

Enabling Location Privacy; Moving beyond K-anonymity, Cloaking and Anonymizers

Ali Khoshgozaran
University of Southern California, Los Angeles, CA 90089
jafkhosh@usc.edu

1. INTRODUCTION AND MOTIVATION

With many applications, locating dynamic objects is of particular interest. In particular, almost all location-based service (LBS) providers are somehow *aware* of their users' locations in order to provide customized services. However, the implicit assumption that clients are willing to share their private location information with a potentially untrusted location server has become the source of many concerns and in some cases distressing privacy violations.

We provide two approaches aiming at satisfying significantly more stringent privacy guarantees (defined in [2,3]) compared to the current practice in location cloaking and K-anonymity and propose PULSE, Private Queries Using Location Aware Space Encoding and SPIRAL, a Scalable Private Information Retrieval Approach to Location Privacy to protect users' private location information. PULSE utilizes space filling curves as one-way transformations to *encode* both user queries and points of interest to evaluate a query blindly in this transformed space. SPIRAL utilizes *private information retrieval* techniques to achieve location privacy by placing a trusted entity (such as a secure coprocessor), as close as possible to the untrusted host to disguise the selection of desired records while processing a query. Although these approaches are significantly different in how they enable blind evaluation of spatial queries, they share the same architecture, as well as the objective of enabling location privacy in location-based services (LBS).

2. PULSE: PRIVATE QUERIES USING LOCATION AWARE SPACE ENCODING

PULSE enables location privacy by *encoding* static objects and user queries into another space unknown to the untrusted server and privately performing spatial queries in the transformed space.

2.1 Space Encoding

Identifying the right *space encoders* is challenging because the transformation used should respect the distance and proximity while being computationally irreversible by an adversary to be able to make the location server *privacy aware*. Such one-way transformations *encode* the original space into another space which is capable of addressing transformed spatial queries privately.

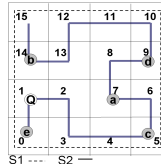


Figure 1: A H_2^2 Pass

We show how locality preserving Hilbert Curves can be treated as space encoders used to perform a spatial query in an encoded space if certain properties of those curves are kept secret from malicious server and are only provided to the users [2]. Let H_d^N for $N \geq 1$ and $d \geq 2$, be the N^{th} order Hilbert curve for a d -dimensional space. H_d^N is a linear ordering which maps an integer set $[0, 2^{Nd} - 1]$ into a d -dimensional integer space $[0, 2^N - 1]^d$: $H = \nu(P)$ for $H \in [0, 2^{Nd} - 1]$, where P is the coordinate of each point in the d -dimensional space. We call the output of this function its *H-value* throughout the paper. An important property of a Hilbert curve that makes it a very suitable tool for our proposed scheme is that ν can be viewed as a one-way function if the curve parameters are not known. These parameters, which collectively form a *key*, include the curve order N , starting point (X_0, Y_0) , orientation θ and scale factor Γ . We term this key, *Space Decoding Key* or *SDK* where $SDK = \{X_0, Y_0, \theta, N, \Gamma\}$. To perform an exhaustive search, the attacker has to obtain the *precise SDK* value. In [2] we proved that PULSE makes it impossible for a computationally bounded server to find *SDK* and decode the mapping.

2.2 Private Spatial Queries

Making a query processing engine privacy-aware based on our idea of space encoding discussed above, requires an *offline* space encoding phase where the curve parameters from which the curve is constructed are chosen and the value of *SDK* is determined. Next, assuming the entire area covering all points of interest is a square S_1 , a H_2^N Hilbert curve is constructed starting from (X_0, Y_0) in a (possibly larger) square S_2 surrounding S_1 until the entire S_2 is traversed (see Figure 1). After visiting each point P , its H-value $= \nu(P.X, P.Y)$ is computed using *SDK*. This process is performed once for all points of interest and thus at the end of this step, a look-up table *LUT* which consists of H-values for all POIs is constructed. Note that the size of *LUT* only depends on the number of POIs to be indexed and not the size of the region.

During the *online KNN query processing*, users compute $H = \nu(X_q, Y_q)$ for their query point (X_q, Y_q) using *SDK* and send H , along with K (i.e., the number of desired nearest neighbors) to the server. The location server then searches from both directions in *LUT* starting from $\nu(X_q, Y_q)$ until K closest matches are found. Note that these matches are nothing but K (distinct or overlapping) encoded objects whose H-values are part of the result set.

We are currently studying *online range query processing*. We de-

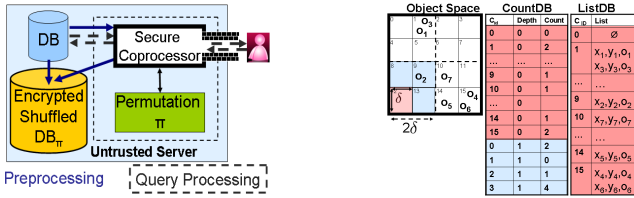


Figure 2: SPIRAL Architecture (left), Grid Structures (right)

compose a 2-D range query $w(x, y, n_1, n_2)$ into a set of 1-D ranges in the transformed space. The challenge is how to do this efficiently and privately. We recursively decompose a range query into a set of square blocks until a partition is completely contained in the range query. Each of these *maximal quadtree blocks* [1] form a set of continuously increasing H-value sequences termed *runs*. We efficiently find the start and end values of a run and consequently, all the H-values in between. A range query is then privately decomposed into a set of runs each sent separately to the location server to retrieve objects whose H-values belong to the run. Knowing *SDK*, users transform the result set back to the original 2-D space, using ν^{-1} to retrieve the original object locations.

