Mobile and Ubiquitous Computing 2022-23
MEIC/METI - Alameda & Tagus

Privacy in Location Based Services

Introduction

- Increased popularity of mobile communication devices with embedded positioning capabilities (e.g., GPS) has generated **unprecedented interest in the development of location-based applications**:
    - browse through maps of their nearby areas and to ask queries about points of interest in their proximity
    - users can generate their own content with geospatial tags
    - location based social networks, or geosocial networks (GSN) allow users to share their whereabouts with friends, to find nearby contacts, and to provide/search for recommendations about points of interest that are close to their current geographical coordinates
    - geospatial crowdsourcing – mobile devices owners can act as mobile sensors (e.g. measure the levels of vehicle traffic congestion, the levels of air pollution, propagate instant information about the weather)
    - study of trajectory traces for mobility planning
- All the above exciting applications benefit from the **availability and potential for sharing of location information**
- However, uncontrolled **location sharing can have dire consequences**, when location data falls in the hands of malicious entities

Location Privacy

- With knowledge of user locations, an adversary can stage a **broad spectrum of attacks** against:
  - individuals, from physical surveillance and stalking, to identity theft, to inferring sensitive information, such as the individual's health status, alternative lifestyles, political and religious affiliations, etc.
- There are **three aspects of location information disclosure**:
  - position awareness,
  - sporadic queries, and
  - location tracking.
- **Position awareness** refers to the case:
  - where a device monitors an individual's location (e.g., an in-car GPS system), but no data is released to another party
  - the user's position is only used locally, to navigate a map for instance, hence no privacy threat occurs.
- The **sporadic (or one-time) queries** case refers to:
  - scenarios where **a user reports his/her current location to a service provider**, in order to find nearby points of interest (e.g., "find the closest restaurant").
- **Location tracking** occurs in:
  - applications that require **frequent updates of the user's position**, e.g., traffic monitoring.

Location Privacy

- Another important aspect in location disclosing is related to the **attacker capabilities**: weak and strong privacy
- **Weak privacy**:
  - requires that **no sensitive data should be *directly* disclosed** to a party that is not trusted
  - i.e. if the current location of the user does not reveal any sensitive information, it is safe to disclose
- **Strong privacy**:
  - **disallows the publication of location snapshots** which, although they do not represent a privacy violation by themselves, may be correlated to additional data to infer the presence of a user at a privacy-sensitive position
  - anonymizing trajectory data is a representative example where strong privacy is necessary
  - enforcing strong privacy must not have a significant negative impact on data accuracy, in the sense that the utility of the published data must be preserved
- These slides:
  - provide an overview of the state-of-the-art in location privacy protection from multiple perspective
  - reviewing the requirements and characteristics of several different location sharing application scenarios
  - solutions range from anonymization techniques using location generalization, to cryptographic techniques, geometric transformations, and differential privacy

Privacy Issues Tackled:

- Can a location based service (LBS) identify my location?
- Can an LBS be able to place me at a sensitive location?
- Can a third-party who accesses an LBS query database reconstruct my trajectory?
- Can a third-party with additional knowledge who accesses an LBS query database reconstruct my trajectory?

Privacy-Preserving Spatial Transformations

Two-Tier Spatial Transformations

Three-Tier Spatial Transformations

Introduction

- To preserve privacy, **the exact location of users that send queries to Location-Based Services (LBS) must not be disclosed**:
  - instead, **location data is first perturbed, and/or encrypted**
  - e.g. generate a few random fake locations and send a number of redundant queries to the LBS to prevent user identification
  - e.g. s*patial k-anonymity (SKA)* is enforced by generating a *Cloaking Region (CR)*—sometimes referred to as Anonymizing Spatial Region (ASR)—which includes the query source as well as $k - 1$ other users
  - e.g. obscure the location data using spatial or cryptographic transformations

