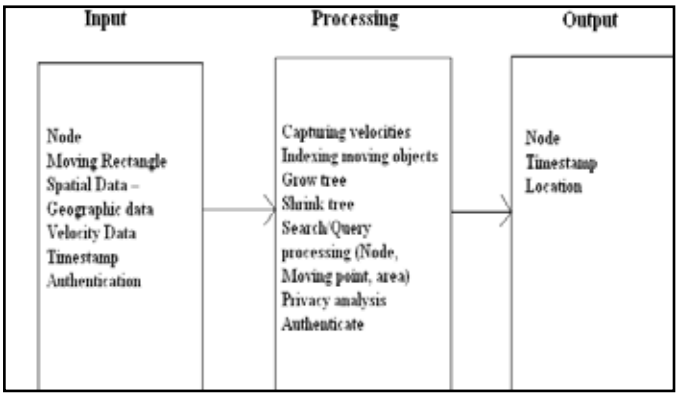

Fig. 1: Block Diagram

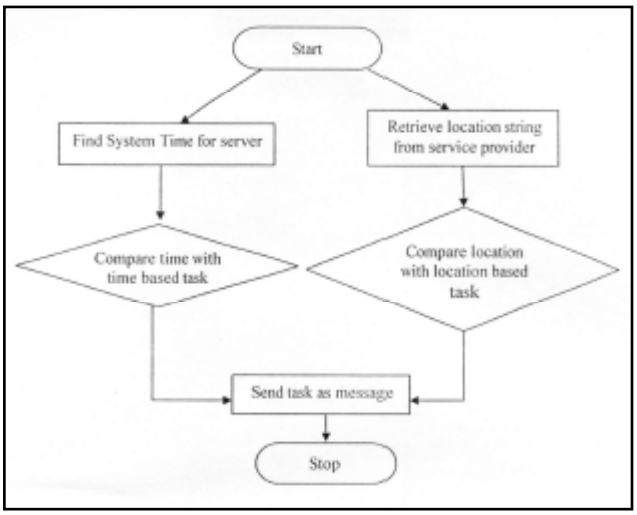

Fig. 2: Flowchart of System

A. Scope

With the location based reminder system, a user gets reminder of what to do , when to do, depending on location and time. It is a client server project in which the boss sitting to the office would set reminders for different locations. It consists of following steps:

1. Server side software has the list of important places of particular city. Boss can frame a set of messages and select the messages with respect to the locations.
2. Stores the database in the form 0f table which contains message to deliver, place , phone no, and date for location based reminder system
3. Store the database in the form of table which contains message to deliver, time, phone no, and date for time based reminder system
4. To the server PC a mobile phone with modem IS connected through serial port.
5. The client application is loaded to the series 60-cell phone with Android O.S.
6. While moving every time the location changes the cell transfers the location details to the server through GSM connection via SMS.
7. Server checks the database if it matches through the mobile connected via serial port the reminder is sent in the form of message.

