Spatial Database Outsourcing

Presentation Outline

- Introduction
- System Architecture
- Space Encryption based Privacy Protection
- Spatial Query Integrity Auditing with Dual Space Encryption Keys
- Experimental Validation
- Future Work

Motivation

- Why outsourcing?
  - Network technology advancements
  - Data management cost is five to ten times higher than the initial acquisition costs
  - Economy of scale
  - Companies can concentrate on their main business
- Popularity of location-based services
  - Spatial data for digital maps, points of interest, etc.

Motivation (Cont.)

- Two main challenges
  - Data privacy (e.g., medical records, road vector data (digital maps))
  - Query integrity
- There is no existing solution which can ensure *both privacy and integrity* for outsourced spatial data.

Contributions

- We propose an innovative approach that simultaneously ensures both the privacy and the integrity of outsourced spatial data.
- Space encryption as the basis
- Data replication based integrity auditing
- Supports the most important spatial query types:
  - Range queries
  - *k* nearest neighbor queries

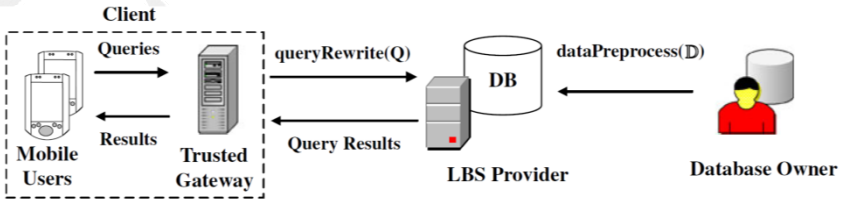
Related Research

- Data privacy protection
  - Executing SQL queries over encrypted databases
  - Existing solutions did not consider the problem of query integrity
- Query Integrity Assurance
  - Results are both correct and complete
  - *Merkle hash tree* based solution – state-of-the-art
  - MR-tree [ICDE'08] supports spatial queries, however it does not consider privacy protection

Presentation Outline

- Introduction
- System Architecture
- Space Encryption based Privacy Protection
- Spatial Query Integrity Auditing with Dual Space Encryption Keys
- Experimental Validation
- Future Work

System Architecture



- Mobile users (e.g., smart phones, tablets, etc.)
- Location-based service providers (could be malicious)
- Database owner (e.g., possessing point of interest data)

System Architecture (Cont.)

- Mobile devices cannot store significant amount of data => submitting queries and analyzing results.
- Assume the DB owner can embed *additional information* in the outsourced DB.
- dataPreprocess(D) => replication and encryption
- queryRewrite(D) => query the encrypted spatial database
- Auditing queries

Presentation Outline

- **Introduction**
- **System Architecture**
- **Space Encryption based Privacy Protection**
- **Spatial Query Integrity Auditing with Dual Space Encryption Keys**
- **Experimental Validation**
- **Future Work**

Space Encryption

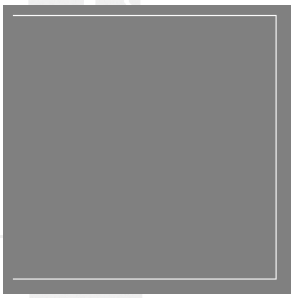
- Employ one-way functions to preserve privacy by encoding the locations of all spatial objects.
- A one-way function is easy to compute but difficult to invert, meaning that some algorithms can compute the function in polynomial time while no probabilistic polynomial-time algorithm can compute an inverse image of the function with better than negligible probability.
- Our solution encrypts the location of every data point.

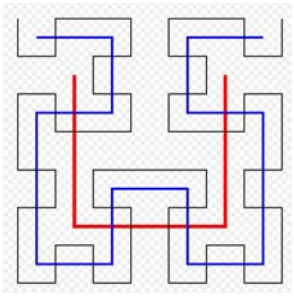
Space Encryption (Cont.)

- Spatial data management => an ideal one-way transformation should respect the spatial proximity of the original space.
- Maintaining the distance properties of the original space => it enables efficient evaluation of spatial queries.
- Space filling curves can be applied as one-way functions for space encryption.

