A Unified Framework for Authenticating Privacy Preserving Location Based Services

Tanzima Hashem[1], Shudip Datta[1], Tanzir Ul Islam[1], Mohammed Eunus Ali[1],
Lars Kulik[2], and Egemen Tanin[2]
[1]Dept of CSE, BUET
Dhaka 1000, Bangladesh
[2]Dept of CIS, University of Melbourne
VIC 3010, Australia
tanzimahashem@cse.buet.ac.bd

ABSTRACT

An important class of location-based services (LBSs) is information queries that provide users with location information of nearby point of interests such as a restaurant, a hospital or a gas station. To access an LBS, a user has to reveal her location to the location-based service provider (LSP). From the revealed location, the LSP can infer private information about the user's health, habit and preferences. Thus, along with the benefits, LBSs also bring privacy concern to the users. Hence, protecting the privacy of LBSs users is a major challenge. Another major challenge is to ensure the reliability and correctness of the provided LBSs by the LSP, which is known as authentication. We develop a novel authentication technique that supports variants of privacy preserving LBSs with less storage and communication overhead. More importantly, we present a unified framework that can handle authentication for a wide range of privacy preserving location-based queries, range, nearest neighbor, and group nearest neighbor queries. We conduct experiments to show the efficiency and effectiveness of our approach in comparison with the state-of-art techniques.

1. INTRODUCTION

With the proliferation of location-aware mobile devices and map based applications (e.g., Google map), the popularity of location-based services (LBS) is daily increasing. Users can access a wide range of LBSs from different location based service providers (LSPs). These LBSs include various types of location based information queries such as finding the nearest point of interest (POI) such as a restaurant or an ATM, searching for a shopping place in 1 km radius around user's location, and finding a meeting place that minimizes the aggregate travel distance for a group of friends. In all these services, users need to reveal their locations to the LSP, from which the LSP could infer the user's health, habit and preferences from the revealed locations. Thus protecting user privacy while accessing LBSs is a major challenge in location-based query processing. Nowadays it is common that the data owner and the LSP are two different entities, and thus, another major challenge is to ensure the authenticity (reliability and correctness) of the location-based services offered by the LSP. In this paper, we develop an authentication technique to process privacy preserving LBSs with less storage and communication overhead, in particular for users' mobile devices. More importantly, we propose *a unified framework* to process *a wide range of privacy preserving location-based queries*, range, nearest neighbor (NN), and group nearest neighbor (GNN) queries, with authentication.

GeoRich '15 May 31 - June 04 2015, Melbourne, VIC, Australia

DOI: http://dx.doi.org/10.1145/2786006.2786009

Due to limited processing capabilities or the lack of technical expertise, a data owner might outsource their data to an LSP. The LSP processes the location-based queries and then sends the answers to the users. Though historically, LSPs were limited to large organizations such as Google and Microsoft, with the widespread use of Android and iOS any small company can act as an LSP. These LSPs use data from a third party, a data owner. Since most of these LSPs are either unknown or little known, users might not want to share their location data with these LSPs due to privacy concerns. A recent survey of Microsoft [12] reveals that 92% of users consider LBSs as valuable services, whereas at the same time 52% of users are worried about their privacy predominantly because they fear that their location data might be shared with unknown organizations and people. In addition, users do not trust these LSPs in terms of the quality of the answers, because it may happen that the LSP works with a third party for personal gain and provides low-quality answers. Therefore, both privacy of users' locations and authenticity of the answer set are major concerns of users of LBSs.

A large body of research works appeared in the last few years to protect user privacy [16, 7]. On the other hand, Voronoi diagrams [9] and Merkle R-tree (MR-tree) [15] are used to enable users to authenticate the reliability and correctness of the obtained results. For authentication, the LSP returns the answer set to the user along with additional information such as a verification object, Voronoi neighbors, and digital signatures obtained from the data owner. The main limitation of existing works is that they can either protect privacy or ensure the authenticity of the result set for LBSs. If we adopt existing authentication techniques to process privacy preserving LBSs, it incurs high communication overhead while authenticating multiple POIs.

