Supporting Anonymous Location Queries in Mobile Environments with PrivacyGrid

Bhuvan Bamba, Ling Liu, Peter Pesti, Ting Wang
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332, USA
{bhuvan, lingliu, pesti, twang}@cc.gatech.edu

ABSTRACT

This paper presents PrivacyGrid – a framework for supporting anonymous location-based queries in mobile information delivery systems. The PrivacyGrid framework offers three unique capabilities. First, it provides a location privacy protection 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, it provides fast and effective location cloaking algorithms for location k-anonymity and location l-diversity in a mobile environment. We develop *dynamic* bottom-up and top-down grid cloaking algorithms with the goal of achieving high anonymization success rate and efficiency in terms of both time complexity and maintenance cost. A hybrid approach that carefully combines the strengths of both bottom-up and top-down cloaking approaches to further reduce the average anonymization time is also developed. Last but not the least, PrivacyGrid incorporates temporal cloaking into the location cloaking process to further increase the success rate of location anonymization. We also discuss PrivacyGrid mechanisms for supporting anonymous location queries. Experimental evaluation shows that the PrivacyGrid approach can provide close to optimal location k-anonymity as defined by per user location P3P without introducing significant performance penalties.

Categories and Subject Descriptors

H.2.7 [**Database Management**]: Database Administration—*Security, integrity, and protection*; H.2.8 [**Database Management**]: Database Applications—*Spatial databases and GIS*

General Terms

Algorithms, Experimentation, Performance, Security

Keywords

Location Privacy, k-Anonymity, l-Diversity

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 [8]. Though LBSs provide many new opportunities, the ability to locate mobile users also presents new threats – the intrusion of location privacy [7, 12]. According to [7], location privacy is defined as the ability to prevent unauthorized parties from learning one's current or past location. Location privacy threats refer to the risk 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 using the device [13]. For example, location information can be used to spam users with unwanted advertisements or to learn about users' medical conditions, unpopular political or religious views. Inferences can be drawn from visits to clinics, doctor's offices, entertainment clubs or political events. Public location information can lead to physical harm, such as stalking or domestic abuse.

Several approaches have been proposed for protecting the location privacy of a user. 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. For those LBSs that require true user 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 location l-diversity, thus enabling anonymous location-based queries in mobile information delivery systems.

Perfect privacy is clearly impossible as long as communication takes place. An important question here is *how much privacy protection is necessary?* Moreover, users often have varying privacy needs in different contexts. In PrivacyGrid, we propose to use location k-anonymity and location l-diversity as two quantitative metrics to model the location privacy requirements of a mobile user. In the context of LBSs and mobile users, location k-anonymity refers to k-anonymous usage of location information. A user is considered location k-anonymous if and only if the location information sent from the mobile user to a LBS is indistinguishable from the location information of at least $k-1$ other users. Location l-diversity is introduced to strengthen the privacy protection of location k-anonymity in situations where location information shared by the k users is sensitive. Increasing l value to two or higher significantly reduces

WWW 2008, April 21–25, 2008, Beijing, China.
ACM 978-1-60558-085-2/08/04.

the probability of linking a static location or a symbolic address (such as church, restaurant, doctor's office) to a mobile user. Location perturbation is an effective technique for implementing *personalized* location k-anonymity and location l-diversity. Cloaking methods typically perturb the location information by reducing its resolution in terms of time and space, referred to as spatial cloaking and temporal cloaking respectively [12].