Space Filling Curves

- A space-filling curve (e.g., Z curve, Gray-coded curve, etc.) is a continuous curve, which passes through every point of a closed space.
- We employ the Hilbert curve as an example.

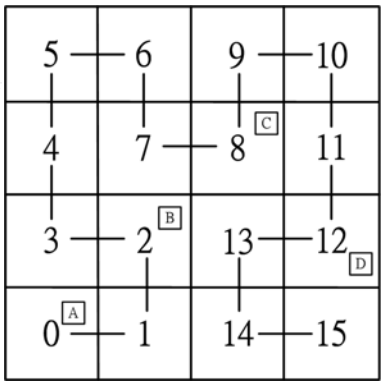




Space Encryption Key

- The curve parameters (e.g., the curve order and orientation) can be applied as ciphers for preserving privacy of outsourced spatial data.
- We can formulate the relationship in a two-dimensional space as $V_{\mathcal{H}} = \mathcal{T}(x, y)$ where x and y are the coordinates of a point in the original space.
- The curve parameters including the curve's starting point (x_0, y_0), curve order O, and curve orientation θ make up the Space Encryption Key (SEK) of the Hilbert curve based one-way function.

Space Encryption Key (Cont.)



Presentation Outline

- Introduction
- System Architecture
- Space Encryption based Privacy Protection
- Spatial Query Integrity Auditing with Dual Space Encryption Keys
- Experimental Validation
- Future Work

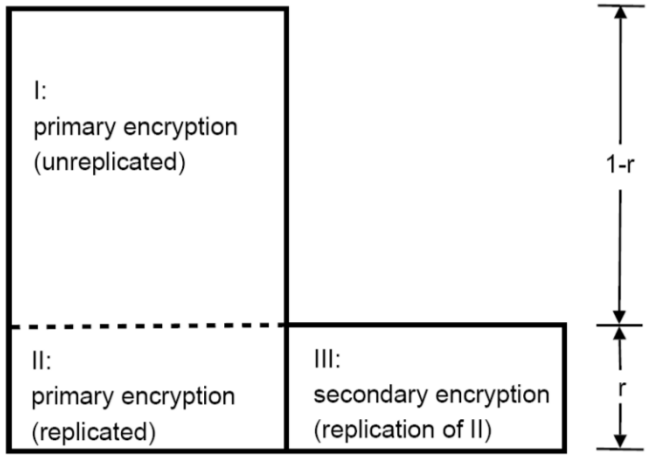
Dual Space Encryption

- Replicate *r* percent of DB and encrypt the duplicate with a secondary encryption key SEK_S which possesses different curve parameters.
- Encrypt the original DB with a primary space encryption key SEK_P.
- Combine the two encrypted datasets and store them at the service provider. The service provider can only see the Hilbert value of each spatial data object.
- For any spatial object *s* in the query result set, a client should be able to verify whether *s* is a valid record of DB and if *s* has a counterpart which is encrypted with another SEK.

Dual Space Encryption (Cont.)

- For supporting object verification, we encrypt the coordinate, non-spatial attributes, and dual information I_d with a symmetric key S_K which is shared by the database owner and all the clients.
- The purpose of the dual information field is for clients to tell if a spatial object has a duplicate in the outsourced database.
- I_d has three values which stand for (i) primary encryption without duplication, (ii) primary encryption with duplication, and (iii) secondary encryption respectively.

Dual Space Encryption (Cont.)



Dual Space Encryption (Cont.)

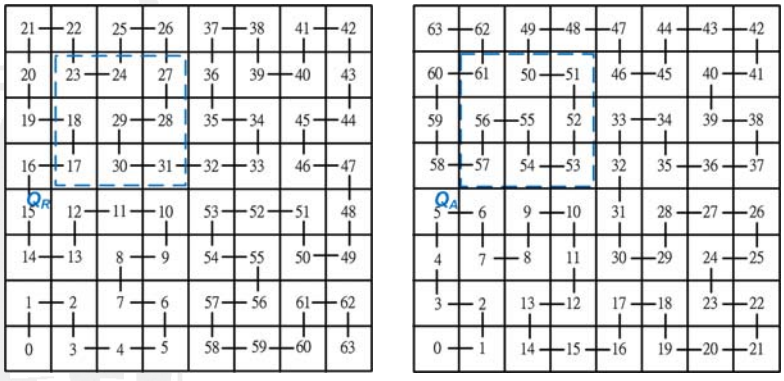
- We apply cryptographic hash functions to generate a signature Ψ for each spatial object with the coordinate and non-spatial attributes as the input message.
- The structure of an encrypted spatial object:

$$s_E = \{V_{\mathcal{H}}, \{x, y, non\text{-}spatial\ attributes, I_d\}_{S_K}, \Psi\}$$

Range Queries

- A client first identifies the Hilbert values covered by the range query based on the parameters of SEK_P.
- The client queries the service provider for retrieving the objects covered by the query range.
- After receiving the query result set R, the client first filters out objects encrypted with SEK_S and verifies the validity of all the remaining objects with their attached signatures.
- If all the objects in R are valid, the client generates an auditing range query Q_A with the same query range size as Q_R and the parameters of SEK_S.

Range Queries (Cont.)



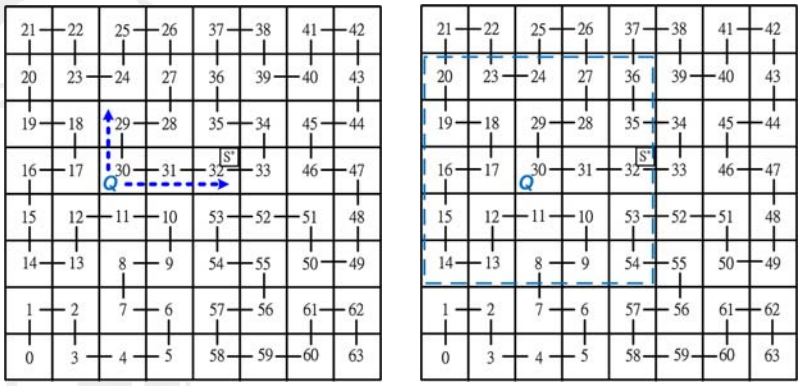
Range Queries (Cont.)

- If the service provider carries out queries honestly, the query result set of the auditing query must contain counterparts of all the objects with duplicates in *R*.
- In practice, the client can launch a single auditing query for verifying a number of regular queries by combining their query ranges for saving resources.

k Nearest Neighbor Queries

- For a given *k*NN query point *Q* located at position (x_Q, y_Q), a client first employs SEK_P to compute $V_{\mathcal{H}} = \mathcal{T}(x_Q, y_Q)$ the query point in the encrypted space.
- Because there is *r* percent duplicate data in D_E which should be filter out from query results, we multiply *k* by (1 + *r*) to get *k'* and apply *k'* as the query parameter.
- The client removes objects encrypted with SEK_S and checks if there are *k* objects leftover in *R*.
- The client retrieves the object s^* which has the longest distance to *Q* in *R*.

k Nearest Neighbor Queries (Cont.)



k Nearest Neighbor Queries (Cont.)

- Because of loss of a dimension in the encrypted space, the objects in *R* may not precisely match the actual *k* nearest neighbors of *Q*.
- The client utilizes the distance between *Q* and s^* (Dist(*Q*, s^*)) as a *search upper bound* and launches a range query Q_R with Dist(*Q*, s^*) to decide the query window size.
- The client audits the range query result.

Attack-aware Auditing Query Composition

- Negligent auditing queries may reveal critical information to allow malicious LBS providers to detect the correspondence among the data with different encryption keys.
- For example, assume a client launches an auditing query after every regular query. Then, a malicious service provider can easily learn the relationship between the two queries and remove the query results of both queries to jeopardize future queries without being detected by clients.

Attack-aware Auditing Query Composition (Cont.)

