PRIVACYGRID: Supporting Anonymous Location Queries in Mobile Environments

Bhuvan Bamba
College of Computing
Georgia Institute of Technology
bhuvan@cc.gatech.edu

Ling Liu
College of Computing
Georgia Institute of Technology
lingliu@cc.gatech.edu

Abstract

We present PRIVACYGRID *– a framework for supporting anonymous location-based queries in mobile information delivery systems. The* PRIVACYGRID *framework offers three unique capabilities. First, we provide a location privacy preference profile model, called location P3P, which allows mobile users to explicitly define their preferred location privacy requirements in terms of both location hiding measures (e.g., location k-anonymity and location l-diversity) and location service quality measures (e.g., maximum spatial resolution and maximum temporal resolution). Second, we develop three fast and effective location cloaking algorithms for providing location k-anonymity and location l-diversity in a mobile environment. The Quad Grid cloaking algorithm is fast but has lower anonymization success rate. The dynamic bottom-up or top-down grid cloaking algorithms provide much higher anonymization success rate and yet are efficient in terms of both time complexity and maintenance cost. Finally, we discuss a hybrid approach that combines the top-down and bottom-up search of location cloaking regions to further lower the average anonymization time. In addition, we argue for incorporating temporal cloaking into the location cloaking process to further increase the success rate of location anonymization. We also discuss the* PRIVACYGRID *mechanisms for anonymous support of range queries. Our experimental evaluation shows that the* PRIVACYGRID *approach can provide optimal location anonymity as defined by per user location P3P without introducing significant performance penalties.*

1 Introduction

With rapid advances in mobile communication technologies and continued price reduction of location tracking devices, location-based services (LBSs) are widely recognized as an important feature of the future computing environment [11]. Though LBSs hold the promise of better safety, more convenience, wider range of entertainment and business opportunities in catering to the growing market of mobile users, the ability to locate mobile users and mobile objects also presents new threats – the intrusion of location privacy [10, 16].

Location privacy is a particular type of information privacy. According to [10], location privacy is defined as the ability to prevent other unauthorized parties from learning ones' current or past location. Location privacy threats refer to the risks that an adversary can obtain unauthorized access to raw location data, derived or computed location information by locating a transmitting device, hijacking the location transmission channel, and identifying the subject (person) using the device [17]. In the United States, privacy risks related to location information have been identified in the Location Privacy Protection Act of 2001 [3]. Many have recognized that without safeguards, extensive deployment of LBSs may open doors for adversaries to jeopardize location privacy of mobile users and to imperil LBSs to significant vulnerabilities for misuse and abuse [12, 16, 25]. For example, location information can be used to spam users with unwanted advertisements or to learn about users' medical conditions, alternative lifestyles or unpopular political or religious views. Inferences can be drawn from visits to clinics, doctors' offices, entertainment clubs or districts, or political events. Public location information can lead to physical harm, such as stalking or domestic abuse.

Several approaches have been proposed for protecting location privacy of a user. Most of them try to prevent disclosure of unnecessary information by techniques that explicitly or implicitly control what information is given to whom and when. We classify these techniques into three categories: (1) Location protection through user-defined or system-supplied privacy policies; (2) Location protection through anonymous usage of information; and (3) Location protection through pseudonymity of user identities, which uses an internal pseudonym rather than the user's actual identity. As described in [10], some location-based services can operate completely anonymously, such as "*when I pass a gas station, alert me with the unit price of the gas*". Others can not work without the user's identity, such as "*when I am inside the office building, let my colleagues find out where I am*". Between these two extremes are those applications that cannot be accessed anonymously but do not require the user's true identity, such as "*when I walk past a computer screen, let me teleport my desktop to it*". For those LBSs that require our true identity, strong security mechanisms, such as location authentication and authorization, have to be enforced in conjunction

with their location privacy policy. In this paper we concentrate on the class of location-based applications that accept pseudonyms and present the PRIVACYGRID framework for performing personalized anonymization of location information through customizable location k-anonymity and enabling anonymous location based queries in mobile information delivery systems.

In the context of LBSs and mobile users, location k-anonymity refers to k-anonymous usage of location information. A subject is considered location k-anonymous if and only if the location information sent from a mobile user to a LBS is indistinguishable from the location information of at least $k-1$ other subjects. A larger k indicates more difficulty in linking a location to a particular user and thus higher guarantees for location privacy. This uncertainty will increase with the increasing value of k. However, the quality of the LBS depends on the accuracy of location of mobile users, and at the same time, the more accurate the location information disclosed, the higher the risk of location privacy being invaded. Perfect privacy is clearly impossible as long as communication takes place. An important question is how much privacy protection is necessary. Moreover, users often have varying privacy needs in different contexts.

Location perturbation is an effective technique for implementing location k-anonymity. One method is to perturb the location information by reducing its location precision (resolution) in terms of time and space [10, 16]. By reducing the spatial resolution, a spatial region that contains $k-1$ other subjects' location information will be used to replace the spatial position of the subject. By reducing the temporal resolution, the message will be delayed for a certain period of time, which may be long enough to include $k-1$ other subjects' location information. The fundamental challenge is how to control the spatial and temporal resolution reduction to the right amount that will allow LBSs to remain effective and valuable, while enabling mobile users to preserve the desired level of location privacy.

In this paper, we present PRIVACYGRID, a framework for supporting anonymous location based queries in mobile information delivery systems. The goal of the PRIVACYGRID design is to provide a unified and yet effective location anonymization framework for all types of location queries so that mobile users can enjoy LBSs without revealing their exact location information. This paper makes three unique contributions.

- First, we provide a location privacy preference profile model, called location P3P, which allows mobile users to explicitly define their preferred location privacy requirements in terms of both location hiding measures (i.e., location k-anonymity and location l-diversity) and location service quality measures (i.e., maximum spatial resolution and maximum temporal resolution). Our location P3P model supports personalized and continuously changing privacy needs of a diverse user base.
- Second, we develop three fast and effective location cloaking algorithms for providing location k-anonymity and location l-diversity while maintaining the utility of LBSs. The Quad Grid cloaking algorithm is simple and fast but has low success rate for location anonymization. In contrast, the dynamic bottom-up grid cloaking and the dynamic top-down grid cloaking provide high anonymization success rate and yet are efficient in terms of both time complexity and grid index maintenance cost. All three algorithms can dynamically compose the location cloaking regions and select the smallest one that meets both the location anonymity requirements and the location QoS requirements as specified in users' location P3P profiles.
- Third, we describe a hybrid approach that combines the top-down and bottom-up search of the minimal location cloaking regions to further lower the average anonymization time. In addition, we briefly describe the possible increase of the anynimization success rate by a careful combination of temporal cloaking with spatial cloaking.
- We also describe the mechanisms for processing perturbed location range queries.
- Finally, we conduct extensive experimental evaluation of PRIVACYGRID approach, showing that the PRIVACYGRID algorithms can provide optimal location anonymity as defined by per user location P3P without introducing significant performance penalties.

The rest of this paper is organized as follows. We give an overview of the PRIVACYGRID framework in Section 2. We present the three grid-based spatial cloaking algorithms in Section 3 and discuss their efficiency and effectiveness through analysis and examples. We extend spatial cloaking by introducing two possible enhancements in Section 4 and discuss the mechanisms for processing anonymized location queries at the LBS servers in Section 5. We report our experimental evaluation results in Section 6 and discuss the related work in Section 7. Section 8 concludes the paper with a summary and brief discussion of future work.

2 PRIVACYGRID: An Overview

We assume that the LBS system powered by PRIVACYGRID consists of mobile users (clients), wireless network, location anonymization server, and LBS servers. Mobile users communicate with the LBS servers through one or more PRIVACYGRID location anonymization servers. Each mobile user establishes communication with an anonymization server through an authenticated and encrypted connection. Each location anonymization server connects to a number of base stations, tracks the location updates of the mobile users in the range of those base stations, and performs the location anonymization for both location queries and location updates from these mobile users.

In this section, we present an overview of PRIVACYGRID. We first describe the three tier system architecture of PRIVACYGRID and briefly discuss the set of location privacy re-

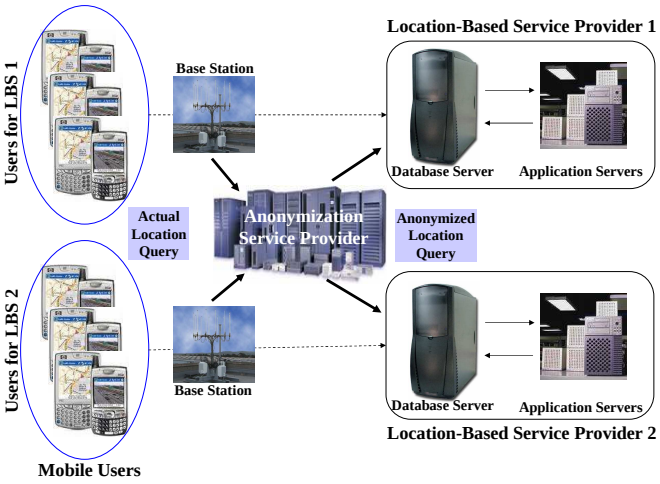


Fig. 1: System Architecture

quirements. Then we define the basic concepts used throughout the paper and outline the location anonymization process.

2.1 System Architecture

The PRIVACYGRID system promotes the three-tier architecture for supporting anonymous information delivery in a mobile environment, as shown in Figure 1. The top tier is the modeling of users' personal location privacy requirements. The middle tier is the location perturbation service typically provided by a trusted third party location server, specialized in location tracking and anonymization service. The third tier is the processing of cloaked location queries at the individual LBS providers. A number of research and development projects have used the trusted third party location anonymizer infrastructure [16, 14, 21] for protecting location privacy of mobile users.

We devise our location privacy preference profile model to allow mobile users to specify what, when, how (and with whom) their location information could be shared. In addition to the standard P3P specification [4], we add four location privacy specific measures and refer to them as location P3P. The first measure is the *location k-anonymity*, which allows the mobile user to control her state of being not identifiable from a set of $k-1$ other users. The second measure is the *location l-diversity*, which allows the mobile user to control her state of being not identifiable from a set of l actual (physical) locations (such as buildings or postal addresses). This measure can be seen as a companion measure of the location k anonymity, and is particularly useful in reducing the risks of unwanted location inference when there are k or more distinct users at a single physical location (such as a clinic office or a political event gathering). The third measure is the *maximum spatial resolution*, which allows the mobile user to control the spatial resolution reduction within an acceptable level of QoS. It can be changed or adjusted according to the type of location services and the time of day, month, or year when the LBS are being offered. Similarly, the fourth measure is the *maximum temporal resolution*, which controls the temporal resolution reduction within the acceptable duration of time to keep the perceived QoS of the mobile user within an acceptable delay based on the type of location services and the time when the LBS is being requested.