- There is a **trade-off** that emerges **between privacy and performance**:
  - Location privacy techniques often degrade application functionality and usability.
  - achieving privacy **incurs an additional overhead in processing queries**
  - e.g. a **larger number of queries** need to be processed in the case of techniques that **generate redundant requests**
  - e.g for **spatial *k*-anonymity** techniques, **query processing is performed with respect to the CR**, which is considerably **more expensive than processing point queries**

Introduction

- We can classify **existing privacy-preserving spatial transformation techniques** into **two categories**, according to the architecture they assume:
    - **two-tier spatial transformations**, and **three-tier spatial transformations**

- **Two-tier** spatial transformations:
    - do not require any trusted third party, and the **query anonymization is performed by the mobile user itself**
    - involve only two parties at query time: the user and the LBS provider
    - assume that no background knowledge is available to the attacker

- **Three-tier** spatial transformations:
    - assume the presence of a **trusted third-party anonymizer server**, and offer better protection against background knowledge attacks (e.g., an attacker may have additional information on user locations, from an external source)

- Trade-off:
    - methods in the **second category generate more runtime overhead**,
    - because they **require the users to constantly update their location with a central authority**, and
    - the algorithms used to generate **protecting cloaking regions are more computationally expensive**

Two-Tier Spatial Transformations

Dummy Locations

- Methods in this category involve only two parties at query time:
  - the **user** and the **LBS provider**
  - assume that **no background knowledge is available to the attacker**
- Simple solution to query privacy:
  - **generate a number of redundant queries for each real query**
  - e.g., user ***u*** **could generate *r* random "fake" locations**, and **send *r* redundant queries to the LBS**, in addition to the actual query containing *u*'s location
- Thus, **dummy locations** are generated such that:
  - the **resulting trajectories mimic realistic movement patterns**
- Dummy-generation algorithms can take into account movement parameters, such as velocity, and certain constraints, e.g., an underlying road network.

Probe (1/5)

Damiani, Maria, Elisa Bertino, and Claudio Silvestri. "PROBE: an obfuscation system for the protection of sensitive location information in LBS." TR2001-145, CERIAS (2008).

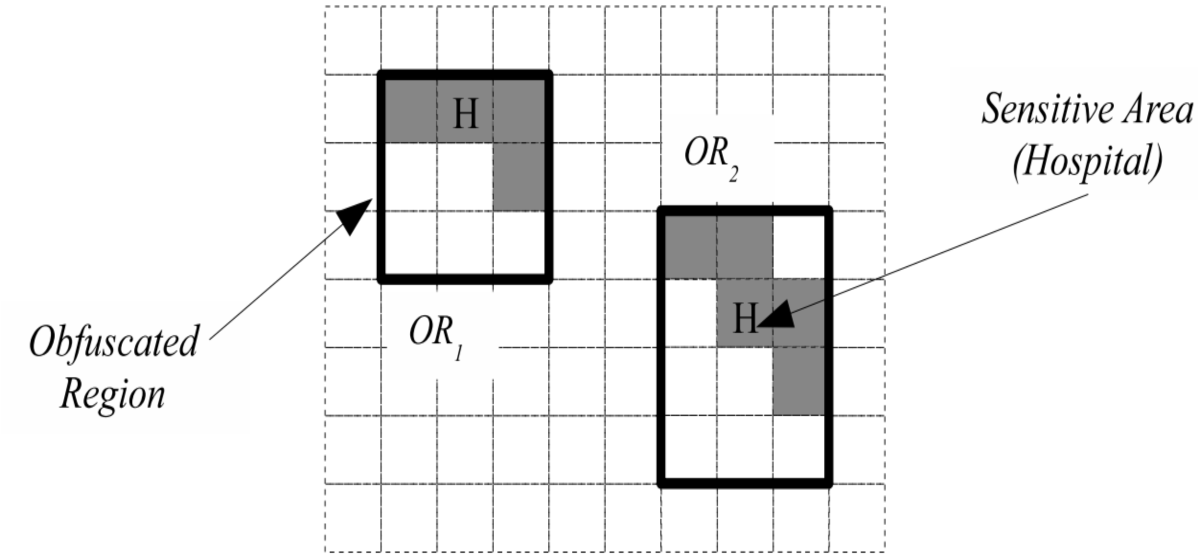
- It **prevents the association between users and sensitive locations**
- It is assumed that the **attacker has access to all sensitive locations from a particular data space** (e.g., a city, a country, etc.)
- **Sensitive locations are represented by *features***, which are **classified into *feature types*** (e.g., hospitals, restaurants, etc.)