In this paper, we develop an authentication technique for privacy preserving LBSs with less storage and communication overhead. More importantly, we present a *unified* framework to process *a wide range* of privacy preserving location queries, range, nearest neighbor (NN), and group nearest neighbor (GNN) queries. An important benefit of our unified framework is that it uses the same index and processing algorithm to handle variants of privacy preserving LBSs. In addition, in our unified framework, a user does

not reveal the query type (range or NN or GNN) to the LSP. A range query returns locations of all POIs within a specific range. An NN query returns the nearest POI for a query location. A GNN query returns the location of a POI that minimizes the aggregate distance from a group of users who are in different locations. The underlying idea of our authentication technique for private LBSs is to incrementally retrieve nearest POIs with respect to a user's false location until the optimal authenticated answer with respect to the actual location has been found.

More specifically, in our approach a user sends her false location instead of her actual location to the LSP for privacy reasons. The LSP returns the nearest neighbor along with the additional information required to authenticate the answer. The user authenticates the returned NN and checks whether the optimal answer is found with respect to the actual location for the specified query (i.e., range, NN, and GNN). If the answer for the required query has not been found, then the user requests for the next NN with respect to the false location to the LSP. The process repeats until the optimal authenticated result with respect to the actual location has been determined by the user.

In this paper, we propose a novel authentication technique for incremental retrieval of nearest POIs, which forms the base of the unified framework to process variants of LBSs. Simply applying existing authentication techniques for NN queries independently for every incrementally retrieved POI would cause retrieval of the same authentication information multiple times and thereby incur high communication overhead. In contrast, our approach ensures that the LSP never returns same POI more than once. In our approach, for every POI, the data owner computes the distance of the farthest Voronoi neighbor (i.e., POI) of the considered POI and determines *count*, i.e., the number of other POIs that have a smaller distance than the farthest Voronoi neighbor. Then the data owner signs the data containing the POI location along with the distance of the POI's farthest Voronoi neighbor and the *count*. Finally, the data owner shares all signed POI information with the LSP. An additional benefit of our approach in contrast to the existing authentication technique [9] is that our approach does not store the locations of Voronoi neighbors of every POI in the LSP's database and significantly reduces the storage overhead.

In response to a user's nearest neighbor request, the LSP returns the nearest POI, the distance of farthest Voronoi neighbor and *count* along with the signature of the data owner. In addition, the LSP returns the locations of those POIs that have a smaller distance than the farthest Voronoi neighbor from the POI and have not been already sent to the user before. The user verifies the signature to ensure the reliability of the returned nearest POI and checks other additional information to verify that the returned POI is the actual nearest one with respect to the user's false location. The incremental retrieval of POIs and authentication continues until the query answer (of the desired query type: range, NN, or GNN) for the user's actual location has been identified.

In summary, the contributions of our paper are as follows:

- We propose an authentication technique that incurs less communication and storage overhead for authenticating multiple POIs retrieved from the LSP, which forms the base of processing privacy preserving LBSs.
- We develop a unified framework to process a wide range of privacy preserving LBSs, range, nearest neighbor, group nearest neighbor queries with authentication.
- We conduct experiments to validate the effectiveness and efficiency of our proposed approach.

The remainder of the paper is organized as follows. In Section 2, we discuss existing literature related to authentication techniques and privacy preserving approaches. In Section 3, we present our approach for authenticating answers of variant privacy preserving location-based queries. Section 4 shows experimental results and Section 5 concludes the paper with future research directions.

2. RELATED WORK

In Sections 2.1 and 2.2, we present existing authentication techniques and privacy preserving approaches for LBSs, respectively.

2.1 Authentication Techniques

Existing authentication techniques mostly use the Merkle R-trees (MR-trees) [14, 15] or Voronoi diagrams [9, 10] to authenticate LBSs. However, neither of these techniques take a user's privacy into account. In MR-tree based approaches [14, 15], a hash digest is stored with the information stored in every leaf node of the R-tree, where a hash digest represents the concatenation of binary representation of information stored in the leaf node. The hash digest stored in an intermediary node of the R-tree represents the summarization of MBRs of child nodes and hash digest stored in the child nodes. The data owner signs the has digest of the root node. When a user requests a location based query, in addition to the query answer, the LSP constructs a verification object from which the user can reconstruct the hash digest of the root node, and compare it against the hash digest signed by the data owner and thereby, confirms the reliability of the answer, i.e., the POI data included in the answer belongs to the data owner. In addition, the verification object includes information from which the user can verify the correctness of the answer. However, the MR-tree based authentication techniques are not suitable if the answer of a location-based query includes multiple POIs due to its high processing overhead [9]. Our proposed unified framework for privacy preserving LBSs requires multiple POI retrieval from the LSP and therefore we cannot adopt the MR-tree based authentication techniques.