The middle tier is the location perturbation service typically offered by a third party location anonymization server. The location anonymization server anonymizes the location information from mobile users before it can be passed to the actual LBS providers. In the first prototype of PRIVACYGRID, we use the spatial and temporal location cloaking techniques to perform location perturbation. The location information of a mobile user (such as her position update or the position where she poses a location query) will be mapped to a location cloaking box based on the location P3P of the user. For those mobile users that do not want to be tracked by others, no perturbation will be performed on their location updates. For those LBSs that offer location dependent information over public data, such as restaurants, gas stations, offices, and so forth, no location updates of mobile users will be passed from the location anonymization servers to the LBS servers. Mobile users who wish to allow their movements to be tracked by certain LBSs or some group mobile users may use their location P3P to specify how they want their location updates to be cloaked and to which LBS servers their location updates can be provided. Similarly, for location queries, there are a couple of alternative ways for the location anonymizer service to pass the location cloaking box to the corresponding LBS provider. For example, one can choose to have the location anonymizer as the middleman between mobile users and individual LBS providers such that location queries are posted to the location anonymizer and passed to the LBS provider and the result is returned to the mobile user through the location anonymizer. Alternatively, before contacting the LBS provider directly, a mobile user can have her location information *filtered* by reducing its precision/resolution in terms of time and space according to her location P3P, ensuring that the location queries sent to the LBS meet her desired location k-anonymity and location l-diversity requirements. In the subsequent sections we present the PRIVACYGRID algorithms for efficient and effective location cloaking in Section 3.

It is important to note that location perturbation may result in the fact that the LBS provider sends more than requested results back to the mobile user. Thus the mobile node needs to perform further filtering before presenting the results to the mobile user, leading to additional communication and processing overhead on mobile nodes. Thus, the third tier of PRIVACYGRID is dedicated to the methods for efficient processing of perturbed location queries at the individual LBS server. In contrast to the existing literature on location query processing that concentrates on spatial positions (points), we need to extend some existing spatial query processing methods to spatial region based techniques. For example, [21] described an approach to process location cloaked kNN queries.

2.2 Location Privacy Requirements

In PRIVACYGRID the following requirements are considered essential for supporting anonymous location queries.

1. **Personalized User Privacy Levels:** We argue that location privacy consists of two measures: location k-anonymity and location l-diversity. The former allows a mobile user to control a state of being not identifiable from a set of $k-1$ other users. The latter allows a mobile user to control a state of being not identifiable from a set of l actual (physical) locations (such as buildings or postal addresses). These two measures are complimentary and particularly useful in reducing the risks of unwanted location inference when there are more than $k-1$ distinct users at a single physical location (such as a clinic office or a church). The system must have the capability to allow a mobile user to specify the desired k value for location *k-anonymity* and the desired l value for location *l-diversity* for each of her location updates or location queries. The user may change her privacy preference levels as often as required or even on a per message basis.
2. **QoS Guarantees:** The PRIVACYGRID framework provides a mobile user with the capability of specifying two QoS metrics: (1) the maximum spatial resolution, indicating that the amount of spatial inaccuracy she can tolerate to maintain meaningful and acceptable service quality; and (2) the maximum temporal resolution, ensuring that the delay introduced for location cloaking is acceptable from QoS standpoint. By utilizing these two quality metrics, PRIVACYGRID aims at devising location cloaking algorithms that find the smallest possible cloaking region for each location cloaking request of a mobile user, which satisfies her privacy requirements defined by location k-anonymity and location l-diversity.
3. **Dynamic Tradeoff between privacy and quality:** PRIVACYGRID location perturbation algorithms should be capable of dynamically making tradeoffs between location privacy and location QoS. Unnecessarily large cloaking boxes will lead to poor QoS in terms of larger result set to transport and filter at the mobile client side, inevitably leading to higher delays for obtaining useful query results.
4. **Efficiency and Scalability:** In PRIVACYGRID a mobile user can change her location P3P at any time. The cloaking algorithms should be effective and scalable in the presence of changing requirements on both the number of mobile users and the content of location P3P. At the same time, the cloaking algorithms must be fast, keeping the perceived delays due to location anomymization as low as possible.
5. **Unified Framework:** A single unified framework should be devised to meet personalized and customizable location anonymization demands and support a variety of anonymous LBSs with respectable performance, privacy guarantees and quality assurance.

2.3 Basic Concepts

In this section we only defines the basic concepts that are required for the subsequent discussion of the PRIVACYGRID framework.

Universe of Discourse (UoD): We refer to the geographical area of interest as the universe of discourse (or map), which is defined by $U = Rect(x, y, w, h)$, where x is the x-coordinate and y is the y-coordinate of the lower left corner of a rectangular region, w is the width and h is the height of the universe of discourse. Basically, we consider maps which are rectangular in shape.

Grid and Grid cells: In our framework, we map the universe of discourse $U = Rect(x, y, w, h)$ onto a grid G of cells. Each grid cell is an $\alpha \times \beta$ rectangular area, where α, β are system parameters that defines the cell size of the grid G. Formally, a grid corresponding to the universe of discourse U can be defined as $G(U, \alpha, \beta) = \{A_{i,j} : 1 \leq i \leq M, 1 \leq j \leq N, A_{i,j} = Rect(x + i \times \alpha, y + j \times \beta, \alpha, \beta), M = \lceil w/\alpha \rceil, N = \lceil h/\beta \rceil\}$. $A_{i,j}$ is an $\alpha \times \beta$ rectangular area representing the grid cell that is located in the ith column and jth row of the grid G.

Position to Grid Cell Mapping: Let $\vec{p} = (p_x, p_y)$ be the position of a moving object in the universe of discourse $U = Rect(x, y, w, h)$. Let $A_{i,j}$ denote a cell in the grid $G(U, \alpha, \beta)$. $Pmap(\vec{p})$ is a position to grid cell mapping, defined as $Pmap(\vec{p}) = A_{\lceil \frac{p_x - x}{\alpha} \rceil, \lceil \frac{p_y - y}{\beta} \rceil}$.

Current Grid Cell of a Moving Object: Current grid cell of a moving object is the grid cell which contains the current position of the moving object. If o_m is a moving object whose current position, denoted as $\vec{p}$, is in the Universe of Discourse U, then the current grid cell of the object is formally defined by $curr\_cell(o_m) = Pmap(\vec{p})$.

User Privacy Preference Profile: In PRIVACYGRID a personalized location privacy model is used. A user registered with the anonymization server specifies her location privacy requirements in terms of her desired user anonymity level k, desired location diversity level l, maximum spatial resolution $\{d_x, d_y\}$, and maximum temporal resolution d_t. Each location P3P record is of the form $\langle object_{id}, LBS_{info}, request_{id}, k, l, \{d_x, d_y, d_t\}\rangle$, where $object_{id}$ identifies the user, LBS_{info} is optional and provides the type and the identifier of the LBS this P3P record is applied to, and $request_{id}$ is optional and is used to uniquely identify a service request posed by the user with the given $object_{id}$. We use $k = 1$ and $l = 1$ as the default setting (neither anonymity nor diversity is required). When $k = 1$ and $l = 1$, d_x, d_y, d_t are set to `nil`.

2.4 Location Anonymization Server

In PRIVACYGRID, each incoming location service request m_s received by the location anonymization server is of the form $\langle object_{id}, request_{id}, \{x, y, t\}, F, k, l, \{d_x, d_y, d_t\}\rangle$. The $object_{id}$ and $request_{id}$ uniquely identify a message. The coordinate (x, y) and the timestamp t together form the three dimensional spatio-temporal location point of the mobile user who issued the message m_s. F denotes the content filter of the request, such as gas stations, french restaurants, or yellow taxi cabs. The parameters $\{k, l, d_x, d_y, d_t\}$ denote the location P3P specified by the mobile user who issued this

request. The location anonymization server will transform the original message m_s to a location perturbed message m_t of the form $\langle h(object_{id}||request_{id}), \{X : [x_s, x_e], Y : [y_s, y_e], I : [t_s, t_e]\}, F\}\rangle$, where h is a secure hash function, $X : [x_s, x_e]$ and $Y : [y_s, y_e]$ denote the spatial cloaking box of the message on x-axis and y-axis respectively, such that $x_e - x, x - x_s \leq d_x$ and $y_e - y, y - y_s \leq d_y$; and $I : [t_s, t_e]$ denotes the temporal cloaking interval such that $t_e - t_s \leq d_t$. Furthermore, there are more than $k - 1$ other mobile users and more than l symbolic addresses located within the same spatio-temporal cloaking box defined by $\langle X : [x_s, x_e], Y : [y_s, y_e], I : [t_s, t_e]\rangle$. We call this process message perturbation through spatio-temporal cloaking. We will describe the three grid-based spatial cloaking algorithms for finding the minimal spatial cloaking box – $\langle X : [x_s, x_e], Y : [y_s, y_e]\rangle$ and the minimal temporal cloaking period $I : [t_s, t_e]$ that meet the k-anonymity and l-diversity requirement in the subsequent sections.

3 PRIVACYGRID Spatial Cloaking Algorithms

In this section we first describe the basic *Quad Grid* algorithm for finding the minimal spatial cloaking box for the given location of a mobile user. By minimal, we mean that there exist no smaller spatial cloaking regions that satisfy both location k-anonymity and location l-diversity as well as maximum spatio-temporal resolution constraints defined in the users' location P3P. We then present two dynamic grid-based cloaking algorithms: bottom up spatial cloaking and top-down spatial cloaking. Both provide much higher anonymization success rate than the basic Quad Grid cloaking algorithm and reduced grid maintenance cost while keeping the desired performance.

We first give an overview of the basic data structures used in PRIVACYGRID. Then we introduce the Quad Grid cloaking approach and illustrate the algorithm by examples. Bottom-up and Top-down spatial cloaking are introduced as two dynamic grid cloaking algorithms that improve the cloaking effectiveness of the Quad Grid approach.