- In an **off-line phase**, an ***obfuscated map*** is constructed by:
  - **partitioning the space into a set of disjoint regions** such that
  - the **probability of associating each region with a certain feature type is bounded by a threshold**

- This process is called ***obfuscation***:
  - may require an additional trusted third party, but
  - in the on-line phase (i.e., at query time) PROBE is a two-tier protocol

Probe (2/5)

- For example, ensure that no region can be associated with the "hospital" feature type with probability higher than 44%:
  - *OR*1 contains nine grid cells in total, four of which are sensitive, and 4⁄9 = 0.44



Probe (3/5)

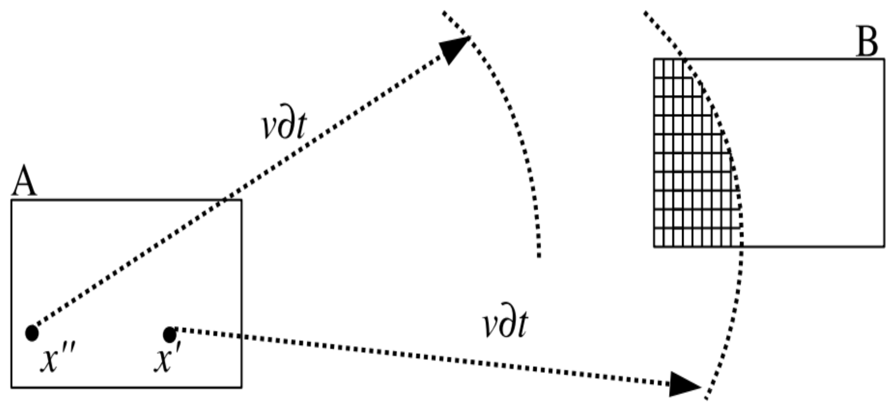
- Another interesting aspect about PROBE is that **it can be extended to protect inference when users move across different obfuscated regions**

- Example:
  - consider the **division of US territory into zip-code areas**
  - the **map is partitioned into disjoint regions**, each of them covering an area of a few square miles; or, at a finer granularity level, a city can be sub-divided into block regions
  - **as the user moves, his/her location can be mapped to a city block identifier, and only the block identifier is disclosed**

- The *privacy requirement* in this case is:
  - **do not to allow an attacker to pinpoint the user location within a sub-region of a reported obfuscated region**

Probe (4/5)

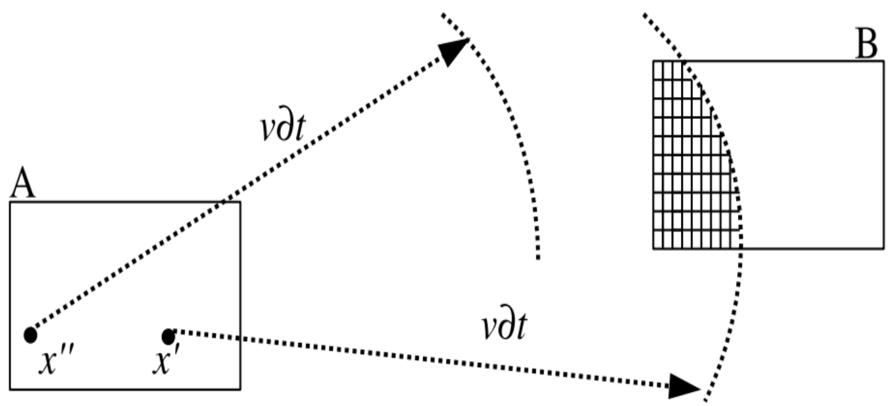
- **Obfuscated regions *A* and *B*** are reported by user **`u`** at timestamps **`ta`** and **`tb`**, respectively where **`ta < tb`**
- Consider **`v` the maximum user velocity**, and let **`δt = |tb - ta|`**
- The **attacker** may try to **prune parts of *A* and *B* to pinpoint *u* in two ways**
- First:
  - determine **if there is any location *x* ∈ *A* from which the user cannot reach some location y ∈ *B***, even by **traveling at maximum speed *v***
  - formally, **an attack is successful iff $\exists x \in A$ *s.t.* $\forall y \in B, d(x, y) > v\delta t$**
  - a user traveling from point *x*' is able to reach a point in the hatched region of *B* within time *δt*; however, **if the initial location of *u* were *x*'', reaching *B* would not have been possible**; therefore, an **attacker can rule out a subset of *A* as possible positions for *u*, hence privacy is breached**