In this paper, we present the PrivacyGrid approach to support anonymous location-based queries in mobile information delivery systems. The design of PrivacyGrid provides a unified and yet effective location anonymization framework for all types of location queries so that mobile users can enjoy the convenience of LBSs without revealing their exact location information. We make three unique contributions in this paper. 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 QoS 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 fast and effective cloaking algorithms for providing location k-anonymity and location l-diversity while maintaining the utility of LBSs. Concretely, our *dynamic expansion* technique for bottom-up grid cloaking and *dynamic reduction* technique for top-down grid cloaking provide high anonymization success rate and yet are efficient in terms of both time complexity and maintenance costs. We also propose a hybrid approach that combines the bottom-up and top-down search of location cloaking regions to further lower the average anonymization time. Last but not the least, we incorporate temporal cloaking functionality into the PrivacyGrid location perturbation process. Deferred processing of location anonymization requests within the temporal delay constraint further enhances the anonymization success rate while maintaining the desired quality of service (QoS). Our discussion on the new capabilities required for processing anonymous location queries exhibits the benefits of using small cloaking regions for anonymous query processing. The PrivacyGrid approach is evaluated through extensive experimentation, thus verifying that PrivacyGrid location cloaking algorithms can provide close to optimal location anonymity as defined by per user location P3P without introducing significant performance penalties.

2. RELATED WORK

The k-anonymity approach to privacy protection was first developed for protecting published medical data [19]. k-anonymity guarantees the inability of the adversary to distinguish an individual record from at least $k-1$ other records. [6, 15] provide solutions for *optimal* k-anonymization. Personalization of privacy requirements has attracted attention recently [10, 20]. Other related work includes anonymization of high dimensional relations [4] and extending the concept of k-anonymization via l-diversity [17], t-closeness [16] and m-invariance [21].

The concept of location k-anonymity was introduced in [12] where k is set to be uniform for all users. The concept of personalized location k-anonymity with customizable QoS specifications, first introduced in [10], is adopted by several others [18, 11]. Most popular solutions for location privacy [12, 10, 18] have adopted the trusted third party anonymization 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 [10] and the Casper system [18]. 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 for large k. The Casper approach performs the location anonymization using the quadtree-based *pyramid* data structure, allowing fast cloaking. However, due to the coarse resolution of the pyramid structure and lack of mechanisms to ensure QoS and constrain the size of the cloaking region, the cloaking areas in Casper are much larger than necessary, leading to poor QoS perceived by the users. Our experiments show that the PrivacyGrid approach outperforms Casper and other existing location anonymization approaches in terms of efficiency and effectiveness, producing cloaking regions that meet both location privacy and location service quality requirements.

In contrast to the trusted third party anonymizer model, a couple of research projects, with Prive [11] being the most representative one, attempt to remove the trusted third party anonymizer by relying on a decentralized cooperative peer to peer model and the existence of a trusted centralized certification server. The main technical challenge handled in this work involves dynamic formation of nearby peer groups that can perform location anonymization for each other. In fact, the PrivacyGrid approach can be easily adapted to such settings to perform the actual location cloaking among selected peer groups. Another thread of efforts is to perform location obfuscation at the mobile clients by introducing random noises or utilizing nearby landmarks [14, 9], assuming mobile clients have sufficient computation and communication resources to participate in both location anonymization and anonymous query processing tasks.

3. PRIVACYGRID: AN OVERVIEW

We assume that the LBS system powered by PrivacyGrid consists of mobile users, a wireless network, location anonymization servers and LBS servers. Mobile users communicate with the LBS servers via one or more PrivacyGrid location anonymization servers by establishing an authenticated and encrypted connection to the anonymization server. 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 location anonymization for both location queries and location updates from these mobile users. Each location anonymization server has access to all publicly available data which can be used for ensuring location l-diversity for user requests.

3.1 System Architecture

The PrivacyGrid system promotes a three-tier architecture for supporting anonymous information delivery in a mobile environment, as shown in Figure 1. The top tier is the location P3P user profile model that captures users' personalized location privacy requirements. The middle tier comprises of the location perturbation service typically provided by a trusted third party location server, specialized in location tracking and anonymization service. The third tier is dedicated to the transformation of raw location queries to anonymous location queries, enabling the processing of cloaked location queries at the individual LBS providers. Each participating LBS provider will need to provide anonymous query processing support and cooperate with the location anonymizer to provide the desired location privacy protection for consumers. In the PrivacyGrid development,