3.1 Data Structures

In PRIVACYGRID, the entire map is divided into a grid of cells of size $\alpha \times \beta$. α and β are system-defined parameters. Each mobile user is responsible for reporting its location to the anonymization server either periodically or when it moves outside its current grid cell [13]. Upon receiving a location update, the location anonymization server maintains the following data structure: the mapping of a mobile user's position to its current grid cell, the CellObjectCountMap (to be defined below), and the hierarchical grid index. When a mobile user moves out of its current cell C_i and entered a new cell C_j, the grid index needs to be updated for both cells on their CellObjectCountMap. Figure 2 illustrates the hierarchical grid index and the Cell Object Count Map by an example.

Cell Object Count Map: In addition to the grid cell to object mapping maintained by the grid index, we also keep a count of the number of mobile objects and the number of still objects

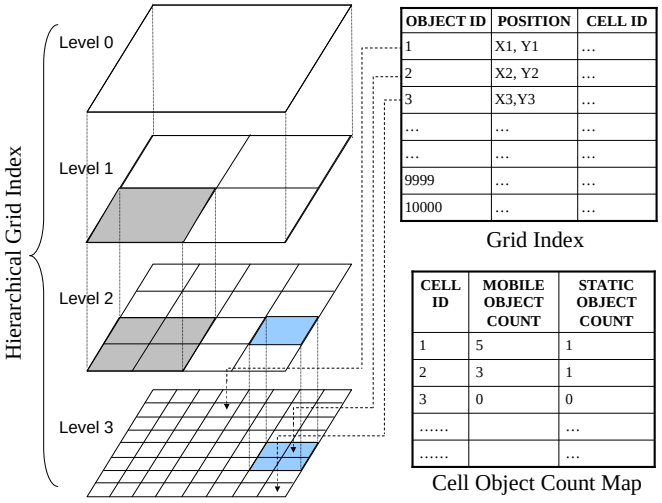


Fig. 2: Grid Index Data Structures for PRIVACYGRID

(so-called symbolic addresses, such as gas stations, restaurants, offices, and so forth) located in each grid cell. This allows for quick computation of the total number of mobile users and the total number of still objects located in a given spatial area using the grid cells and the grid index. For each grid cell, the count of still objects remains unchanged most of the time. However, the count of mobile objects may change as mobile users move from one grid cell to another. The mobile users' movement across its current grid cell requires the mobile object count for the old cell to be reduced by one and the corresponding count for the new cell to be increased by one.

Hierarchical Grid Index: The Hierarchical Grid Index (HGI) is a *multi-level* [24] data structure which allows for fast and efficient computation of object counts belonging to a particular region of the map. The construction of a HGI is shown in Figure 2 and is performed by subsequent splitting of grid cells into four smaller equal sized cells at the next lower level of the index. The number of cells at the level l (≤ 0) of the index is 4^l, where *l* indicates the level of the index. At level zero ($l = 0$) the index comprises of a single cell representing the entire map. This cell is split into four equal sized cells to form level one of the index. We call the cell at level i the *parent cell* of the four *children cells* at level j where $j = i + 1$. Subsequently the cells at level j may further be split into four cells each to form the level $j + 1$ of the index. Figure 2 displays an HGI structure of height three ($l = 2$) showing the parent-child cell relationships for each level of the index. The HGI maintains the object to cell mapping only for the lowest level of the index. However, the cell object count map is also maintained for the higher levels of the index in order to aid fast calculation of cloaking areas (see Section 3.2 for detail). Mobile object movement may lead to changes in the mobile object count for cells at the lowest level and for the subsequent parent cells too.

3.2 The Quad Grid Cloaking Algorithm

The Quad Grid Cloaking algorithm presents a basic and straightforward way of utilizing the HGI data structure to perform spatial cloaking. The algorithm takes as the input

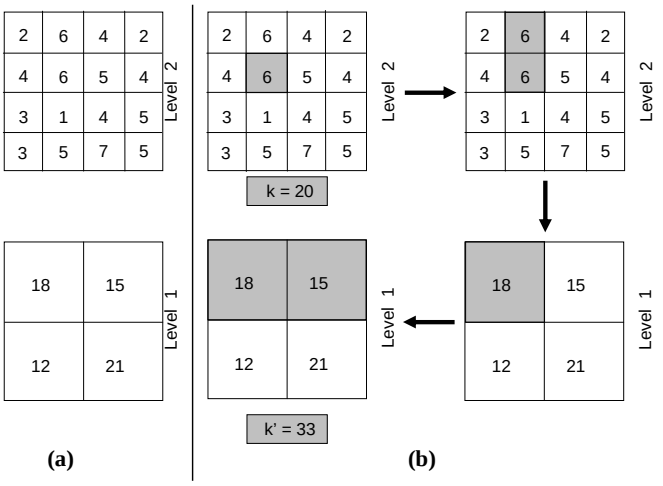


Fig. 3: Quad Grid Cloaking Example

the original message m_s from a mobile user and produces the perturbed message m_t by replacing the two dimensional spatial location point (x, y) with the minimal spatial cloaking box $\langle X : [x_s, x_e], Y : [y_s, y_e]\rangle$, which satisfies the mobile users' location privacy requirement $\{k, l\}$, and the location service quality requirement $\{d_x, d_y, d_t\}$. Algorithm 1 presents a sketch of the algorithmic detail. It first invokes the *GridIndexSearch* function to obtain the current cell identifier (cid) of the mobile user using her $object_{id}$ and her current spatial location point (x, y). Then the algorithm performs the spatial cloaking recursively and each iteration proceeds in three steps. It first locates the number of moving objects (MN) and the number of still objects (SN) in the current cell cid. Then it compares $\{MN, SN\}$ with the location privacy requirement $\{k, l\}$ of the mobile user ($object_{id}$) and computes the minimal spatial cloaking box. If the current cell does not meet the anonymity requirements, then the parent cell of cid will be used to start the next iteration.

Concretely, the algorithm first uses the current cell identifier for the mobile user to obtain the number of moving objects (MN) and the number of still objects (SN) within this particular cell by searching the *Cell Object Count Map* data structure. Then it performs k-anonymity and l-diversity check on this grid cell. If $MN \geq k$ and $SN \geq l$, then this single cell can potentially form the spatial cloaking box for this request and may be returned as the answer after verifying that it does not violate the maximum spatial resolution constraints (lines 2–5). Otherwise, the algorithm attempts to extend the search for cloaking box in vertical or horizontal direction of the current cell. We define the *vertical neighbor (cid_v)* of cell cid as the cell located above or below cid with the same parent cell in the HGI. The *horizontal neighbor (cid_h)* is identified as the cell located on either side of cid with the same parent cell in the HGI. The algorithm will then calculate the object counts MN and SN of cid and cid_v as well as cid and cid_h as shown in line 8 of the algorithm. If only one of these two cell combinations satisfies the k-anonymity and l-diversity requirement (lines 9–17), the algorithm attempts to choose that combination to continue the verification of whether it meets

Algorithm 1 Quad Grid Cloaking

Input: $\{object_{id}, request_{id}, x, y, t\}, \{d_x, d_y, d_t\}, \{k, l\}$
Output: $MinimalSpatialCloakingBox$

1: $cid \longleftarrow GridIndexSearch(object_{id}, x, y)$
2: FUNCTION QUAD_GRID_CLOAKING($k, l, \{x, y\}$,
3: $\{d_x, d_y\}, cid$)
4: $(MN, SN) \longleftarrow$ CellObjectCountMapSearch(cid)
5: **if** $(cid.MN \geq k)$ && $(cid.SN \geq l)$ **then**
6: $CheckCloakingBoxValidity(x, y, d_x, d_y)$
7: return cid;
8: **end if**
9: $cid_v \longleftarrow$ Vertical neighbor cell of cid.
10: $cid_h \longleftarrow$ Horizontal neighbor cell of cid.
11: $MN_v = cid.MN + cid_v.MN; MN_h = cid.MN + cid_h.MN$;
12: $SN_v = cid.SN + cid_v.SN; SN_h = cid.SN + cid_h.SN$;
13: **if** $(((MN_v \geq k)$ && $(SN_v \geq l))$ $\|$ $((MN_h \geq k)$ && $(SN_h \geq l)))$ **then**
14: **if** $((MN_v \geq k$ && $MN_h \geq k$ && $MN_h > MN_v)$ $\|$ $MN_v < k)$ **then**
15: $CheckCloakingBoxValidity(x, y, d_x, d_y)$
16: return cid, cid_h;
17: **else**
18: **if** $(MN_h == MN_v)$ **then**
19: **if** $(SN_h \geq SN_v)$ **then**
20: $CheckCloakingBoxValidity(x, y, d_x, d_y)$
21: return cid, cid_h;
22: **else**
23: $CheckCloakingBoxValidity(x, y, d_x, d_y)$
24: return cid, cid_v;
25: **end if**
26: **end if**
27: **else**
28: $CheckCloakingBoxValidity(x, y, d_x, d_y)$
29: return cid, cid_v;
30: **end if**
31: **else**
32: QUAD_GRID_CLOAKING($k, l, \{x, y\}, \{d_x, d_y\}$, PARENT($cid$));
33: **end if**

the maximum spatial resolution constraint. If both cell combinations satisfy the k-anonymity and l-diversity requirement, the algorithm picks the combination which provides a higher k anonymity level (or higher l-diversity level when both combinations have the same k value). Upon passing the privacy check, the algorithm will validate whether the selected cell combination meets the maximum spatial resolution constraint of this request, and if so, it is returned as the minimal spatial cloaking box (line 11 and line 14). However, if this selected cloaking box is does not meet the maximum spatial resolution requirement (i.e., bigger than the range defined by the maximum spatial resolution), the algorithm has to drop this message (unless temporal cloaking is turned on). In case that neither of the two combinations satisfy the k-anonymity and l-divrsity requirements, the algorithm starts the next iteration

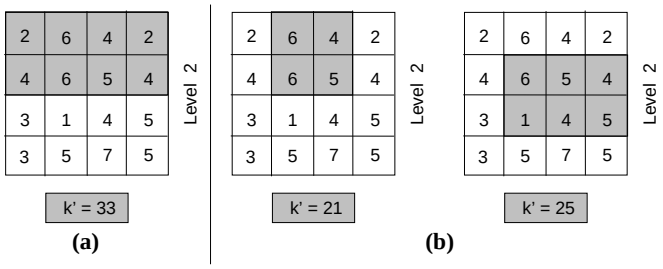


Fig. 4: Quad Grid Cloaking Weakness

with the parent cell of the current cid.