Probe (5/5)

- Second:
  - determine **if there is any location *y* ∈ *B* which the user cannot reach from some initial location *x* ∈ *A*, even by traveling at maximum speed *v***
  - **formally, $\exists y \in B$ s.t. $\forall x \in A$, $d(x, y) > v\delta t$**

- **To prevent privacy breaches**, it is necessary to **ensure that none of the two above equations ever holds**



Limitation

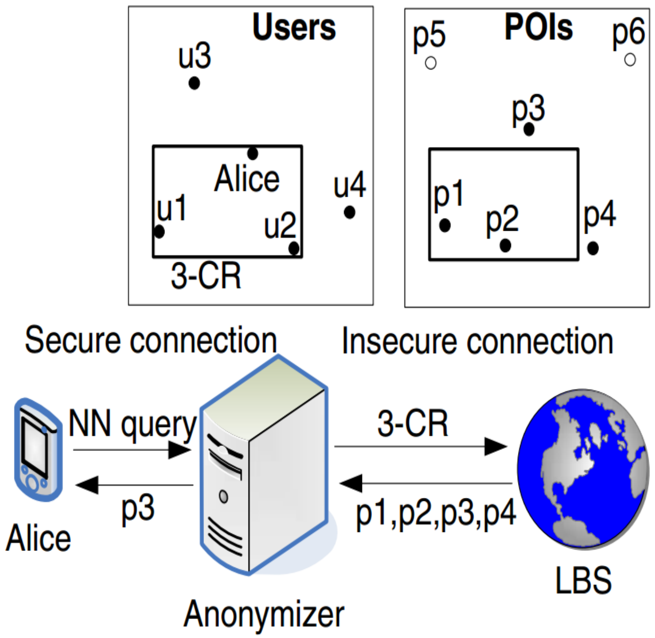
- None of the two-tier spatial transformation solutions can prevent re-identification of the query source if an attacker has knowledge about specific user locations

- Example:
  - if **user *u* situated in a remote location issues a query**, **an attacker who knows that *u* is the only person residing in that area can associate *u* with the query**, breaching user privacy
  - the next category of query anonymization methods deals with this issue

Three-Tier Spatial Transformations

Spatial *k*-anonymity

- Methods in this category implement the **spatial *k*-anonymity paradigm**:
  - A **cloaking region (CR) that contains *k* – 1 users in addition to the query source** is generated, and
  - the **LBS processes the query with respect to the CR**
- Since **all the *k* locations enclosed by the CR correspond to actual users** (as opposed to “fake” locations in the previous category),
  - the **probability to identify the query source is at most 1/*k***, even if the attacker has knowledge about exact user locations
- Most solutions in this category employ a three-tier architecture