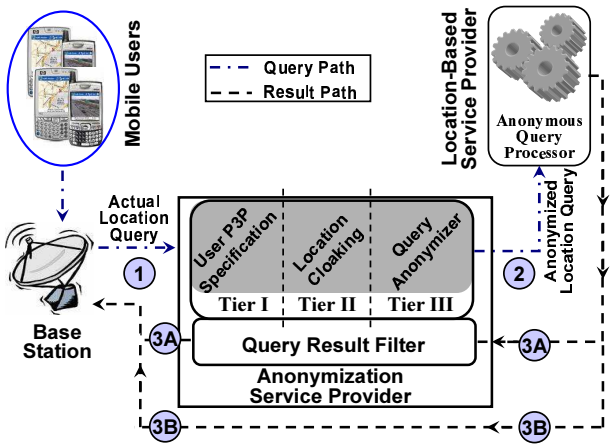


Figure 1: System Architecture

we consider the anonymous query processing component as an integral part of the solution.

Our location P3P model allows mobile users to specify what, when, how and with whom their location information can be shared. In addition to the standard P3P specification [2], we add four location privacy specific measures, two for location hiding constraints and two for QoS constraints. The first measure is location k-anonymity, which allows a mobile user to control her state of being unidentifiable from a set of $k-1$ other users. The second measure is location l-diversity, which allows the mobile user to control her state of being unidentifiable from a set of l different physical locations (such as churches, clinics, business offices). This measure can be seen as a companion measure of location k-anonymity and is particularly useful in reducing the risk of unwanted location inference when there are k or more distinct users at a single physical location (such as a clinic or a political gathering). The third measure is *maximum spatial resolution*, which allows the mobile user to control the spatial resolution reduction within an acceptable QoS specific range. It can be changed or adjusted according to the type of location service, the time of day, month or year, and on a per message level. Similarly, the fourth measure is *maximum temporal resolution*, which controls the temporal delay acceptable for maintaining the desired QoS.

The location anonymization component anonymizes the location information from mobile users, by performing spatial and temporal cloaking, based on their P3P profiles before passing the location information to the actual LBS providers (path 2 in Figure 1). The detailed location cloaking algorithm design will be discussed in subsequent sections.

The query anonymizer component is responsible for transforming each raw location query into two components: (1) an anonymized location query by replacing the exact location of the mobile user with the location cloaking box produced by the location anonymization module; and (2) the filtering condition that will prune the candidate set of query results to produce the exact result to the original raw query posed by the mobile user. For those LBSs that offer location dependent services over public locations, such as restaurants, gas stations, offices and so forth, only anonymized location queries are passed from the location anonymizer to the respective LBS provider. Mobile users who allow their movements to be tracked by certain LBSs 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.

Upon receiving anonymized location queries, the LBS providers will invoke the anonymous query processing module. Anonymous location query processing consists of two key steps. First, each anonymized query will be evaluated at the LBS provider to produce the set of candidate results. Second, the candidate set of query answers will need to be filtered to get exact results. There are a couple of alternative ways of performing the filtering step for anonymous query processing. For example, one can choose to have the location anonymizer as the middleman between mobile users and individual LBS providers such that the filtering task is carried out at the location anonymizer and the exact query result is generated and returned to the mobile user through the location anonymizer (path 3A in Figure 1). This approach adds additional load and bottleneck to the location anonymizer. Alternatively, filtering can be performed at the mobile client. In this case, for each location query received by the location anonymizer, the anonymized query will be passed to the LBS provider, and the filter condition will be returned to the mobile user who issued the query. The LBS provider will pass the candidate set of answers to be filtered at the client (path 3B in Figure 1). Filtering at the client will introduce additional communication and processing overhead. Thus it is critical to develop techniques that can minimize the amount of processing to be performed at the mobile client side. We will dedicate Section 6 for discussing issues related to processing of anonymous location queries.