We illustrate the working of Algorithm 1 by example. Figure 3(a) displays a HGI structure of height two. For simplicity we only display the mobile object count for each cell at a particular time instant within each cell since the still object count is relatively stable. We observe that the mobile object count for each cell at level one is the sum of the object counts for its children cells at level two. Figure 3(b) illustrates the working of the Quad Grid Cloaking algorithm for a given location anonymization request issued by a mobile object within the shaded cell (the cell with object count of 6). Suppose that this anonymization request has the k-anonymity level set to $k = 20$. Neither $MN_v = 12$ nor $MN_h = 10$ satisfy this k-anonymity requirement of 20, so the algorithm selects the parent cell at level one of HGI. However, this parent cell has the mobile object count of 18, thus it is still insufficient to meet the desired k-anonymity level of 20. The algorithm needs to further expand the candidate cloaking box in either vertical or horizontal direction. If the expansion proceeds in vertical direction, the candidate cloaking box provides k-anonymity level of k'= 30, otherwise we obtain k'=33 by expanding the box in the horizontal direction. Given that the cloaking area will be the same irrespective of whether the expansion is along the vertical or horizontal direction, the algorithm selects the candidate cell combination that provides a higher anonymity level. In this example, the horizontal expansion is chosen as the final cloaking box as displayed in the shaded area at the bottom left part of Figure 3(b).

3.3 Problems with Quad Grid Cloaking

The Quad Grid cloaking algorithm is extremely fast as it uses the HGI data structure that maintains the object counts at different levels of the Grid index. However, the algorithm is restricted by the static nature of the Quad Grid data structure when performing the cell-based expansion for finding the minimal spatial cloaking box that meets both privacy and quality constraints of the mobile user. We illustrate the performance penalty of this problem by example in this section and provide experimental evaluation to validate our analysis in section 6.

Again for simplicity we only deal with the mobile object counts in this example as the still object counts are insensitive to the movement of mobile users. Figure 4(a) displays the cloaking area constructed by the Quad Grid algorithm (at the lowest level of HGI) for the example given in Figure 3. We observe that the minimal cloaking area chosen is unnecessarily larger than required even though the achieved anonymity level (k'=33) is well above the required anonymity level of k=20. Figure 4(b) displays a couple of scenarios where the cloaking area can be constructed using fewer number of base level cells while still meeting the required anonymity level. There are a number of weaknesses that prevent the Quad Grid approach from finding the smallest possible cloaking area within the user specified privacy and quality requirements.

1. **Rapid and constrained area expansion:** At each iteration, the Quad Grid algorithm expands the cloaking area to twice its current size by selecting a horizontal or vertical neighboring cell. In case that the iteration involves moving to a higher level of the HGI (line 18 in algorithm 1), the area expands to four times of its size at the beginning of the iteration. At the higher levels of a HGI, this leads to a rapid expansion in the candidate cloaking area, restricting the ability of the algorithm to find the minimal cloaking box that meets the location P3P requirements.
2. **Unnecessarily High k-Anonymity:** From the above example we observe that the Quad Grid cloaking algorithm achieves much higher anonymity levels than the desired levels. Unnecessarily large anonymity levels have an associated cost of a larger cloaking area which hurts the QoS provided to the user.
3. **Anonymization Success Rate:** An important goal of the location cloaking algorithm is to anonymize messages at a higher success rate while meeting the user specified privacy preference profile. The Quad Grid algorithm, due to rapid expansion of the cloaked areas, often overshoots the maximum spatial resolution, thus resulting in higher percentage of messages being dropped due to its inability to find a satisfactory perturbation (see Section 6 for experimental results). This severely hurts the performance of the algorithm.
4. **Pre-defined Cloaking Path:** The Quad Grid algorithm utilizes a fixed hierarchy of the HGI data structure to perform cell expansion in searching for minimal spatial cloaking box, thus limiting its ability to explore all options for cell-based expansion. As a result, the algorithm can only select the cloaking areas through a pre-defined quad grid cell composition structure along the hierarchy of HGI.

To overcome the problems with Quad Grid cloaking, we need to relax the rigid hierarchical quad grid cell expansion process implied by the construction structure of HGI. This motivates us to look into the dynamic cell expansion approach. In the rest of the paper we focus on the bottom-up and the top-down grid cloaking algorithms. Unlike the Quad Grid cloaking approach, the dynamic grid cloaking approach is able to produce close to optimal cloaking areas. The algorithm accepts the same input arguments as the Quad Grid approach (recall Section 3.2).

3.4 Dynamic Bottom-Up Grid Cloaking

The *Bottom-Up* approach to dynamic cloaking starts with the base cell containing the object from which the cloaking request has originated. A sketch of the algorithm is given in Algorithm 2. The algorithm first determines if the current cell (cid) has sufficient mobile object count and still object count to satisfy the privacy requirements and verifies the validity of the cloaking box in terms of the user specified maximum spatial resolution levels (lines 2–6).

Algorithm 2 Bottom-Up Dynamic Grid Cloaking

Input: $\{object_{id}, request_{id}, x, y, t\}$, $\{d_x, d_y, d_t\}$, $\{k, l\}$
Output: $MinimalSpatialCloakingBox$

1: $cid \longleftarrow GridIndexSearch(object_{id}, x, y)$
2: FUNCTION BOTTOM_UP_GRID_CLOAKING(k, l,
3: $(x, y), (d_x, d_y), cid$)
4: **if** (cid.MN $\geq k$) && (cid.SN $\geq l$) **then**
5: $CheckCloakingBoxValidity(x, y, d_x, d_y)$
6: return cid;
7: **end if**
8: **while** (selectedCells.MN $< k \parallel$ selectedCells.SN $< l$) **do**
9: $Row_N \longleftarrow$ Row above uppermost selected row.
10: $Row_S \longleftarrow$ Row below lowermost selected row.
11: $Col_E \longleftarrow$ Right column of rightmost selected column.
12: $Col_W \longleftarrow$ Left column of leftmost selected column.
13: $CheckRowSpatialValidity(x, d_x, Row_N)$;
14: $CheckRowSpatialValidity(x, d_x, Row_S)$;
15: $CheckColSpatialValidity(y, d_y, Col_E)$;
16: $CheckColSpatialValidity(y, d_y, Col_W)$;
17: $MN_N = selectedCells.MN + Row_N.MN$;
18: $SN_N = selectedCells.SN + Row_N.SN$;
19: $MN_S = selectedCells.MN + Row_S.MN$;
20: $SN_S = selectedCells.SN + Row_S.SN$;
21: $MN_E = selectedCells.MN + Col_E.MN$;
22: $SN_E = selectedCells.SN + Col_E.SN$;
23: $MN_W = selectedCells.MN + Col_W.MN$;
24: $SN_W = selectedCells.SN + Col_W.SN$;
25: **odd iteration:**
26: selectRowOrColumnToAdd($MN_N, MN_S, MN_E,$
27: $MN_W, SN_N, SN_S, SN_E, SN_W$);
28: **even iteration:**
29: **if** (addedRowInPreviousIteration) **then**
30: selectColumntoAdd(MN_E, MN_W, SN_E, SN_W);
31: **else**
32: selectRowtoAdd(MN_N, MN_S, SN_N, SN_S);
33: **endif**
34: **end while**
35: MinimalCloakingBox $\longleftarrow$ CloakingArea(selectedRows, selectedColumns)
36: return MinimalCloakingBox;

In case that the current cell does not meet the user's privacy requirements, the algorithm expands the current cell (i.e., the candidate cloaking box) to any of the four neighboring cells. This is in contrast to the Guad Grid approach that restrict the expansion to only those neighboring cells with the same parent in HGI. The decision on which of the four cells to choose first is based on the highest object count in the candidate cells. The cells composing the cloaking box are identified by their rows and columns in the grid index. The *selected rows* and *selected columns* are maintained by the algorithm (in an incremental order) and can be used to infer the selected cells for forming the final cloaking area. The current candidate cloaking box may be expanded further in any direction (*North, South, East or West*) by adding the row above the uppermost selected row (or below the lowermost selected row) or the column to the right of the rightmost selected column (or to the left of the leftmost selected column), thus dynamically building the cell-based cloaking box by adding suitable rows or columns. The rows denoted by Row_N, Row_S or columns denoted by Col_E, Col_W (lines 9–12) are used to calculate the cell count after addition (lines 17–24). The validity of the rows or columns to meet the maximum spatial resolution requirements is checked before proceeding with the addition (lines 11–14). The algorithm selects the row or column which leads to the maximum object count after addition. For every odd iteration, the algorithm determines whether to add a row or column as the cloaking area may be expanded in any of the four directions (lines 25–27). For even iterations, the algorithm expands the cloaking area, depending on whether a row or column was added in the previous iteration, in order to ensure that no skew is introduced in any direction (lines 28–33). For example, if the algorithm added a row during the previous iteration, the current iteration would involve addition of either the column Col_E or Col_W. The steps (lines 8–34) are recursively repeated as long as the total object count of *all* cells in the selected rows and columns is less than the required *k-anonymity* and l-diversity requirements. Upon meeting the privacy and quality requirements, the algorithm uses the selected rows and columns to determine the selected cells and composes the minimal cloaking area in terms of the selected cells. It returns the final minimal spatial cloaking area and terminates.