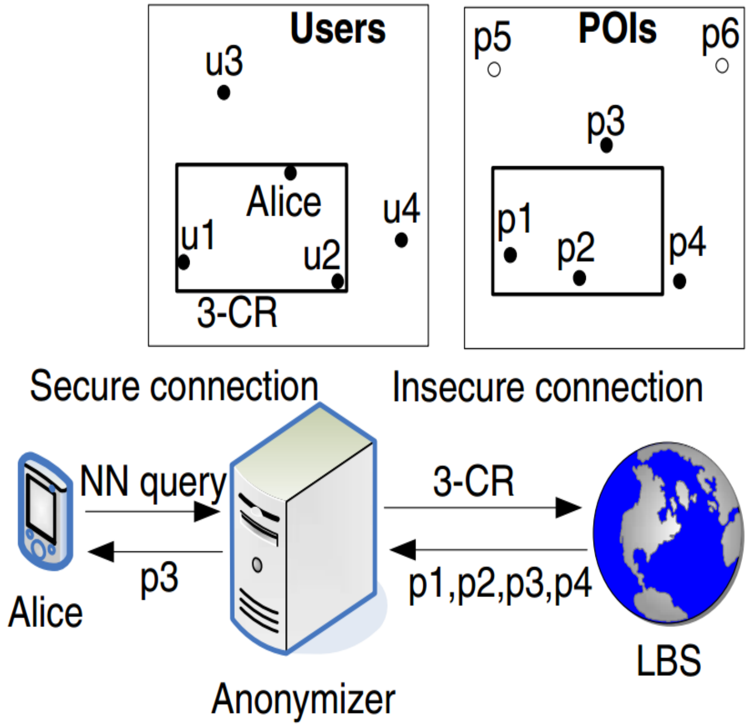


k-anonymity

- ***k*-anonymity** was first discussed **in relational databases**, where **published data** (e.g., census, medical) **should not be linked to specific persons**:
    - methods for **computing aggregate functions** (e.g., *sum, count*) under the condition that **the results do not reveal any specific record**
    - compute **value distributions**, suitable for data mining, in confidential fields
- Recent work has focused on ***k*-anonymity** defined as:
    - **a relation satisfies *k*-anonymity if every tuple is indistinguishable from at least *k*-1 other tuples with respect to a set of *quasi-identifier* attributes**
- **Quasi-identifiers are attributes** (e.g., date of birth, gender, zip code) that **can be linked to publicly available data to identify individuals**
- **Records** with **identical quasi-identifiers** form an **anonymized group**

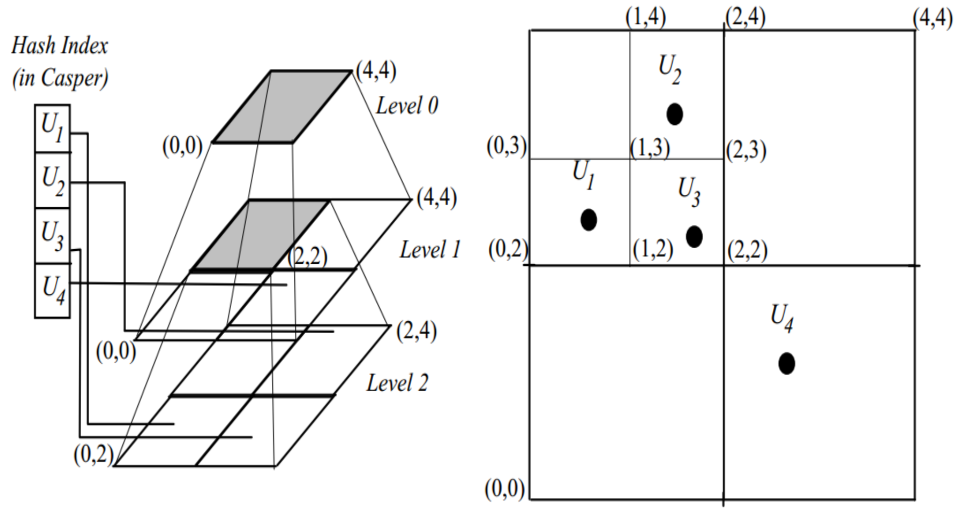
Spatial *k*-anonymity

- A **trusted centralized *anonymizer* acts as an intermediate tier** between the users and the LBS
- All **users subscribe to the anonymizer and continuously report their location** while they move
- Each **user sends his query to the anonymizer, which constructs the appropriate CR and contacts the LBS**
- The **LBS computes the answer based on the CR**, instead of the exact user location; thus, the **response of the LBS is a superset of the answer**
- Finally, the **anonymizer filters the result from the LBS and returns the exact (?) answer to the user**