In Voronoi diagrams based authentication techniques [9, 10], the data owner computes Voronoi neighbors of every POI and shares the signed POI information that includes the POI location, the count of its Voronoi neighbors, the location information of Voronoi neighbors, with the LSP. In the LSP's database, this signed POI information is stored on the leaf node of the R-tree. When a user requests an LBS, the LSP evaluates the answer and returns additional signed information (i.e., the count and locations of Voronoi neighbors) along with the actual answer. From the signature and locations of the Voronoi neighbors, the user verifies the reliability and the correctness of the query answer. The limitation of these techniques is that they incur high storage overhead because they store information of Voronoi neighbors for every POI. Further, we need to retrieve multiple POIs from the LSP to process LBSs in a privacy preserving manner. Since multiple POIs can share same Voronoi neighbors, this would also incur high communication overhead because of sending the same information multiple times to the user for authenticating different POIs. We develop a Voronoi neighbor based technique, where the information of Voronoi neighbors are not stored in the LSP's database and same authentication information is not sent multiple times to the user.

In [8, 1], the authors have focused on authenticating a query answer and protecting privacy of the stored data in the LSP's database, while our approach aims to protect privacy of users' locations instead of protecting privacy of returned POI data to the user.

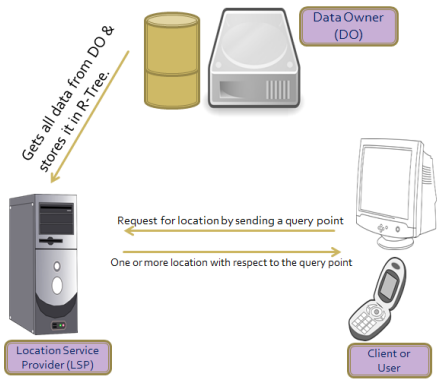

Figure 1: System Overview

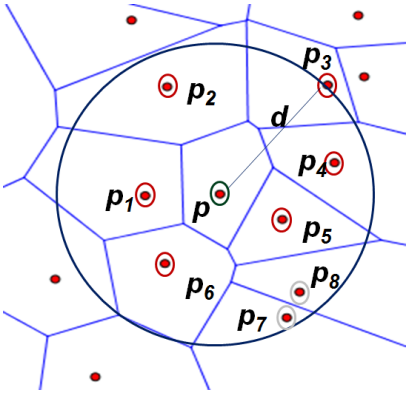

Figure 2: Voronoi neighbors of a POI p

2.2 Privacy Preserving LBSs

There exist a large number of approaches on protecting user privacy while accessing LBSs that include k-anonymity [2, 6], spatial cloaking [7], false location [3, 16, 11, 17] and cryptography [4]. None of these privacy preserving approaches consider authenticating the answers of location-based queries. In this paper, we develop an authentication technique to verify the reliability and correctness of query answer while protecting privacy of a user's location using the privacy model based on false locations [3, 16, 11, 17].

3. OUR APPROACH

We develop a novel authentication technique for processing variants of LBSs based on incremental retrieval of POIs from the LSP. In our approach, the data owner (DO) shares signed locations of POIs along with additional parameters related to POIs with the location-based service provider (LSP). The LSP indexes the received data using an R-tree. When a user requires to access an LBS, the user incrementally requests nearest POIs with respect to a false location instead of her actual location for privacy reason. The LSP returns the locations of nearest POIs, other parameters related to POIs received from the data owner, and the signature of the data owner to the LSP. From the signature, the user can verify the reliability, i.e., whether the POI information comes from the data owner. In addition, the LSP returns locations of other nearby POIs from which the user verifies the correctness of the answer, i.e., the POI is the actual nearest ones with respect to the false location. The incremental retrieval of nearest POIs with respect to a false location continues until the authenticated query answer with respect to the user's actual location has been identified.