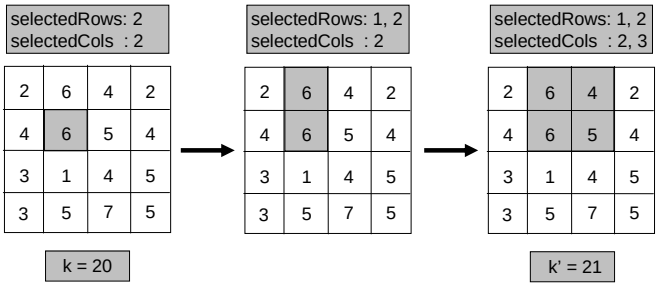


Fig. 5: Bottom-Up Dynamic Hierarchical Grid Cloaking Example

The working of the *Bottom-Up* dynamic approach is explained through an example in Figure 5. For simplicity we only use the mobile object count in this example. The cloaking request originates from the shaded cell with an object count of six. As this is insufficient to meet the k-anonymity requirement, the algorithm starts expanding the selected cell. Note that the algorithm works with a flat grid index (or the lowest level of the HGI data structure). Thus no additional in-

formation related to higher levels of the HGI hierarchy needs to be maintained. The current cell is located at the second row and the second column in the grid, which are marked as *selectedRows* and *selectedCols* by the algorithm respectively. All neighboring cells of the shaded cell are considered and the first row to the north which increments the object count to 12 is chosen as the first cell to expand and added into the *selectedRows*. As the total object count of 12 in this candidate cloaking box does not meet the k-anonymity requirement of $k = 20$, the algorithm starts the next iteration. In this iteration, we first consider the column to the left (Col_W), which is not sufficient to meet the privacy requirements. Then we consider the addition of the right column (third column in the grid) which provides a cloaking area with the object count of k'=21, which is sufficient to meet the anonymity requirement. Thus the algorithm terminates and returns $selectedRows = \{1, 2\}$ and $selectedCols = \{2, 3\}$. We can see the area provided by the dynamic bottom-up grid cloaking approach is much smaller than the one provided by the Quad Grid approach (in Figure 3), even though both meet the privacy requirements.

3.5 Dynamic Top-Down Grid Cloaking

Dynamic cloaking may also proceed by starting with the largest possible cloaking area as permitted by the maximum spatial resolution. We call this approach the *Top-Down* dynamic gird cloaking and Algorithm 3 gives the algorithmic sketch. First, the top-down algorithm calculates the cells needed to compose the largest cell-based candidate cloaking box, which meets the maximum spatial tolerance requirement (line 4). The cloaking area is expressed as a set of *selectedRows* and *selectedCols*, as in the bottom-up approach. If the largest possible candidate cloaking box fails to meet the required privacy requirements, the message cannot be cloaked using the user-defined privacy and quality metrics and the algorithm terminates (lines 5–7). The algorithm proceeds beyond this step only if it is possible to cloak the message. Otherwise, the top-down approach repeatedly removes appropriate rows or columns from the *maximum cloaking area* generated in line 4. Each odd iteration selects the outermost rows or columns (lines 9–12) with minimum object counts, so that the selected cloaking area (after removing a row or column) has the maximum possible object count (lines 13–33). If any of the calculated values are higher than the k-anonymity requirement, rows or columns may be removed appropriately, provided that the row or column containing the object which initiated the cloaking request is not removed (line 21–24). Even iterations may remove rows or columns dependent on the steps performed by the previous iteration (lines 25–30). The algorithm terminates if none of the object counts are higher than the user specified *k* value and *l* value (lines 31–33). It returns the final cloaked spatial area defined by the *selectedRows* and *selectedCols*. The top-down approach speeds up the cloaking in certain scenarios when compared to the bottom-up approach.

The example in Figure 6 illustrates the Top-Down approach with the same starting conditions as in the previous examples. The shaded area in the leftmost figure displays the initial maximum possible cloaking area. The end result with the top-down approach is similar to the result obtained using the bottom-up approach in this example.

Algorithm 3 Top-Down Dynamic Grid Cloaking

Input: $\{object_{id}, request_{id}, x, y, t\}$, $\{d_x, d_y, d_t\}$, $\{k, l\}$
Output: $MinimalSpatialCloakingBox$

1: $cid \longleftarrow GridIndexSearch(object_{id}, x, y)$
2: FUNCTION TOP_DOWN_GRID_CLOAKING($k, l,$
3: $(x, y), (d_x, d_y), cid$)
4: selectedCells = $MaxCloakingArea\{x, y, d_x, d_y\}$;
5: **if** (selectedCells.MN $<$ k) $\|$ (selectedCells.SN $<$ l) **then**
6: break;
7: **end if**
8: **while** (selectedCells.MN $>$ k $\&\&$ selectedCells.SN $>$ l) **do**
9: $Row_N \longleftarrow$ Uppermost selected row.
10: $Row_S \longleftarrow$ Lowermost selected row.
11: $Col_E \longleftarrow$ Rightmost selected column.
12: $Col_W \longleftarrow$ Leftmost selected column.
13: $MN_N = selectedCells.MN - Row_N.MN$;
14: $SN_N = selectedCells.SN - Row_N.SN$;
15: $MN_S = selectedCells.MN - Row_S.MN$;
16: $SN_S = selectedCells.SN - Row_S.SN$;
17: $MN_E = selectedCells.MN - Col_E.MN$;
18: $SN_E = selectedCells.SN - Col_E.SN$;
19: $MN_W = selectedCells.MN - Col_W.MN$;
20: $SN_W = selectedCells.SN - Col_W.SN$;
21: **if** $((MN_N \geq k \;\&\&\; SN_N \geq l) \,\|\, (MN_S \geq k \;\&\&\; SN_S \geq l) \,\|\, (MN_E \geq k \;\&\&\; SN_E \geq l) \,\|\, (MN_W \geq k \;\&\&\; SN_W \geq l))$ **then**
22: **odd iteration:**
23: selectRowOrColumnToRemove($MN_N, MN_S,$
24: $MN_E, MN_W, SN_N, SN_S, SN_E, SN_W$);
25: **even iteration:**
26: **if** (removedRowInPreviousIteration) **then**
27: selectColtoRemove(MN_E, MN_W, SN_E, SN_W);
28: **else**
29: selectRowtoRemove(MN_N, MN_S, SN_N, SN_S);
30: **endif**
31: **else**
32: break;
33: **end if**
34: **end while**
35: MinimalCloaking Box $\longleftarrow$ CloakingArea(selectedRows, selectedColumns)
36: return MinimalCloaking Box;

4 Possible Enhancements

In this section we discuss two enhancements for the PRIVACYGRID spatial cloaking algorithm: the *hybrid cloaking* approach and the incorporation of *temporal tolerance* into the spatial cloaking algorithms.

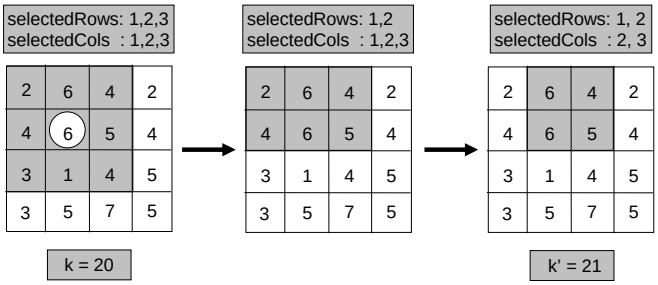


Fig. 6: Top-Down Dynamic Hierarchical Grid Cloaking Example

4.1 Hybrid Cloaking

The hybrid cloaking algorithm combines the bottom-up and top-down approaches to improve the performance of finding minimal cloaking region for a location service request. There are several ways that one can combine the top-down and bottom up approach. For example, making a choice upon receiving a location anonymization request. In this case, the main challenge is how to appropriately decide whether to proceed in a bottom-up or a top-down manner upon receiving a message cloaking request. When a request has lower k-anonymity level and higher maximum spatial resolution value, the hybrid algorithm proceeds in a bottom-up manner. However, for the request with a higher k-anonymity value and a low maximum spatial resolution value, the top-down approach is chosen as it works faster in finding the minimal cloaking box. We provide a brief analysis on the factors for making such decision when we assume a relatively stable state of cells being considered and a uniform object distribution.

Consider the map with a grid comprising of cells of size $\alpha \times \beta$ superimposed on top of it. We assume that each cell has n objects on average as this analysis assumes a uniform object distribution. Our dynamic approaches advocate addition (or removal) of rows and columns alternately. We assume that the final cloaking area is constructed by adding an equal number of rows and columns. Given an anonymization request with anonymity level k, we conclude that,

$$r = \sqrt{\frac{k}{n}} \tag{1}$$

where r^2 is the average number of cells estimated to meet our anonymity requirements, consequently we need to add *r* rows and an equal number of columns to form the required cloaking area. We assume that the maximum cloaking area as defined by the maximum spatial resoluton values consists of a rows and b columns which can be approximately quantified as below.

$$a = 2 \times \lfloor \frac{d_y}{\beta} \rfloor + 1 \tag{2}$$

$$b = 2 \times \lfloor \frac{d_x}{\alpha} \rfloor + 1 \tag{3}$$

The bottom-up approach starts with a singe cell and may expand to include $a \times b$ cells, requiring addition of $a - 1$ rows and $b - 1$ columns. However, on an average it is expected to add only $r-1$ rows and $r-1$ columns where r is as defined in equation 1 above. The top down approach starts with a rows and b columns and it is expected to remove $a - (r - 1)$ rows and $b - (r - 1)$ columns on an average. Hence the expected number of iterations performed by the bottom-up approach is,

$$i_{bu} = 2 \times \{\sqrt{\frac{k}{n}} - 1\} \tag{4}$$

Similarly for the top down approach the expected number of iterations is,

$$i_{td} = a - \{\sqrt{\frac{k}{n}} - 1\} + b - \{\sqrt{\frac{k}{n}} - 1\} \tag{5}$$

The iterations performed by the top-down approach, on average, require more computation when compared with the iterations performed by the bottom-up approach as the top-down iterations are acting on a much larger number of rows and columns. Let the average cost of iterations performed by the top-down approach be γ times the average cost of iterations performed by the bottom-up approach. From the above equations we can determine the cost of proceeding in a bottom-up manner as,

$$T_{bu} = 2 \times \{\sqrt{\frac{k}{n}} - 1\} \times Cost_{bu} \tag{6}$$

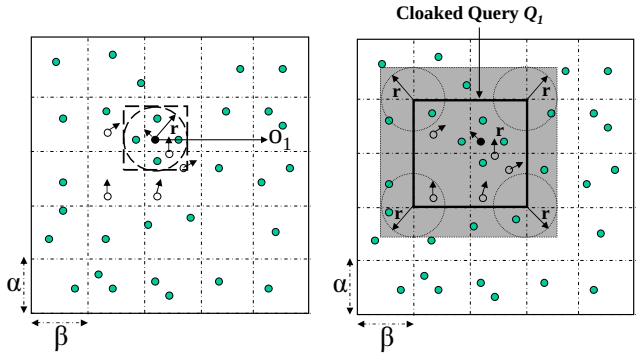
where $Cost_{bu}$ is the average cost of a single bottom-up iteration. Similarly, if the cost of one top-down iteration is $Cost_{td} = \gamma \times Cost_{bu}$,

$$T_{td} = \{a - \{\sqrt{\frac{k}{n}} - 1\} + b - \{\sqrt{\frac{k}{n}} - 1\}\} \times \gamma \times Cost_{bu} \tag{7}$$

For any anonymization request, the hybrid cloaking algorithm proceeds in a top-down manner if $T_{td} < T_{bu}$, otherwise, it proceeds in a bottom-up manner.