3.2 Location Privacy Requirements

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

Personalized user privacy profile: We provide two measures for mobile users to specify their location privacy requirements: location k-anonymity and location l-diversity. The former allows a mobile user to control her state of being not identifiable from a set of $k-1$ other users. The latter allows a mobile user to control her state of being traceable to a set of at least l distinct public locations, which are typically referred to as symbolic addresses. Location l-diversity is particularly useful in reducing the risks of unauthorized location inference when there are k or more distinct users at a single physical location (such as a clinic or a church). A mobile user may change her privacy preference level (k and l values) as often as required or on a per message basis.

QoS guarantees: In order to provide effective location cloaking, PrivacyGrid provides two QoS measures which allow a mobile user to specify critical QoS constraints. The first QoS measure is the maximum spatial resolution, indicating the amount of spatial inaccuracy the user is willing to tolerate to maintain acceptable QoS. The second QoS measure is the maximum temporal resolution, ensuring that the delay introduced for cloaking a request message should be within an acceptable time interval. By utilizing these two quality metrics, PrivacyGrid aims at devising cloaking algorithms that will find the smallest possible cloaking region meeting desired privacy levels for each location anonymization request.

Dynamic tradeoff between privacy and quality: In PrivacyGrid, we stress that location perturbation algorithms should be capable of dynamically making trade-offs between privacy and QoS. Unnecessarily large cloaking boxes will lead to not only poor QoS for the mobile users but also larger result sets to be transported from the corresponding LBS provider and higher processing costs for filtering at either the mobile client side or at the location anonymizer, inevitably leading to larger delays for obtaining useful query results.

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 and capable of keeping the perceived delays due to location anonymization to a minimum.

3.3 Basic Concepts

In this section, we define 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 define 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: PrivacyGrid uses a personalized location privacy model. 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 $\langle obj_{id}, LBS_{info}, req_{id}, k, l, \{d_x, d_y, d_t\}\rangle$, where obj_{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 req_{id} is used to uniquely identify a service request posed by the user with the given obj_{id}. We use $k = 1$ and $l = 0$ or $l = 1$ as the default setting, neither anonymity nor diversity are considered. When k and l use their default settings, d_x, d_y, d_t are set to unknown value *null*.

3.4 Location Anonymization Server

In PrivacyGrid, each message m_s received by the anonymizer is of the form $\langle obj_{id}, req_{id}, \{x, y, t\}, k, l, \{d_x, d_y, d_t\}\rangle$. The obj_{id} and req_{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. 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 $\langle h(obj_{id}||req_{id}), \{X : [x_s, x_e], Y : [y_s, y_e], I : [t_s, t_e]\}\rangle$, 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 at least $k-1$ other mobile users and at least 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 refer to this process as spatio-temporal cloaking based message perturbation. We will describe the PrivacyGrid spatial cloaking algorithms for finding an ideal spatial cloaking box $\langle X : [x_s, x_e], Y : [y_s, y_e]\rangle$ that meets the k-anonymity and l-diversity requirements in Section 4.

3.5 Evaluation Metrics

In this section, we define several metrics that will be used to evaluate the effectiveness and efficiency of PrivacyGrid location cloaking algorithms. The anonymization success rate (ASR) and relative anonymity (relative diversity) levels are important measures for evaluating the effectiveness of the cloaking algorithms. Another useful effectiveness measure is the user location distribution (ULD) with respect to the cloaking box. It measures the strength of the location cloaking algorithm against inference attacks that attempt to guess the actual location of the mobile users with respect to the center of the cloaking region. Important efficiency measures include relative spatial resolution and message processing time.

Anonymization Success Rate (ASR): The primary goal of our location cloaking algorithms is to maximize the number of messages perturbed successfully while maintaining their anonymization constraints, specified by their privacy and QoS requirements. We define the anonymization success rate as the fraction of messages cloaked successfully by an algorithm with respect to the set of received anonymization requests. Let M denote the set of anonymization requests issued to the system. The set of messages that are successfully perturbed can be computed by $\{m_t | m_t = f_{cloak}(m_s), m_s \in M\}$, where $f_{cloak}(m_s)$ denotes a PrivacyGrid location cloaking algorithm. Thus, the anonymization success rate of $f_{cloak}(m_s)$ is defined as follows:
$ASR(f_{cloak}(m_s)) = \frac{|\{m_t | m_t = f_{cloak}(m_s), m_s \in M\}|}{|M|}$.