Interval Cloak

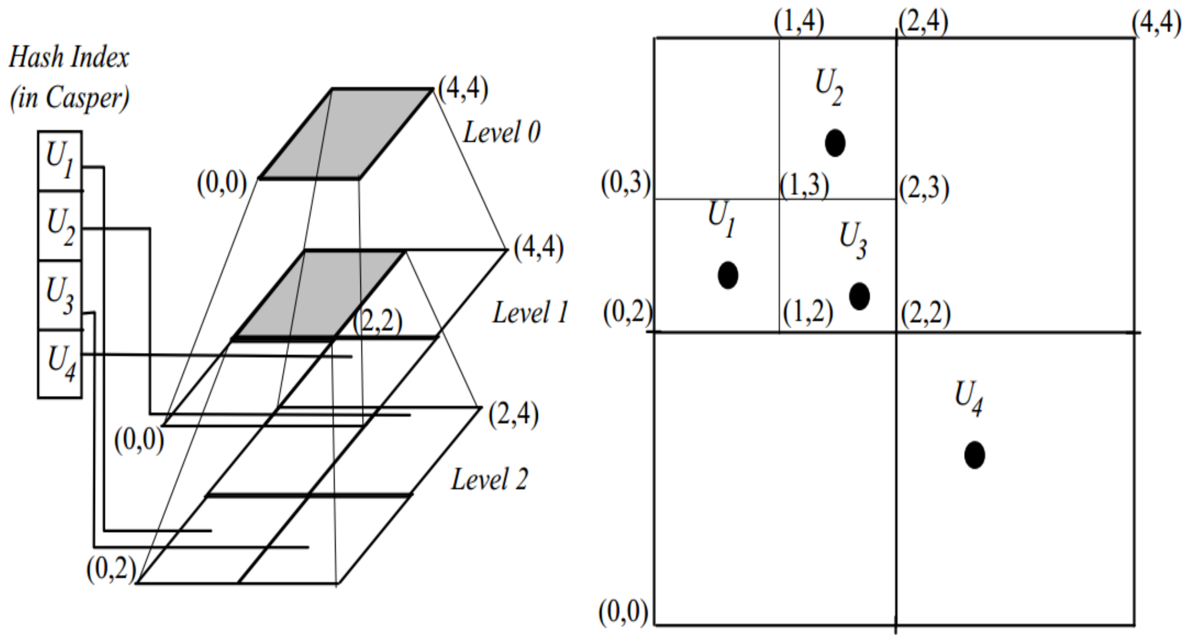
- **Two techniques can be used to transform a relation to a *k*-anonymized one** (both leading to information loss):
  - ***suppression***, where some of the attributes or tuples are removed, and
  - ***generalization***, which involves replacing specific values (e.g., phone number) with more general ones (e.g., only area code)
- ***Interval Cloak*** is based on quadtrees:
  - a quadtree recursively partitions the space in quadrants until the points in each quadrant fit in a page/node
- The anonymizer maintains a quadtree with the locations of all users
- Once it receives a query from a user U, it traverses the quadtree (top-down) until it finds the quadrant that contains U and fewer than k-1 users
- Then, it selects the parent of that quadrant as the k-CR and forwards it to LBS



space partitioning and a simple quadtree assuming that a node contains a single point