Figure 1 shows the steps involved in our proposed system. In Sections 3.1, 3.2, and 3.3, we present the detailed description of these steps, respectively.

3.1 The Data Owner

The data owner has location information of all POIs. Before sharing this information with the LSP, the data owner constructs an object for every POI that consists of three parameters: (i) the location of a POI, (ii) the distance of the POI's farthest Voronoi neighbor from the POI, and (iii) *count*, i.e., the number of other POIs that are in less distance than the farthest Voronoi neighbor.

Figure 2 shows an example, where a POI p has 6 Voronoi neighbors, p_1, p_2, p_3, p_4, p_5, and p_6. The parameter d represents the distance between p and its farthest Voronoi neighbor p_3. There are 8 POIs whose distances from p are less than or equal to d (shown in Figure 2). Thus, the DO constructs an object for p as $\{p|d|8\}$. Then the DO signs the object. The signature is essential to ensure the reliability of the POI information. For the signature generation, an RSA cryptography and SHA-1 algorithm are used. The user verifies the signature with the public key of the data owner and confirms that the object has been constructed by the DO.

3.2 LSP-Side Processing

After receiving the POI dataset from the data owner, the LSP indexes them using an R-tree. For each POI, the POI location, the distance of the farthest Voronoi neighbor from the POI, *count*, and the signature of the DO are stored in the leaf node of the R-tree. When a user requests nearest POIs for the first time, the user provides a query location (i.e., a false location) q and the number of nearest POIs. For the subsequent requests, the user only specifies the number of required nearest POIs. For example, for the first request, if the user's required number of nearest POIs is 3, the LSP returns 1^{st}, 2^{nd}, and 3^{rd} nearest POIs with respect to q. For the second request, if the user's required number of nearest POIs is 2, the LSP returns 4^{th}, and 5^{th} nearest POIs with respect to q. The user continues to request nearest POIs until the user has identified the query answer with respect to her actual location.

To retrieve nearest POIs incrementally, our approach uses best first search (BFS) [5]. For authentication purposes, the LSP needs to send additional information to the user. For every returned POI p to the user, the LSP also provides the distance of the farthest Voronoi neighbor from the POI, *count*, and the signature of the DO. In addition, for authentication purposes, the LSP sends the information of POIs (i.e., the POI location, the distance of the farthest Voronoi neighbor from the POI, *count*, and the signature of the DO) whose distances are less than or equal to the distance between p and the farthest Voronoi neighbor of p. When a user receives this information the user verifies whether the number of these POIs sent by the LSP equals to *count* signed by the DO for authentication purpose (a detail discussion is given in Section 3.3).

Retrieving the additional POI information from the database to authenticate the correctness of every nearest POI independently would incur high processing overhead as a single POI might be retrieved multiple times for authenticating multiple nearest POIs. Our proposed approach avoids a repeated retrieval of POI data and thereby reduces computational and communication overhead.

To retrieve the required POI information for authentication purpose in a single traversal, the LSP first computes the *required maximum distance* from q. Without loss of generality, Figure 3 shows an example to compute the required maximum distance. In Figure 3,

a user requests 3 nearest POIs with respect to q. The LSP first retrieves 3 nearest POIs p_1, p_2, and p_3 and the distances from their Voronoi nearest neighbors, *count*s, and the signatures. Let d represent the distance from q to a nearest POI p_i for $i \in 1,2,3$ and f represent the distance from p_i to its farthest Voronoi neighbor. Figure 3 shows that $d+f$ for p_2 is the maximum among 3 POIs p_1, p_2, and p_3 and the required maximum distance is computed as $d+f$ for p_2. Thus, to ensure that all POIs required to authenticate p_1, p_2, and p_3, the LSP retrieves information of the POIs whose distances are less or equal to the required computed maximum distance. The LSP sends information (i.e., the POI location, the distance of the farthest Voronoi neighbor from the POI, *count*, and the signature of the DO) of p_1, p_2, p_3, and other retrieved POIs to the user.