3. SPIRAL: A SCALABLE PIR APPROACH TO LOCATION PRIVACY

While *PULSE* utilizes space transformation techniques to enable location privacy, our more recent work dubbed *SPIRAL* which is a *Scalable Private Information Retrieval Approach to Location Privacy* [3] takes a radically different approach to utilize private information retrieval (PIR) techniques to enable location privacy.

3.1 Private Information Retrieval

There is a wide spectrum of scenarios in which a user needs to gain access to a specific record of a database but does not want to reveal the record in which he is interested. More formally, a Private Information Retrieval (PIR) protocol allows a user to retrieve the i th record from a database of size n stored at an untrusted server, without revealing i to the server. The class of PIR approaches can be roughly divided into cryptographic and hardware-based approaches. While cryptographic approaches make use of homomorphic encryption, quadratic residues and other cryptographic properties to achieve PIR, hardware-based techniques utilize a secure coprocessor which acts as a securely protected computing space residing at the untrusted *host* machine that enables private querying of the data. The idea behind using a secure coprocessor for PIR is to place a trusted entity as close as possible to the untrusted host to disguise the selection of desired records within a black box.

Building on the PIR scheme discussed, we enable blind evaluation of spatial queries. The key idea is to perform the query processing on a trusted computing environment while the required data to perform the query is being privately queried from an untrusted data owner (Figure 2-left). Therefore, users' private location information is protected as the server does not learn any information about users' queries. What distinguishes our work from the use of encrypted databases is the impossibility of blindly evaluating a sophisticated spatial query on an encrypted database without a linear scan of all encrypted items.

During a *preprocessing phase*, the server needs to index the objects and perform as much preprocessing as possible in order to achieve minimum query response and communication time. Furthermore, as these data structures are privately kept at the server, they should allow efficient pruning of the space based on how they have stored

object information. The challenge here is to minimize the required number of private retrievals from the database. Figure 2 (right) illustrates our underlying grid structure created during the offline phase. The grid index uniformly partitions the unit square into cells with side length δ ($0 < \delta < 1$). These cells are then used to construct *listDB* storing the exact objects locations and *countDB* which represents hierarchically the number of objects in each *super cell*. A super cell can be a regular cell as defined before or can be constructed by recursively grouping several lower level cells.

3.2 Private Spatial Queries

Using the proposed index structures, upon receiving a *range query*, the secure coprocessor computes the cells overlapping with the range query and privately queries the database's *listDB* to retrieve the set of encrypted objects located in each cell and sends the encrypted results back to the user.

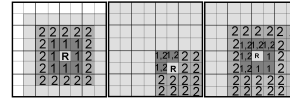


Figure 3: Progressive, Hierarchical and Hybrid Expansion Strategies

We are currently studying three variants of evaluating *K-nearest neighbor queries*. All algorithms utilize the following general approach to respond to a query: (i.) create a rectangular region R and set it to the cell containing the query point q (ii.) expand R until it encloses K objects using *countDB*, and (iii.) expand R to

R' that is the region guaranteed to enclose the actual query results. Note that the main difference among the proposed algorithms is how to perform the step (ii) mentioned above. Using a *Progressive Expansion* algorithm, we incrementally grow a spiral region around the query point while in a *Hierarchical Expansion* approach at each step we move one level higher in the parent hierarchy until R encloses K objects. Finally, the *Hybrid Expansion* approach blends the above expansion strategies by alternating between them. Figure 3 illustrates the number(s) in each cell representing the step(s) at which a cell is examined for each variant. For readability, cells examined in step 3 are slightly shaded instead of being numbered.

4. CONCLUSION AND FUTURE WORK

We proposed two different techniques to enable location privacy based on the notion of space encoders and PIR both avoiding an anonymizer as part of the query processing allowing us to support stringent privacy requirements defined in [2,3], compared to hiding a user between $K - 1$ other users (in K -anonymity) or in a larger region (in location cloaking techniques).

We are extensively evaluating each approach based on its efficiency and resilience to attacks. We are also enhancing *PULSE* and *SPIRAL* to support dynamic queries on dynamic objects which is a fundamentally more challenging problem.

5. REFERENCES

- [1] K.-L. Chung, Y.-H. Tsai, and F.-C. Hu. Space-filling approach for fast window query on compressed images. *IEEE Transactions on Image Processing*, 9(12):2109–2116, 2000.
- [2] A. Khoshgozaran and C. Shahabi. Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy. In *SSTD'07, Boston, MA*.
- [3] A. Khoshgozaran, H. Shirani-Mehr, and C. Shahabi. SPIRAL, a scalable private information retrieval approach to location privacy. In *PALMS'08, Beijing, China*.