Interval Cloak

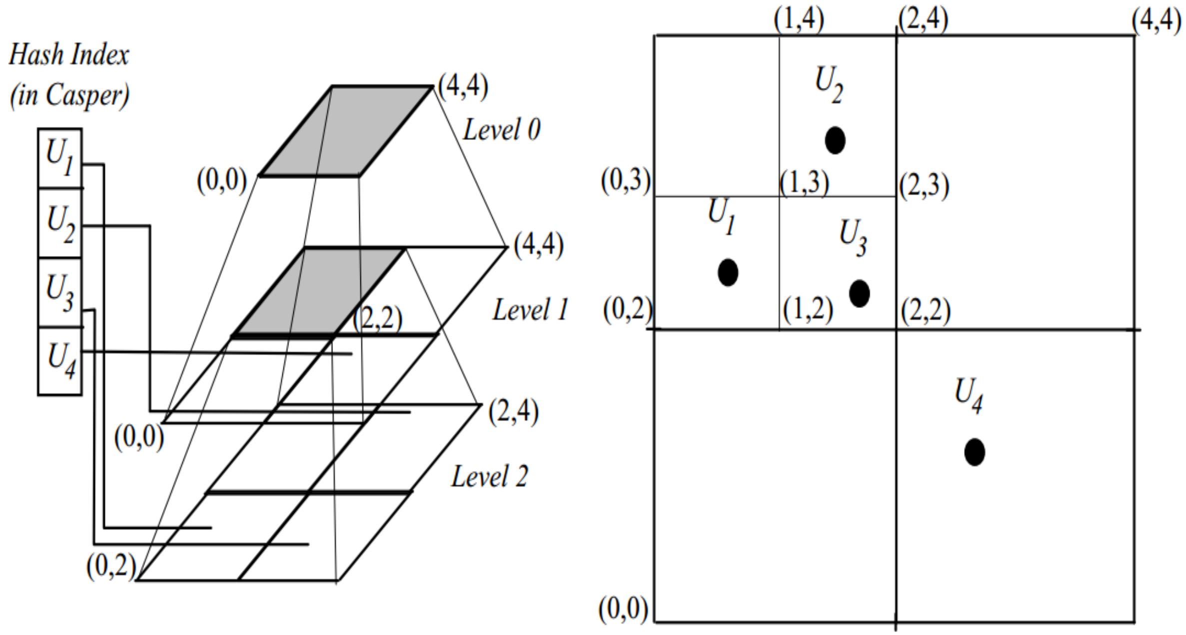
- *Assume that* ***U*****1 issues a query with** ***k*****=2**
- **Quadrant 2 [*(*0*,* 2*)*, *(*1*,* 3*)*] contains only *U*1**, and **its parent [*(*0*,* 2*)*, *(*2*,* 4*)*] becomes the 2-CR**
- Note that the **CR may contain more users than necessary**:
  - in this example **it includes *U*1*,* *U*2*,* *U*3**, although two users would suffice for the privacy requirements



- A **large CR burdens**:
  - the **query processing cost at the LBS**, and
  - the **network overhead for transferring a large number of candidate results** from the LBS to the anonymizer

Casper

- **Similar to *Interval Cloak*, *Casper* is based on quadtrees.**
- The **anonymizer uses a hash table on the user id pointing to the lowest-level quadrant where the user lies.**
- Thus, **each user is located directly**, without having to access the quadtree top-down
- Furthermore**, the quadtree can be adaptive**, i.e., **contain the minimum number of levels that satisfies the privacy requirements**



- e.g., the second level for quadrant [*(*0*,* 2*),* *(*2*,* 4*)*] is never used for $k \geq 2$ and can be omitted

Summary

- Methods in the category of **three-tier spatial transformations rely on the presence of other users to achieve spatial *k*-anonymity**
- Generally, these methods offer **stronger privacy guarantees than two-tier techniques**.
- The privacy features of PROBE and spatial *k*-anonymity methods are not directly comparable:
  - PROBE does not achieve *k*-anonymity, but it does provide spatial diversity
  - on the other hand, three-tier techniques may not always prevent association of users to sensitive locations
  - for instance, it is possible for an entire CR to fall within a sensitive region (e.g., hospital)