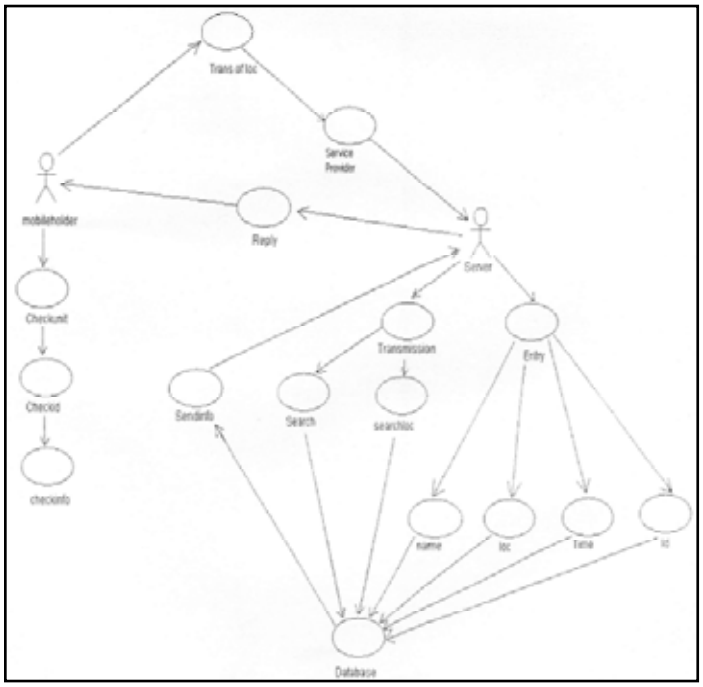

Fig. 3: Use Case Diagram of System

IV. Temporal Obfuscation

Many of the research activities have been done in the area of spatial obfuscation [3- 4], but, no mature proposals for obfuscating the temporal data of users exist. So, the focus on this issue in this section. Similar to spatial obfuscation, temporal obfuscation will degrade the exact value of time t_0 to the vague temporal value [t[,t]], where t[$< t_0 <$ t]. For example, instead of saying that "the position of user will be in location x_0, y_0) in the next 15 minutes", we can obfuscate the time value by saying that "the position of user will be in location (x_0, y_0) in the next 13 to 16 minutes". By combining the spatial and temporal dimension, a spatio-temporal value can be calculated by obfuscating both the spatial and temporal value. For example, according to the above example, we can say: "The user's position is somewhere in the area of 1.2 square kilometer, including the location n (x_0, y_0), and within the next 13 to 16 minutes in the future".

A. Temporal Obfuscation

The obfuscated value of timestamp t_0 is the temporal interval [t[,t]] which includes the real timestamp t_0 with the probability:

$P(t_0 \in [t[,t]])=$ (1)

B. Spatio-Temporal Obfuscation

The obfuscated value of user's exact position (x_u, y_u) at a timestamp t_0 is a rectangular area (x_c, y_c, w, h) centered on the geographical coordinates (x_c, y_c) with width w, height h, at a temporal interval [t[,t]], which includes the user's exact position (x_u, y_u) at a real timestamp t_0 with the probability: $P((x_u, y_u) \in \text{Rectangle}(x_c, y_c, w, h)$ AND $t_0 \in [t[,t]])=1$ (2)

1. Authorization

An authorization α is a 4- tuple <idsp, iduser, Δs, Δt> where idsp is the identity of service provider, iduser is the identity of user, Δs, Δt is the degree of accuracy of user's position (spatial data) and time, respectively. The meaning of an authorization is that a user with the identity iduser allows only the service provider with the identity idsp to access his/her sensitive information of position and time with the degree of accuracy of Δs, Δt, respectively. For

example, a user with the identity #U232 is willing to reveal his position in the next 10 minutes with the accuracy of position and time being 600 square meters, 3 minutes, respectively, to the advertising service with the identity #S101. This authorization can be expressed as α1 = <#S101, #U232, 600m2, 3m>. If the user's exact position in the next 10 minutes is located at a coordinate <x0, y0>, the result returned from the next 9 to 12 minutes to the service provider is a rectangle which has the area of 600 square meters and contains the coordinate <x0, y0> in case of time and position, respectively.

V. Index Structure

The base structure of the OST-tree is that of the TPR-tree for indexing the spatio-temporal data. However, in order to specify the authorization and the degree of accuracy of user's position and time, the node structure will be modified to attach more information. Specifically, in addition to the tpbr, each node contains a pointer p pointing to the list of entries. Each entry has the form of a 4-tuple <id_{sp}, id_{user}, Δs, Δt>, indicating that a service provider with