4.2 Integrating Spatial Cloaking with Temporal Cloaking

All cloaking algorithms we have discussed so far start composing the minimal spatial cloaking box that meets both privacy and QoS requirements immediately upon the arrival of a new request message, regardless whether the top-down or bottom-up cell composition is used. Recall that some messages may have to be dropped during spatial cloaking due to the fact that the algorithm cannot find the cloaking area that meets both privacy requirement and the maximum spatial resolution requirement specified by the mobile user. Instead of dropping the message, we can improve the situation by invoking the temporal cloaking to introduce some delay in terms of when to start the spatial cloaking process within the maximum temporal resolution constraint. For example, if we delay the start of spatial cloaking for γ time units ($0 \leq \gamma \leq d_t$), more mobile users may issue requests over the same area, and lead to higher probability for more messages to be perturbed, providing higher anonymization success rate. The critical challenge is how to set the appropriate γ value. If $\gamma = 0$ the spatio-temporal cloaking is reduced to immediate spatial cloaking. If



(a) Actual Query (b) Anonymized Query

Fig. 7: Anonymous Query Processing

$\gamma = d_t$, the cloaking will be performed right before the expiration of the message (i.e., after a maximum allowed delay of d_t time units. However this extreme setting will result in higher latency which in many cases are unnecessary. Given a request message and its current cell cid, let C_{max} be the neighboring cell of cid with the highest object count. In PRIVACYGRID, we determine the timing for starting the deferred cloaking process based on a number of parameters. Concretely, we perform the spatial cloaking for a new message if the total object count of the current cell cid and the neighboring cell C_{max} is larger than or equal to a system defined fraction of k, say θ, namely $MN(cid) + MN(C_{max}) \geq \theta \times k$. $\theta < 1$ is a system parameter that adjusts the amount of anonymization messages to be deferred. Smaller θ values push more messages to be processed immediately upon arrival. We can set θ at initialization time based on experimental studies or have it adaptively tuned during runtime by observing the rate of successful anonymizations with different θ values. A similar threshold value may be maintained for the l-diversity specifications too.

5 Processing Perturbed Location Queries

We briefly describe the anonymous query processing mechanisms required at the LBS server in order to aid processing of queries associated with cloaked spatial regions instead of spatial points. Figure 7 displays an object o_1 which requests for all static objects (e.g. gas stations) within the distance r from its curent position. Figure 7(a) displays the Minimum Bounding Rectangle (MBR) which forms the result set to be explored for the actual query. The cloaked query region identified by the location anonymization server is as shown in Figure 7(b). The actual object o_1 which makes the query request may be present anywhere within the cloaked query region, even at the any of the corner points of the region. Thus the query processor needs to explore the region at a maximum distance r from each corner point to ensure that the probability of relevant results being excluded from the evaluation is zero. The shaded area in the figure displays the query result evaluated using the cloaked query region. As is clearly evident from the figure, the query result will include all relevant results for the original query Q_1. This clearly illustrates the need for finding smaller cloaking regions as unnecessarily large cloaking regions will lead to larger result sets. It is important to note that no other optimizations of any kind during query processing can guarantee all relevant results will be included in the returned candidate result set.

Theorem 1. *The MBR (as evaluated in Figure 7(b)) for the cloaked query Q_1 includes all relevant results for the actual query Q_1.*
Proof Skipped.

6 Experimental Evaluation

We divide the experimental evaluation of PRIVACYGRID into two components: the effectiveness of our cloaking algorithms in terms of privacy and quality requirements, and their performance in terms of time complexity and scalability. Before reporting our experimental results, we first describe our evaluation metrics and the experimental setup, including the road-network based mobile object simulator used in the experiments.

6.1 Evaluation Metrics

We define the following metrics to evaluate the effectiveness and efficiency of PRIVACYGRID location cloaking algorithms.

Anonymization Success Rate: The anonymization success rate is also referred to as the anonymization hit rate. The success rate of a cloaking algorithm measures its ability to cloak messages according to the privacy requirements – the *k*-anonymity value and the *l*-divisity value – and the QoS requirement – the *maximum spatial resolution* value and the *maximum temporal resolution* value. We define the anonymization success rate by measuring the fraction of messages cloaked successfully by an algorithm among all anonymization requests. This is the most important measure for evaluating the performance of the cloaking algorithms. A primary goal of the cloaking algorithm is to maximize the number of messages perturbed successfully according to their privacy and QoS requirements. Hence, the higher success rate a location cloaking algorithm has, the more effective it is.

Relative Anonymity Level (RAL): This metric is used to measure the achieved anonymity level for successfully cloaked messages by the cloaking algorithm, normalized by the specified level of anonymity (k value) and the specified level of diversity (l value) in the mobile user's location privacy preference profile.

$$RAL = \frac{k'}{k} \times \frac{l'}{l} (k' \geq k, l' \geq l) \tag{8}$$

In PRIVACYGRID, the location cloaking algorithms aim at obtaining higher anonymity for the same cloaking area. However, excessive anonymity achieved at the cost of cloaking the location to a larger region hurts QoS during query processing. Hence, the lower the relative anonymity level (RAL), the better the performance of the algorithm.

Relative Spatial Resolution (RSR): This metric measures the ability of the spatial cloaking algorithm to provide the

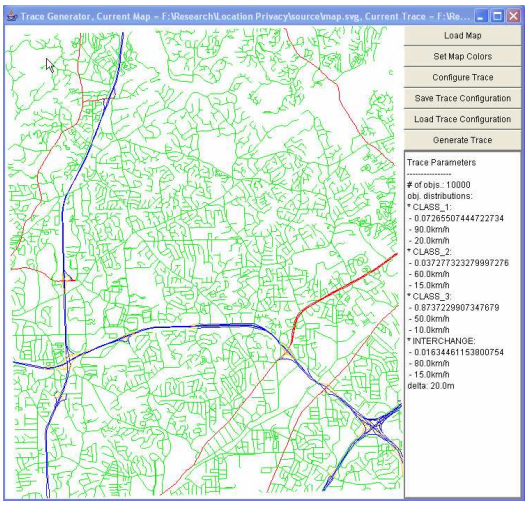


Fig. 8: Simulator for Experimental Setup

smallest cloaking area sufficient to meet the anonymity requirements. We calculate the relative spatial resolution by using the minimum spatial cloaking area, as calculated by the cloaking algorithm, normalized by the maximum allowed spatial cloaking area defined by the specified maximum spatial resolution $\{d_x, d_y\}$.

$$RSR = \sqrt{\frac{2 \times d_x \times 2 \times d_y}{Area(selectedCells)}} \quad (9)$$

The relative spatial resolution has to be greater than one in all cases for successfully anonymized messages. Higher relative spatial resolution measure implies that the cloaked spatial region is smaller and the cloaking algorithm is more effective.

Message Anonymization Time: This metric measures the run-time performance of the cloaking algorithm in terms of time complexity. Efficient cloaking implies that the cloaking algorithm spends less time but perturbs more messages.

6.2 Experimental Setup

We extend the simulator from [14] to evaluate the effectiveness and performance of PRIVACYGRID cloaking algorithms. The simulator generates a trace of cars moving on roads, and generates requests based on the position information from the trace. The trace generated by the simulator simulates a real-world road network obtained from maps available at the National Mapping Division of the USGS [7] in Spatial Data Transfer Format (SDTS) [6]. A transport layer of 1:24K Digital Line Graphs (DLGs) is used to extract the road-based network. The data is converted to the Scalable Vector Graphic (SVG) [5] format using the GlobalMapper tool [2]. The simulator extracts the road network based on three types of roads – *expressway*, *arterial* and *collector* roads. Traffic volume data in [16] is used to estimate the number of cars for different road classes. Cars are randomly placed on the road network according to the traffic densities and are moving on the roads. At intersections, they move in one direction or the other. The simulator attempts to keep the number of cars on each type of roads constant with time. Our experimentation

Road type	Expressway	Arterial	Collector
Mean of car speeds (km/h)	90	60	50
Std. dev. of car speeds (km/h)	20	15	10
Traffic volume data (cars/h)	2916.6	916.6	250

Table 1: Motion Parameters

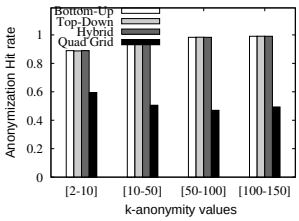
uses a map from Chamblee region of Georgia (Figure 8) to generate the trace used in this paper, which covers a region of approximately 168 km^2. Most of our experiments use the trace with a duration of two hours. We simulate the movement of a set of 10,000 cars on the road network for Chamblee. Table 1 lists mean speeds, standard derivation and traffic volume values for each road type. Each car generates a set of messages during the simulation. By default, each message specifies an anonymity level k from the range of $[1, 150]$ using a zipf parameter of 0.6 with higher k being the most popular. The maximum spatial and temporal resolution values of the message are selected independently using normal distributions with $600m$ as the default mean spatial resolution and $30m^2$ as the variance in maximum spatial resolution. The default mean temporal resolution is set to be 15s with $12s^2$ variance in temporal resolution. Though all parameters take their default values if not stated otherwise, the settings of many parameters will be changed in different experiments to show the impact of these parameters on the effectiveness and efficiency of the algorithms.