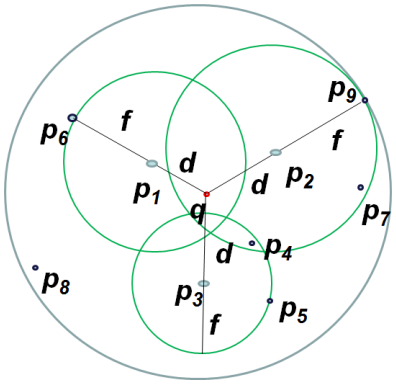

Figure 3: Query processing by the LSP

It may happen that POIs that are not required for the authentication of current nearest POIs (e.g., p_8 is not required to authenticate p_1, p_2, and p_3) have been retrieved and sent to the user. However, the LSP keeps track of the *covered maximum distance*, i.e., the distance of the farthest POI from q that has been already sent to the user and does not send any POI again to the user whose distance from q is less than or equal to the covered maximum distance even if it is required for the authentication of nearest POIs in subsequent user requests.

To avoid retrieval of the same POI multiple times from the database, the LSP always stores the already retrieved POI information from the database in a temporary buffer. For example, if in the next request the user queries the 4^{th} and 5^{th} nearest POIs from q, the 4^{th} and 5^{th} nearest POIs might have been already sent to the user but the additional POIs for authenticating the 4^{th} and 5^{th} nearest POIs may have not been sent to the user. Thus, to compute the required maximum distance for the 4^{th} and 5^{th} nearest POIs, the LSP needs to know the distances to their farthest Voronoi neighbors. In our approach the LSP can get these distances from the temporary buffer and does not need to access database again for this purpose. After computing the required maximum distance, the LSP checks whether it is greater than the covered maximum distance. If yes, the LSP retrieves information of those POIs that have distances from q greater than the covered maximum distance, stores them in the temporary buffer and also sends them to the user. The covered maximum distance is now updated to the required maximum distance. When the user terminates the incremental nearest POI retrieval process, i.e., the query answer with respect to the user's actual location is found, the LSP removes all POI information from the temporary buffer.

3.3 User-Side Processing

As we mentioned earlier, the user incrementally requests nearest POIs with respect to a false location q. After retrieving a number of nearest POIs with additional authentication information, the user checks the reliability of those nearest POIs by verifying the signatures using the public key of the DO. To verify the correctness of a j^{th} nearest POI p, i.e., the distance of p from q is the j^{th} smallest among the distances of all POIs on the LSP's database from q, the user first determines the number of POIs that have less or equal distances to the distance between p and the farthest Voronoi neighbor of p. If the determined number is equal to *count* then the user is confirmed that the LSP has returned all Voronoi neighbors of p. Note that our approach does not store information Voronoi neighbors such as [9], instead our approach only stores the distance of a POI to its farthest Voronoi neighbor and *count*, and thus, reduces storage overhead significantly. Then, our approach computes the distances of q from the retrieved POIs from the LSP and if p has the j^{th} smallest distance among the computed distances, then p is the actual j^{th} nearest POI from p. The following theorem shows the correctness of our claim.

THEOREM 3.1. *If P represents the set of retrieved POIs from the LSP, q represents the query point (i.e., the user's false location), and a POI p has the j^{th} smallest among the distances of POIs in P from q, then p is the actual j^{th} nearest POI from q for $j > 0$.*

PROOF. (By contradiction)
Let an LSP send a false j^{th} nearest neighbor of q, p', which has the j^{th} smallest distance among the distances of POIs in P from q. According to our approach, the LSP has also sent the Voronoi neighbors of p' to the user, otherwise *count* must not have been verified. For $j = 1$, since p' is not the actual 1^{st} nearest POI, one of its Voronoi neighbors, which are included in P, must have less distance from q than that of p'. Thus, for $j = 1$, p' does not have the smallest distance among the distances of POIs in P from q, which contradicts our assumption.