Relative Anonymity and Relative Diversity Levels: This metric measures the achieved anonymity and diversity levels for successfully cloaked messages normalized by the anonymity level k and diversity level l in the mobile user's location P3P. Intuitively, Relative Anonymity Level (RAL) measures the ratio of anonymity achieved by the cloaking algorithm to the user specified k-anonymity level, i.e., $\frac{k'}{k}$ and Relative Diversity Level (RDL) provides a similar measure for l-diversity $\frac{l'}{l}$, where k, l denote the user-defined values for a message m_s and k', l' denote the actual values obtained for the perturbed message $m_t (k' \geq k, l' \geq l)$. Note that for successful anonymization relative anonymity level cannot go below 1. Although, the location cloaking algorithms aim at obtaining higher anonymity for the same cloaking area, excessive anonymity achieved at the cost of cloaking the location to a much larger region leads to poor QoS and costly processing of anonymous queries. Hence, the lower the relative anonymity and relative diversity levels, the more effective the cloaking algorithm.

Relative Spatial Resolution (RSR): This metric measures the ability of a cloaking algorithm to provide the smallest cloaking area that meets the k-anonymity and l-diversity requirements. Given a message m_s and its perturbed version

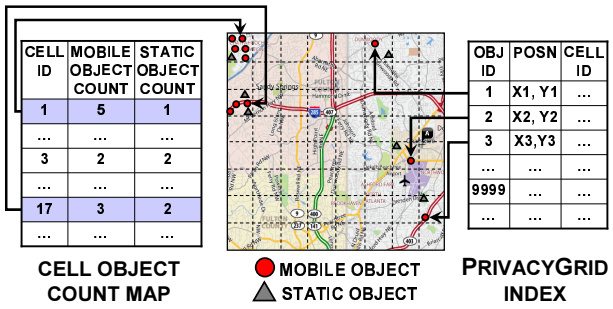


Figure 2: Data Structures for PrivacyGrid

m_t, we can measure the RSR by using the minimum spatial cloaking area as calculated by the cloaking algorithm. We define the RSR of a cloaking algorithm over a set of perturbed messages T as follows:

$RSR = \frac{1}{|T|} \sum_{m_t=f_{cloak}(m_s) \in T} \sqrt{\frac{2 \cdot m_s.d_x \cdot 2 \cdot m_s.d_y}{||B_{cl}(m_t).X|| \cdot ||B_{cl}(m_t).Y||}}$,

where (d_x, d_y) denote the maximal spatial resolution constraints for message m_s and $(B_{cl}(m_t).X, B_{cl}(m_t).Y)$ represent the dimensions of the cloaking box $B_{cl}(m_t)$. Higher relative spatial resolution measure implies that the cloaked spatial region is smaller relative to the user-specified maximum spatial resolution area and the cloaking algorithm is more effective.

User Location Distribution(ULD): This metric is used to measure the level of difficulty in inferencing the location of a user within the cloaking region shared by k users. We determine the user location distribution within the cloaked area by measuring the normalized distance of the actual user position to the center of the cloaked area for each successfully anonymized message. A uniform user location distribution implies the algorithm is more effective in terms of robustness against aforementioned inference attacks as the actual user location may lie anywhere within the cloaked region.

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 processing time to perturb messages.

4. PRIVACYGRID SPATIAL CLOAKING ALGORITHMS

In this section we introduce basic data structures and two dynamic grid-based location cloaking algorithms: bottom-up spatial cloaking and top-down spatial cloaking, both aiming at finding the smallest spatial cloaking box given the location of a mobile user. Ideally, this means that there exists no smaller spatial cloaking box that satisfies both location k-anonymity and location l-diversity requirements as well as the maximum spatio-temporal resolution constraints defined in the users' location P3P.