6.3 Experimental Results

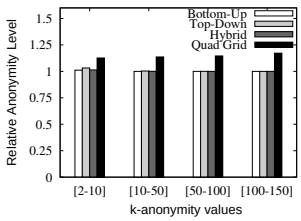
Our experimental evaluation of the PRIVACYGRID algorithms consists of three parts. First, we evaluate the effectiveness of the location anonymization algorithms by measuring anonymization hit rate (success rate), relative anonymity level obtained, average cloaking time and relative spatial resolution and observe how these parameters behave when we vary the settings of a number of parameters, such as grid cell size, the user-specified anonymity level k, and the user-specified maximum spatial resolution $\{d_x, d_y\}$. Then we evaluate the scalability of the algorithms in terms of cloaking time and update cost by varying the number of mobile users. Finally we evaluate the effectiveness of combining temporal cloaking with spatial cloaking by measuring the anonymization success rate (fraction of messages anonymized) when varying both the maximum temporal resolution values and the maximum spatial resolution values. Our results show that the PRIVACYGIRD dynamic grid cloaking algorithms are fast, effective, scalable and outperform all existing location cloaking approaches in terms of both anonymization success rate and cloaking time in the presence of larger range of k values.

6.3.1 Varying Size of Grid Cells

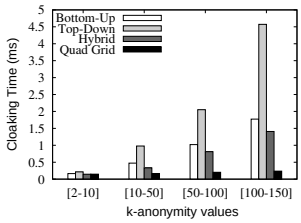
This set of experiments aims at measuring cloaking time, anonymization hit rate (success rate), relative anonymity level and relative spatial resolution obtained by using different settings of grid cell size. Figure 9 shows the results measured



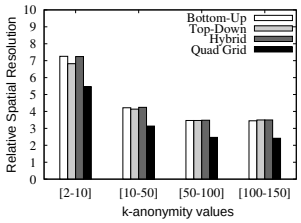
(a) Anonymization Hit Rate



(b) Relative Anonymity Level

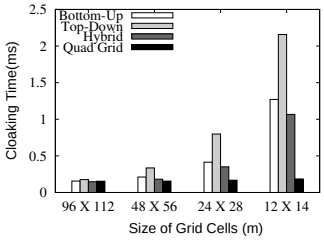


(c) Message Anonymization Time



(d) Relative Spatial Resolution

Fig. 10: Results with Varying Anonymity Levels



(a) Message Anonymization Time

	Cell Size (meters)	Hit Rate	RAL	RSR
Bottom-Up	24 X 28	91.5%	1.002	3.795
Approach	48X 56	91.4%	1.0025	3.773
Top-Down	24 X 28	91.5%	1.003	3.795
Approach	48X 56	91.4%	1.01	3.715
Hybrid	24 X 28	91.5%	1.002	3.831
Approach	48X 56	91.4%	1.0021	3.775
Quad Grid	24 X 28	42.8%	1.149	2.722
Approach	48X 56	42.7%	1.153	2.685

(b) Other Metrics

Fig. 9: Results with Varying Size of Grid Cells

for four different settings of grid cell size: $[96m \times 112m]$, $[48m \times 56m]$, $[24m \times 28m]$, and $[12m \times 14m]$. Recall that a HGI of height *l* implies a lowest level grid comprising of $2^l X 2^l$ cells. Thus the four different grid cell sizes are equivalent to four different settings of the lowest level grid size, ranging from 128×128 cells to 1024×1024 cells. The user-defined anonymity levels (k value) for this set of experiments are chosen in the range [10 – 50] with a Zipf distribution using parameter 0.6, indicating that messages with higher anonymity levels are more popular.

Figure 9(a) shows that the quad grid cloaking algorithm is fast in terms of cloaking time and the cloaking time does not increase significantly with the decrease in the size of grid cells. However, both the bottom-up and top-down dynamic grid cloaking algorithms will incur relatively higher cloaking time (ms) with the decreasing sizes of grid cells. This is because more rows (or columns) need to be added (or removed) to obtain the optimal cloaking regions. Interesting to note is that the actual cloaking time of all dynamic approaches is still below 2.5 ms in all cases, and such low delays are hardly perceivable.

From Figure 9(b) we observe two interesting results. First, the anonymization hit rate, the relative anonymity level (RAL), and the relative spatial resolution (RSR) do not change much as we vary the size of grid cells. Second, given a fixed grid cell size, say $[24m \times 28m]$, we see sharp differences when comparing the Quad Grid cloaking approach with the dynamic grid cloaking approaches such as bottom-up, top-down and hybrid. The Quad Grid cloaking, though faster (recall Figure 9(a)), has only 43% of the messages being anonymized successfully, while all the dynamic approaches have similar but much higher rate of success ($> 91\%$). All the dynamic grid cloaking approaches give low relative anonymity levels, which are close to one, whereas the Quad Grid approach has about 15% higher relative anonymity level, indicating that it might be cloaking requests to unnecessarily larger spatial regions. This is confirmed by the relative spatial resolution (RSR) measurement, which is about 40% higher for the dynamic cloaking approaches when compared to the Quad Grid cloaking approach.

6.3.2 Varying User-defined Anonymity Level k

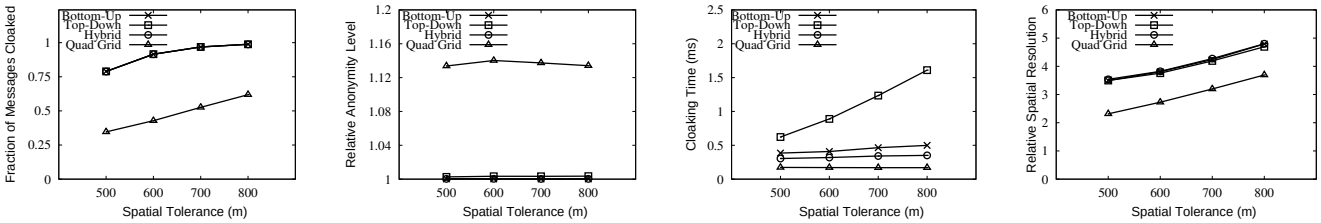
This set of experiments measures anonymization hit rate (success rate), relative anonymity level, cloaking time, and relative spatial resolution when varying k, the user-defined anonymity level, from various ranges: [2–10], [10–50], [50–100] and [100–150]. Spatial tolerance values for the anonymity ranges are 400m, 800m, 1200m and 1600m (mean values with 5% standarad deviation) respectively and are chosen to be large enough to theoretically allow cloaking of a large fraction of the messages. The results are as displayed in

Figure 10 shows that the Quad Grid approach is able to cloak only around 60% of the messages with anonymity level k set in the range of [2–10] and the success rate falls further to 45-50% with increasing k values. In contrast, the dynamic approaches cloak 90-99% of the messages within user-defined maximum spatial resolution values (Figure 10(a)).

From Figure 10(b), we see that the Quad Grid cloaking incurs higher relative anonymity level but all dynamic cloaking approaches have low relative anonymity levels (close to one), indicating that the anonymity levels obtained in all perturbed messages (k' values) are very close to the user-defined k.

Figure 10(c) shows the impact of varying the user-defined anonymity level (k values) on the cloaking time of all algorithms. The quad grid cloaking algorithm is the fastest and its cloaking time does not increase much with the increase in the user-defined k values. Though all dynamic cloaking algorithms will incur relatively higher cloaking time (ms) with the increasing k values, the amount of increase in cloaking time for bottom-up and hybrid is much slower when compared to the top-down approach. It is important to note that the cloaking time for the worst case (where the top down approach is used) is still around 4.5 ms for k values in [100–150] (with higher k being more popular), which is hardly perceivable by most users.

Figure 10(d)) displays the impact of changing k values on relative spatial resolution (RSR) obtained for the perturbed messages. Clearly, the dynamic grid cloaking algorithms have considerably higher RSR (28-43%) than the Quad Grid approach for all k values, though RSR values decrease as the k values become larger.



(a) Anonymization Hit Rate (b) Relative Anonymity Level (c) Message Anonymization Time (d) Relative Spatial Resolution

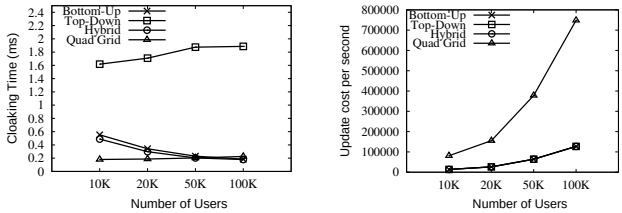
Fig. 11: Results with Varying Spatial Tolerance

6.3.3 Varying Spatial Tolerance

This set of experiments examines the performance of the algorithms by varying the maximum spatial resolution settings and measures the anonymization hit rate (success rate), relative anonymity level, cloaking time, and relative spatial resolution. Messages are generated with anonymity level k from the range [10–50] with Zipf distribution using parameter 0.6, favoring messages with higher k values. We vary the maximum spatial resolution value from 500m to 800m (mean values) with 5% standard deviation and examine the effect of different settings of maximum spatial resolution on the effectiveness of both the Quad Grid and the Dynamic Grid cloaking approaches. Figure 11 displays the results. The dynamic approaches are able to cloak all messages which can be theoretically cloaked for each maximum spatial resolution value, whereas the Quad Grid approach fails to cloak a large number of messages (40% less as shown in Figure 11(a)). Figure 11(b) shows that the relative anonymity levels for all cloaking algorithms do not change much when the user-defined maximum spatial resolutions change significantly. Figure 11(c) shows that only the top-down cloaking algorithm increases the cloaking time as the maximum spatial resolution values increase, while other cloaking algorithms are not very insensitive to the changes in the maximum spatial resolution values. Finally, Figure 11(d) shows that with the increase in the maximum spatial resolution values, the relative spatial resolution (RSR) values for all cloaking algorithms will increase proportionally with a close to constant gap between the Quad Grid approach and the dynamic grid algorithms.

6.3.4 Scalability

Finally we report the set of experiments designed to study the scalability of the PRIVACYGRID system with respect to the changing number of mobile users. Obviously, as the number of users in the system increases, we can expect the cloaking time for all algorithms to decrease as messages will be anonymized more easily, but the update costs for the grid-based structures will increase. We use a similar setup to that in Section 6.3.3 with the mean spatial resolution fixed at 800m with 5% standard deviation. We vary the number of users from 10K to 100K and observe the effect on the cloaking time and update cost. Figure 12 shows the measurement results. From Figure 12(a) we observe a number of interesting results. First, the amount of differences in cloaking time among the algorithms changes slightly with the increase in the number of mobile users. Second, the cloaking time for the Quad Grid approach is less sensitive to the increase in the number of users. Third, the top-down approach shows a slow increase in cloaking time with the increase in the number of mobile users in the system. This is because the approach requires more iterations as messages can be cloaked to smaller spatial regions now. However, the bottom-up approach displays a reverse trend – the cloaking time decreases as the number of users increases. This is because higher density of mobile users per grid cell will enable the bottom up cloaking to find the minimal cloaking box faster. Finally, we observe that the hybrid approach adapts well to the increase in the number of users, offering similar performance as the bottom-up approach in terms of cloaking time.