For $j > 1$, we assume that $j-1$ nearest POIs with respect to q have been authenticated. Similar to the case of $j = 1$, since p' is not the actual j^{th} nearest POI, one of its Voronoi neighbors, which are included in P, must have distance from q less than that of p' and greater than those for authenticated $j-1$ nearest POIs. Thus, for $j > 1$, p' does not have the j^{th} smallest distance among the distances of POIs in P from q, which again contradicts our assumption. Thus, a POI that is the j^{th} nearest POI of q has the j^{th} smallest among the distances of POIs in P from q. □

After authenticating the reliability and the correctness of the nearest POIs, our approach computes an *authenticated known region*. An authenticated known region represents the area, where location of all POIs are known and the POIs are authenticated. The authenticated known region is a circle centered at q and having radius equal to the distance of the farthest authenticated nearest POI from q. The authenticated known region expands with the incremental retrieval of nearest POIs.

In the following subsections, we discuss how through the use of the authenticated known region a user determines the query answer with respect to her actual location l for range, nearest neighbor (NN), and group nearest neighbor (GNN) queries.

3.3.1 Range Queries

A range query returns locations of all POIs within a given range. In a privacy preserving range query, a user retrieves nearest POIs with respect to a false location and the authenticated known region gradually expands. If the authenticated known region covers the user's specified range then it is guaranteed that all POIs within the range have been identified and authenticated.

3.3.2 Nearest Neighbor Queries

In [16], the authors have shown that if a circle centered at the user's location l and having a radius equal to the distance between l and a POI p does not contain any other POI, then p is the nearest neighbor of l. Thus, similar to the technique proposed in [16], our approach identifies the POI with the smallest distance from l among all POIs and computes the circle for that POI. If the authenticated known region covers the circle, then it is guaranteed that p is the authenticated nearest neighbor with respect to the user's actual location l. If not, then the user continues to retrieve more nearest POIs from the LSP until the authenticated nearest neighbor has been identified.

3.3.3 Group Nearest Neighbor Queries

A GNN query returns the location of a POI that minimizes the aggregate distance with respect to the group members. In [13], the authors have shown that if a circle centered at the geometric centroid of the locations of group members and having a radius equal to the average distance of a POI p from the group members does not contain any other POI, then p is the group nearest neighbor. In our approach, the user computes the average distance of retrieved POIs from the group members, determines the POI with the smallest average distance and computes the circle with the center at the geometric centroid of the locations of group members and radius equal to the smallest average distance. If the authenticated known region covers such a circle then it is guaranteed that the corresponding POI is the authenticated group nearest neighbor. On the other hand, if the authenticated known region does not include the circle, the user requests more nearest POIs from the LSP until the authenticated group nearest neighbor has been identified.

3.3.4 Privacy Analysis

The LSP can also compute the authenticated known region in a similar fashion to the user, and knows the algorithms used by the user to determine the answers for range queries, nearest neighbor queries and group nearest neighbor queries with respect to the user's actual location. Thus, the LSP can reverse engineer and refine the user's location within a region. The larger the authenticated region the larger is the imprecision for the predicted location of the user. Thus, to increase the privacy level, even if the query answer for the user's actual location has been identified, a user may continue to retrieve more POIs from the LSP to expands the authenticated known region.

4. EXPERIMENTAL STUDY

In this section we evaluate the performance of our proposed incremental algorithm for processing range and nearest neighbor queries that protects user privacy and ensures authenticity of the query answer. We name this approach, Incremental Authenticated Privacy-preserving (IAP) approach. Since we are the first to address the authentication for privacy preserved location based services (e.g., range, NN queries), we have adopted the best known technique for authenticating for LBSs [9] for our experimental comparison. In this approach, to preserve the privacy, a user needs to send multiple NN requests to the server and the server evaluates each of these requests independently. We name this approach as Base Authenticated Privacy-preserving (BAP) approach.

We simulate our experiments on a system with an Intel Core i5 2.67 GHz processor and 4 GB of memory running Windows 7.

We use a synthetic dataset of 10,000 data objects generated using uniform random distribution. We also generate queries using uniform random distribution. For each set of experiment, we run 50 queries and then present the average results. We measure the storage overhead and processing time as our performance metric. For measuring the storage overhead, we compute the storage cost in bytes for each object. For processing time, in both IAP and BAP approaches, the server us BFS technique for NN search, we only compare the processing time at the user device as our performance metric.