- Generally, a checking query Q_A, should not leak any correspondence information among the data objects in D_E and Q_A should be hard to differentiate from other regular queries.
- The main principle is to apply a single query to evaluate the integrity of multiple queries.
- Auditing query $\mathcal{Q} = \{q_1, \ldots, q_n\}$

Presentation Outline

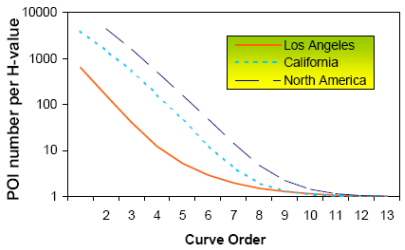
- Introduction
- System Architecture
- Space Encryption based Privacy Protection
- Spatial Query Integrity Auditing with Dual Space Encryption Keys
- Experimental Validation
- Future Work

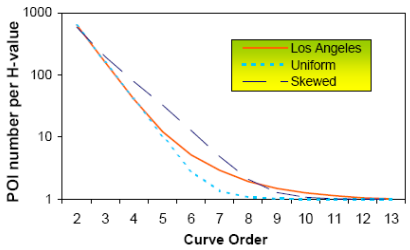
Simulation Datasets

- Synthetic and real-world datasets

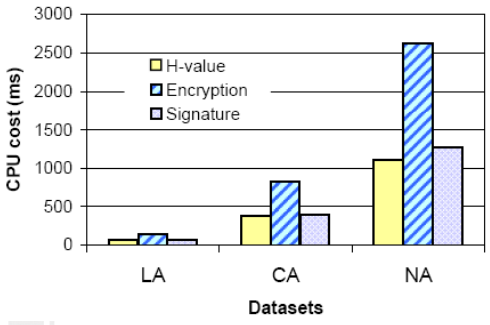
Name	Number of POIs	Source
Uniform	10,163	Synthetic
Skewed	10,163	Synthetic
Los Angeles (LA)	10,163	NAVTEQ
California (CA)	62,556	US Census Bureau
North America (NA)	569,120	US Census Bureau

Encoded POI Density

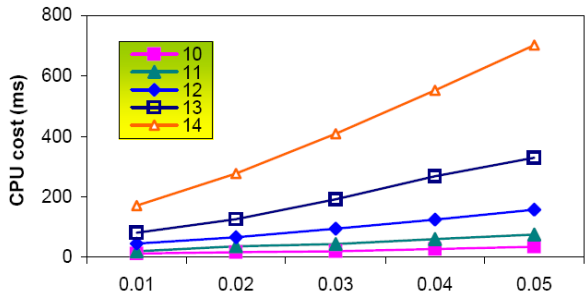




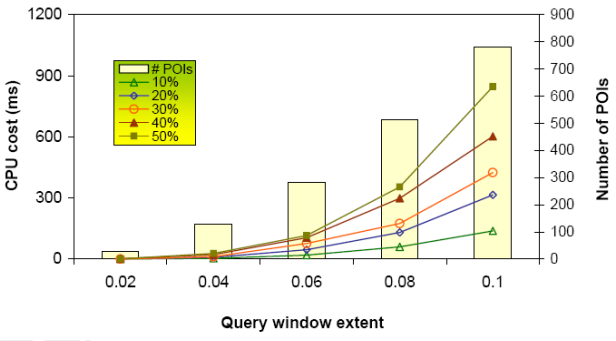
Spatial Database Outsourcing Initialization



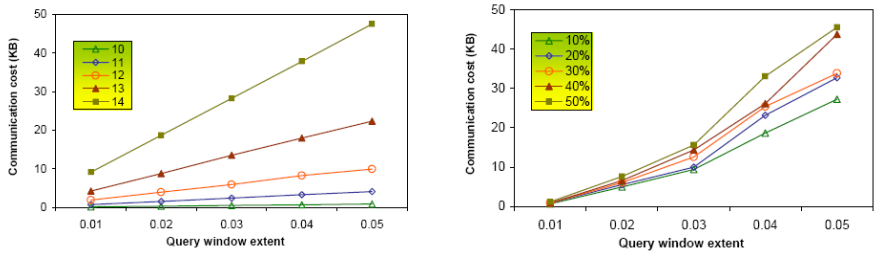
Query Processing on the Client Side



Integrity Auditing



Communication Cost



Against Malicious Attacks