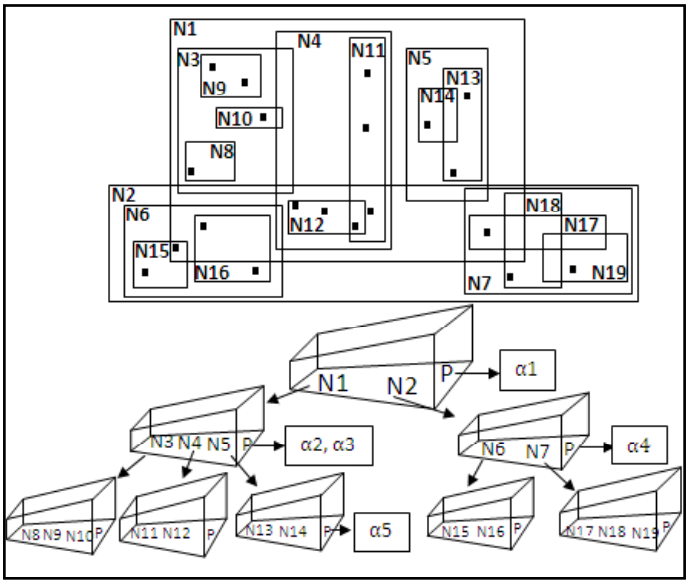

Fig. 4: EOST-Tree Structure

the identity id_{sp} can access sensitive information of a user with the identity id_{user} at the degree of accuracy of user's position and time specified by the value Δs and Δt, respectively. Fig. 4, illustrates the structure of the OST tree. For the illustration purpose, the values of authorizations αi (i=1..5) in this fig. are α1 = <#S101, #U232, 1000m2, 3m>, α2 = <#S101, #U134, 600m2, 3m>, α3 = <#S102, #U232,500m2, 3m>, α4 = <#S101, #U135, 550m2, 4m>, and α5 = <#S103, #U232, 0m2, 0m>. Our goal is to develop an index structure that can incorporate the accuracy degree of user's position. Therefore, this accuracy degree parameter must be in the hierarchical form. The OST-tree achieves this hierarchy well. Since the tpbr in a TPR-tree is already organized in hierarchical structure, the OST-tree inherits this property to hierarchically organize the bounding rectangle containing the user's exact position that will be returned to the service providers. More specifically, when traversing from the root node to a leaf node in the OST tree, the degree of accuracy of user's position increases because the area of the bounding rectangle is smaller and vice versa. For example, in the traversal path N1-N5-N14 (see fig. 4), the areas of the returned rectangles reduce from 1000m2 to 500m2 and 0 m2 corresponding to α1, α3, and α5. This means that the degree of accuracy of user's position increases. Based on this property, if service providers have a higher level of trust from a user, their identities will be placed on the node nearer to the leaf node and vice versa. For instance, the service provider with the identity #S103 has the highest level of trust from a user with the identity #U232, and so it can obtain the user's exact position (Δs=0). This service provider's identity is, therefore, placed on the leaf node.

A. Privacy Information Overlaying and Insertion

The privacy information overlaying and insertion process happen in parallel. We traverse the OST-tree from the root node down to the leaf node to place the new object in the suitable leaf node (by applying the insertion algorithm as shown in [5, 8]) and, at the same time, recursively compare the degree of accuracy of user's position (Δs) with a spatial extent of each node (N■) in the insertion path to find the appropriate node overlaying privacy information. We have two possible scenarios for this comparison:

- Case 1: If (N is the appropriate sub-tree) and (Δs≥ N■), we overlay α on N and continue the insertion process.
- Case 2: If (Δs< N■), depend on the level of N, we have two scenarios: If N is a non-leaf node, we choose an appropriate sub-tree rooted at N (complying with the algorithm ChooseSubtree of R*-trees [8]) and continue the overlaying process. If N is a leaf node, we overlay and insert the new object into this node. If a moving object has already existed in the index structure and the user wants to add new policies, we find the appropriate node in the insertion path to overlay privacy information. Since the authorization is put as high as possible in the OST-tree, the search process can stop at some internal node if the match occurs. Thus, we do not always have to traverse to the leaves to find a user's exact position as in algorithms separated from the database level. For example, if the service provider #S101 wants to obtain the position of user #U135, the search process stops at the internal node N6-N7 and returns the result. But, in the case of an algorithm separated from the database level, we have to traverse to the leaf node N15, where the position of #U135 belongs to, to retrieve the user's exact position and then obfuscate it.

B. Privacy Analysis

1. Adversary Model

The adversary tries to manipulate the obfuscated region to infer the user's exact location. For obfuscation techniques, the relevance is used to measure the location privacy protection. The lower the relevance, the higher the location privacy protection is, and thus the lower the probability an adversary can infer the user's exact location. So, in order to analyze the location privacy protection of proposed approach with that of the approach that separates the algorithm from the database level, we will compare the relevance values of the two approaches. For the approach that separates the algorithm from database level, the relevance is:

$$R_s = \frac{(Ai \cap Af)^2}{Ai * Af} \quad (3)$$

where, A_i is the location measurement [5] and depends completely on the positioning technology, and A_f is the obfuscated region created by the privacy-preserving algorithm.

To calculate the relevance of proposed approach, we can simply replace A_f by Δs in (3). However, the concept of relevance in [5] only concerns about spatial privacy protection. By taking into account the temporal element, we extend the relevance concept to use for both spatial and temporal privacy protection as follows:

$$R_s = \frac{(Ai \cap Af)^2}{Ai * Af} \frac{1}{\Delta t} \quad (4)$$

where Δs and Δt are the degree of accuracy of user's position and time, respectively. From (3) and (4), we can see that $Rs \geq Rst$ (since $A_f = \Delta s$ and $\Delta t \geq 1$), meaning that the degree of privacy protection of proposed approach is higher than that of approach separating the algorithm from database level. More specific, incorporating the temporal dimension into the relevance concept reduces the probability that an adversary can infer the user's exact location because the adversary has to guess not only where, but also when the user's exact position belongs to.