Table 1: Storage overhead of the existing approach

Object Attribute	Byte
location	16
neighbor number	4
neighbours (avg)	6*16
signature	128
size per object	244

Table 2: Storage overhead of our approach

Object Attribute	Byte
location	16
d	8
count	4
signature	128
size per object	156

4.1 Storage Overhead

We compare the storage cost of our proposed approach, IAP with the existing approach BAP. Table 1 and Table 2 show the storage cost in bytes in BAP and IAP approaches, respectively. We observe that BAP takes 56% more storage than IAP to store a single object. The reason is as follows. BAP needs to store all the neighbors' of an object, whereas, IAP only needs to store the *count* of neighbors. Thus, we can see that BAP needs 244 bytes, whereas IAP needs 156 bytes to store an object.

4.2 Computational and Communication Overhead

In this set of experiments, we measure the computational and communication overhead for processing range and NN queries. To measure the computational overhead, we measure the processing time (in milliseconds) of the client to find the answer of the query. To measure the communication overhead, we measure the number of bytes that need to be transferred from the server to the client.

4.2.1 Range Queries

In this set of experiments, we vary the query range as 200x200, 400x400, 600x600, 800x800, and 1000x100 areas in the dataspace. We measure the processing time (Figure 4) and communication

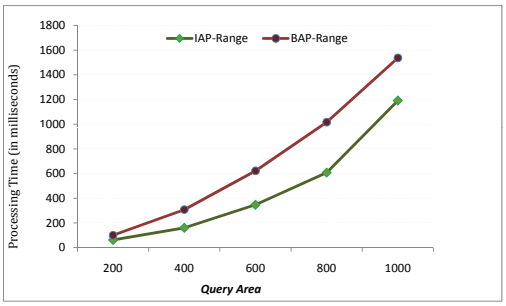

Figure 4: Processing time for range queries

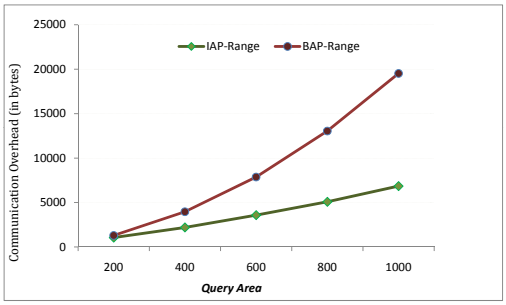

Figure 5: Communication overhead for range queries

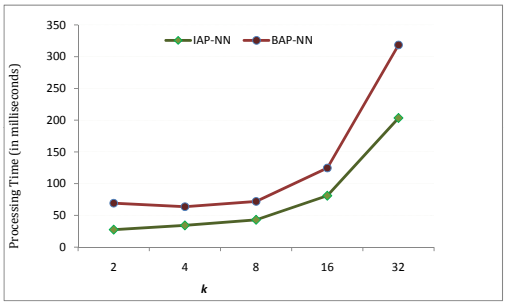

Figure 6: Processing time for NN queries

overhead (Figure 5) for processing and authenticating privacy preserving range using BAP and IAP approaches. In Figure 4, We observe that BAP requires on average 65% more processing time than IAP. We also see from Figure 5 that BAP requires more than a double number of bytes than IAP. Figure also shows that our approach IAP outperforms BAP in a greater margin for an increased query range area.

Next, we present the processing time and communication overhead of range and NN queries.

4.2.2 *NN Queries*

In this set of experiments, we measure the performance of BAP and IAP for processing NN queries. Figure 6 and Figure 7 show the processing time (in milliseconds) and the communication overhead (in bytes), respectively, of two approaches by varying k as 2, 4, 8, 16, and 32.

Figure 6 shows that the processing time of IAP is on average 73% less than that of BAP. We also observe in Figure 7 that IAP needs 39% (on average) less number of bytes transfer than BAP. Figure also shows that our approach outperforms BAP by a greater margin for a larger value of k.