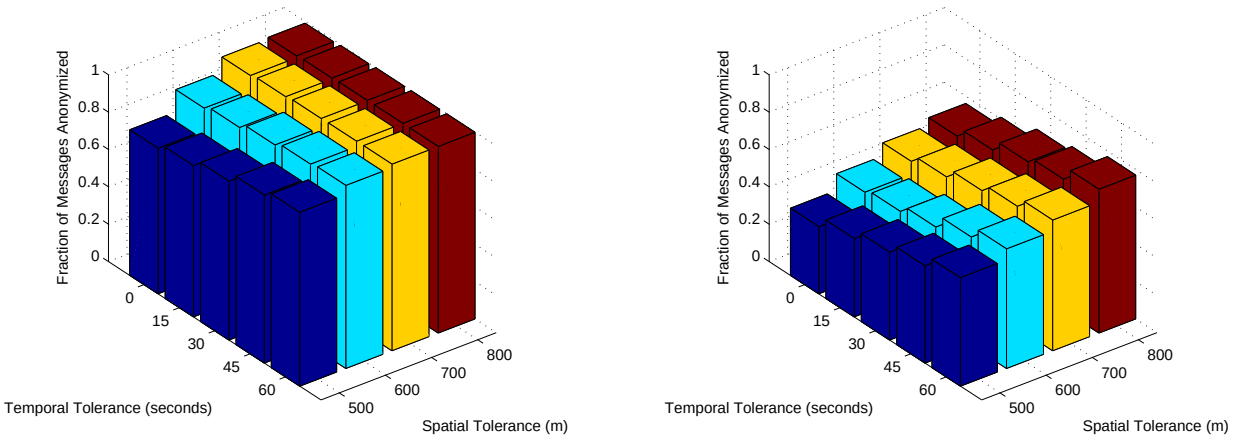
(a) Message Anonymization Time (b) Update Costs

Fig. 12: Results with Varying Number of Users

Figure 12(b) measures the total number of updates per second required to update the grid-based data structures as the number of mobile users increases. For this experiment, the grid index is maintained as a main memory data structure. Each car provides a location update to the system after moving a distance of 20m. We observe that the Quad Grid approach uses the HGI data structure and requires a large number of updates as the number of users increases. The HGI used in this experiment is a nine level grid index, requiring an average of 10–11 updates per location update request. In contrast, the dynamic cloaking approaches use the simple grid index, requiring only 1.8–1.9 updates per location update request, which is significantly lower than the Quad Grid approach in terms of update cost.

6.3.5 Effects of Maximum Temporal Resolution

This set of experiments is dedicated to study the effects of utilizing maximum temporal resolution values to delay the message anonymization process within an acceptable time period. Again we use the same experimental setup as in Section 6.3.3. We measure the success rate by varying both maximum temporal resolution d_t from 15 seconds to 60 seconds (mean values with 5% standard deviation) and varying the maximum



(a) Dynamic Cloaking

(b) Quad Grid Cloaking

Fig. 13: Effects of Temporal Tolerance

spatial resolutions from 500m to 800m. Figure 13(a) displays the results for the dynamic grid cloaking approaches and Figure 13(b) shows the results for the Quad Grid approach. We observe that the use of maximum temporal resolution helps increase the fraction of messages being cloaked for both the dynamic approaches and the Quad Grid approach by 10–20%.

7 Related Work

The *k-anonymity* approach to privacy protection was first developed for protecting published medical data [23, 22]. k-anonymity guarantees the inability to distinguish an individual record from atleast $k-1$ other records. [9, 18] attempt to provide solutions for *optimal* k-anonymization. Personalization of privacy requirements has attracted attention recently [14, 26]. Other related work includes anonymization of high dimensional relations [8] and extending the concept of k-anonymization via *l-diversity* [20], *t-closeness* [19] and *m-invariance* [27].

The concept of location k-anonymity was introduced in [16] where k is set to be uniform for all users. The concept of personalized location k-anonymity with customizable QoS specifications, first introduced in [14], is adopted by several others [21, 15]. Most solutions for location privacy adopt the trusted third party model which has been successfully deployed in other areas such as Web browsing [1]. Two representative approaches to personalized location anonymization are the *CliqueCloak* algorithm introduced in [14] and the Capser system [21]. The *CliqueCloak* algorithm relies on the ability to locate a clique in a graph to perform location cloaking, which is expensive and shows poor performance when k is large. The Casper approach addresses location anonymization using the *pyramid* data structure and allows the system to quickly locate cloaking boxes. However, due to the coarse resolution of the pyramid structure and the lack of QoS support, the cloaking areas in Casper are much larger than necessary, leading to poor QoS perceived by the user.

8 Conclusion and Future Work

We have described PRIVACYGRID – a framework for supporting anonymous location-based queries in mobile information systems. This paper has made three unique contributions. First, we propose to use location k-anonymity and location l-diversity as the two location hiding measures and maximum spatial resolution and maximum temporal resolution as the two location service quality measures. Second, we develop the Quad Grid approach and three dynamic grid based spatial cloaking algorithms for providing location k-anonymity and location l-diversity in a mobile environment. The Quad Grid cloaking algorithm is fast but has lower anonymization success rate. The dynamic grid cloaking algorithms provide high anonymization success rate and yet are efficient in terms of both time complexity and update cost. Third but not the least, we incorporate the maximum temporal resolution into the location cloaking process, which leads to further increase in the success rate of location anonymization by introducing controlled delay in terms of when to start location anonymization. We also described the PRIVACYGRID mechanisms for processing perturbed range queries. Our experimental evaluation shows that the PRIVACYGRID approach is efficient and effective for performing personalized location anonymization, while providing optimal location anonymity as defined by per user location privacy preference profiles.

Acknowledgement

This work is partially supported by grants from NSF CyberTrust, NSF SGER, NSF CSR, AFOSR, IBM SUR grant, and IBM faculty award. The authors would like to thank Bugra Gedik for providing the mobile-object simulator.

References

[1] Anonymous Web Surfing. http : //www.anonymizer.com.

[2] Global Mapper Software LLC. http : //www.globalmapper.com.

[3] Location Privacy Protection Act of 2001. http : //www.techlawjournal.com/cong107/privacy/location/s1164is.asp.

[4] Platform for Privacy Preferences (P3P) Project. http://www.w3.org/P3P/.

[5] Scalable Vector Graphics Format. http://www.w3.org/Graphics/SVG.

[6] Spatial Data Transfer Format. http://www.mcmcweb.er.usgs.gov/sdts/.

[7] U.S. Geological Survey. http://www.usgs.gov.

[8] C. Aggarwal. On k-Anonymity and the Curse of Dimensionality. In *Proceedings of the International Conference on Very Large Data Bases*, pages 901–909, 2005.

[9] R. Bayardo and R. Agrawal. Data Privacy Through Optimal k-Anonymization. In *Proceedings of the International Conference on Data Engineering*, pages 217–228, 2005.

[10] A. Beresford and F. Stajano. Location Privacy in Pervasive Computing. *Pervasive Computing, IEEE*, 2(1):46–55, 2003.

[11] Computer Science and Telecommunications Board. IT Roadmap to a Geospatial Future. *The National Academics Press*, 2003.

[12] S. Duri, M. Gruteser, X. Liu, P. Moskowitz, R. Perez, M. Singh, and J. Tang. Framework for Security and Privacy in Automotive Telematics. In *Proceedings of the Second International Workshop on Mobile Commerce*, pages 25–32, 2002.

[13] B. Gedik and L. Liu. MobiEyes: Distributed Processing of Continuously Moving Queries on Moving Objects in a Mobile System. In *Proceedings of the International Conference on Extending Database Technology, EDBT*, 2004.

[14] B. Gedik and L. Liu. Location Privacy in Mobile Systems: A Personalized Anonymization Model. In *Proceedings of IEEE International Conference on Distributed Computing Systems*, pages 620–629, 2005.

[15] G. Ghinita, P. Kalnis, and S. Skiadopoulos. PRIVE: Anonymous Location-Based Queries in Distributed Mobile Systems. In *Proceedings of the Sixteenth International World Wide Web Conference*, pages 371–380, 2007.

[16] M. Gruteser and D. Grunwald. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *Proceedings of the International Conference on Mobile Systems, Applications and Services*, pages 31–42, 2003.

[17] M. Gruteser and D. Grunwald. Enhancing Location Privacy in Wireless LAN Through Disposable Interface Identifiers: A Quantitative Analysis. *Mobile Networks and Applications*, 10(3):315–325, 2005.

[18] K. LeFevre, D. DeWitt, and R. Ramakrishnan. Incognito: Efficient Full-Domain K-Anonymity. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 49–60, 2005.

[19] N. Li, T. Li, and S. Venkatasubramanian. t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. In *Proceedings of International Conference on Data Engineering*, 2007.

[20] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-Diversity: Privacy Beyond k-Anonymity. In *Proceedings of the International Conference on Data Engineering*, 2006.

[21] M. Mokbel, C. Chow, and W. Aref. The New Casper: Query Processing for Location Services without Compromising Privacy. In *Proceedings of the International Conference on Very Large Data Bases*, pages 763–774, 2006.

[22] L. Sweeney. Achieving k-Anonymity Privacy Protection Using Generalization and Suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):571–588, 2002.

[23] L. Sweeney. k-Anonymity: A Model for Protecting Privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[24] S. Tanimoto and T. Pavlidis. A Hierarchical Data Structure for Picture Processing. *Computer Graphics and Image Processing*, 4(2):104–119, 1975.

[25] R. Want, A. Hopper, V. Falcão, and J. Gibbons. The Active Badge Location System. *ACM Transactions on Information Systems (TOIS)*, 10(1):91–102, 1992.

[26] X. Xiao and Y. Tao. Personalized Privacy Preservation. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 229–240, 2006.

[27] X. Xiao and Y. Tao. m-Invariance: Towards Privacy Preserving Re-publication of Dynamic Datasets. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, 2007.