Vi. Comparative Study

Since the node of OST-tree contains authorizations, the number of records in each OST-tree's node is smaller than that of the TPR-tree. So, the OST-tree requires more nodes to contain the same number of moving objects pointer. Also, when traversing to leaf nodes is required in two indexes, the TPR-tree has the lower height and requires less disk accesses than that of the OST-tree since the OST-tree has to reserve the space to store the authorization in each node. However, in most cases, we do not have to traverse to the OST-tree leaves to retrieve the result. Because if the pair value <idsp, iduser> of the query's authorization is matched with that of some internal node, we will stop at this node and return the result without further traversing on the OST-tree. Hence, the OST-tree requires less disk accesses than that of the TPR-tree. Only in the worst case where users are willing to reveal their exact position to service providers, we have to traverse to the leaf node to retrieve the exact result. By incorporating the time into the privacy model, the average relevance of proposed approach is smaller than that of the obfuscation algorithm. The insert cost of OST-tree is higher than that of TPR-tree since we have to spend extra time (or number of I/O operations), besides the time for insertion process, to find appropriate node to overlay authorization. Given the mobility of users, the update cost of OST-tree is higher than that of TPR-tree, because OST-tree has to incur the additional cost of updating the authorization (moving α from current node to another node corresponding to the newly updated position of a user).The query in this case is point location queries, which retrieves the rectangle containing the location of user. In general, the query cost of OST-tree is better than that of TPR-tree since we do not have to traverse to the leaf node to get the result in OST-tree. Only in some cases, where users want to reveal their exact locations to service providers, the OST-tree is not better than TPR-tree, because we have to traverse to the leaf nodes of OST-tree. Hence, OST-tree is better than TPR-tree in cases users just want to reveal a low degree of accuracy of their locations to service providers. So there is use of EOST.

VII. Conclusion and Future Work

In this work, we have introduced the EOST-tree capable of obfuscating the spatio-temporal data of users. The EOST-tree requires less storage space and update overhead, it achieves the lower querying cost and higher privacy protection comparing to the OST-tree. Also the probability distribution of user's position so that the probability that a user's position (x,y) belongs to a region is not uniformly distributed. Because, in real life, the region where a user belongs to depends on many factors related to geography, it is easy for the adversary to infer a user's exact position in the obfuscated area if the probability distribution of user's position is uniformly distributed.

References

[1] C.A. Ardagna, M. Cremonini, E. Damiani, S.D.C. Vimercati, P.Samarati,"Location-Privacy Protection through Obfuscation-based Techniques", DBSEC, 2007.

[2] F.M. Mohamed,"Privacy in Location-based Services: State-of-the-art and Research Directions", MDM, 2007.

[3] T.K. Dang, Q.C. To,"An Extensible and Pragmatic Hybrid Indexing Scheme for MAC-based LBS Privacy-Preserving in Commercial DBMSs", ACOMP, pp. 58–67, 2010.

[4] J. H. Jafarian, M. Amini, R. Jalili,"Protecting Location Privacy through a Graph-based Location Representation and a Robust Obfuscation Technique", ICISC, 2008.

[5] C.A. Ardagna, M. Cremonini, S.D.C. Vimercati, P. Samarati,"An Obfuscation-Based Approach for Protecting Location Privacy", TDSC, 8(1):13-27, 11.

[6] T.T. Canh, T. Q. Chi, T.K. Dang,"An Adaptive Grid-Based Approach to Location Privacy Preservation", ACIIDS, pp. 133–144, 2010.

[7] S. Saltenis, C.S. Jensen, S.T. Leutenegger, M.A. Lopez, "Indexing the Positions of Continuously Moving Objects", ACM SIGMOD, pp. 331–342, 2000.

[8] C.A. Ardagna, M. Cremonini, S.D.C. Vimercati, P. Samarati,"An Obfuscation-Based Approach for Protecting Location Privacy", TDSC, 8(1):13-27, 2011.

[9] A. Guttman,"R-trees: A Dynamic Index Structure for Spatial Searching", SIGMOD, pp. 47–57, 1984.

[10] M.F. Mokbel, T.M. Ghanem, W.G. Aref, "Spatio-temporal access methods". IEEE Data Engineering Bulletin, 26(2): 40–49, 2003.

[11] Sohn T.,Li. K. Lee,G.Smith, LScott J. Griswold, W., "Place - Its: Location - Based Reminders on Mobile Phones", Proc. Intl. Conf. on Ub, 2005.