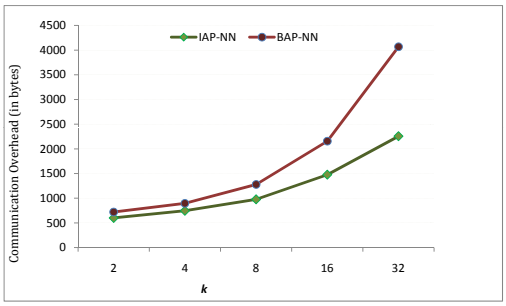

Figure 7: Communication overhead for NN queries

5. CONCLUSION

In this paper, we propose the first approach that considers both privacy of users and the authenticity of answers for processing LBSs. We present a uniform framework to authenticate the answers for a wide range of location-based queries, range, nearest neighbor, and group nearest neighbor queries in a privacy preserving manner. In contrast to existing authentication technique [9], our approach does not need to store the information of Voronoi neighbors of every POI in the LSP'database and thus, reduces the storage overhead significantly. Furthermore, n our authentication approach, the LSP does not send the same POI information required to authenticate more than one POI multiple times and thereby, reduces communication overhead.

In future, we will consider authenticating other type of privacy preserving LBSs such as trip planning queries. We also aim to focus on authenticating LBSs using other privacy models that include k-anonymity, spatial cloaking and cryptography.

6. REFERENCES

[1] Q. Chen, H. Hu, and J. Xu. Authenticating top-k queries in location-based services with confidentiality. *PVLDB*, 7(1):49–60, 2013.

[2] C. Chow, M. F. Mokbel, and W. G. Aref. Casper*: Query processing for location services without compromising privacy. *ACM Trans. Database Syst.*, 34(4), 2009.

[3] M. Duckham and L. Kulik. A formal model of obfuscation and negotiation for location privacy. In *PERVASIVE*, pages 152–170, 2005.

[4] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K. Tan. Private queries in location based services: anonymizers are not necessary. In *SIGMOD*, pages 121–132, 2008.

[5] A. Guttman. A dynamic index structure for spatial searching. *SIGMOD*, pages 47–57, 1984.

[6] T. Hashem and L. Kulik. Safeguarding location privacy in wireless ad-hoc networks. In *UbiComp*, pages 372–390, 2007.

[7] T. Hashem, L. Kulik, and R. Zhang. Privacy preserving group nearest neighbor queries. In *EDBT*, pages 489–500, 2010.

[8] H. Hu, J. Xu, Q. Chen, and Z. Yang. Authenticating location-based services without compromising location privacy. In *SIGMOD*, pages 301–312, 2012.

[9] L. Hu, W. Ku, S. Bakiras, and C. Shahabi. Verifying spatial queries using voronoi neighbors. In *ACM SIGSPATIAL*, pages 350–359, 2010.

[10] L. Hu, W. Ku, S. Bakiras, and C. Shahabi. Spatial query integrity with voronoi neighbors. *IEEE Trans. Knowl. Data Eng.*, 25(4):863–876, 2013.

[11] H. Kido, Y. Yanagisawa, and T. Satoh. An anonymous communication technique using dummies for location-based services. In *ICPS*, pages 88–97, 2005.

[12] Microsoft. Location & privacy: Where are we headed?, 2011 (accessed September 2, 2011). `http://www.microsoft.com/privacy/dpd`.

[13] S. Namnandorj, H. Chen, K. Furuse, and N. Ohbo. Efficient bounds in finding aggregate nearest neighbors. In *DEXA*, pages 693–700, 2008.

[14] S. Papadopoulos, Y. Yang, S. Bakiras, and D. Papadias. Continuous spatial authentication. In *SSTD*, pages 62–79, 2009.

[15] Y. Yang, S. Papadopoulos, D. Papadias, and G. Kollios. Authenticated indexing for outsourced spatial databases. *VLDB Journal*, 18(3):631–648, 2009.

[16] M. L. Yiu, C. S. Jensen, X. Huang, and H. Lu. Spacetwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services. In *ICDE*, pages 366–375, 2008.

[17] M. L. Yiu, C. S. Jensen, J. Møller, and H. Lu. Design and analysis of a ranking approach to private location-based services. *ACM Trans. Database Syst.*, 36(2):10, 2011.