4.1 Data Structures

In PrivacyGrid the entire UoD is divided into grid cells of $\alpha \times \beta$ size by superimposing a grid on top of the UoD. α and β are system-controlled parameters that can be tuned based on a number of factors, such as the cloaking speed, granularity of cloaking boxes and average size of user-defined maximum spatial resolutions. Figure 2 illustrates the PrivacyGrid *Index (PI)* and the *Cell Object Count Map (COCM)* data structures.

PrivacyGrid Index (PI): The *PI* data structure allows for fast and efficient computation of object counts belonging to a particular region of the UoD. Figure 2 illustrates the composition and construction of this grid-based object index. The mobile object index and the still object index share the same data structure though maintained separately.

Cell Object Count Map (COCM): In addition to the mapping of each object to its current grid cell maintained by *PI*, we use this data structure to keep a count of the number of mobile objects and a count of the number of static objects (symbolic addresses, such as gas stations, restaurants, offices, and so forth) located in each grid cell. Maintaining this data structure allows for fast computation of the total number of mobile users and the total number of static objects located in a given spatial area. For each grid cell, the count of static objects remains unchanged most of the time. However, the count of mobile objects may change as mobile users move from one cell to another.

4.2 Bottom-Up Grid Cloaking

The bottom-up grid cloaking approach starts the cloaking process by taking the base cell containing the mobile object from which the cloaking request has originated as the candidate cloaking area. It performs two checks for each message with k or l higher than one in order to determine whether this candidate cloaking area meets the privacy and QoS requirements to be qualified as the ideal cloaking region. The first check is to determine if the current cell meets the user-specified maximum spatial resolution constraints. A second check looks up the cell object count map to determine if k-anonymity and l-diversity requirements are met. If the second check is successful, the candidate cloaking area will be chosen as the cloaking region. If not, the algorithm will start the cell expansion process to enlarge the candidate cloaking area to neighboring cells. The cell expansion process stops when both k-anonymity and l-diversity requirements for the cloaked message are met.

The detailed description of the bottom-up cloaking algorithm [5] is omitted here due to space constraints. The core idea behind bottom-up cloaking is the execution of dynamic cell expansion when the candidate cloaking region fails to meet the location privacy and QoS constraints. Dynamic cell expansion takes an opportunistic approach to expand the candidate cloaking region to any of the four neighboring set of cells.

The decision on which of the four cells to choose first is based on the object counts; the neighboring cell(s) with the highest object count will be chosen for expansion, generating the new candidate cloaking box. Each candidate cloaking box is composed of a set of adjacent cells and is encoded by the row and column index of the these selected cells. We maintain the *selected rows* and the *selected columns* for all candidate cloaking boxes in order to infer the selected cells of the final cloaking area. The current candidate cloaking box may be expanded 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 selecting and adding the rows or the columns which lead to the maximum object count collectively.

For every odd iteration, the algorithm determines whether to add a row or a column as the cloaking area may be expanded in any direction. 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 vertical or horizontal skew is introduced. For example, if the algorithm added a row during the previous it-

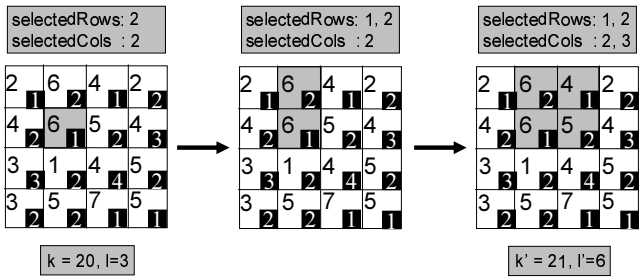


Figure 3: Bottom-Up Dynamic Expansion Example

eration, the current iteration would expand the cloaking box by addition of an adjacent column. The cell expansion steps are recursively repeated as long as the sum of object counts in *all* cells in selected rows and columns is less than the required k-anonymity and l-diversity levels. Upon meeting both the privacy and the QoS requirements, the algorithm uses the selected rows and columns to determine the grid cells forming the final cloaking area.

Figure 3 presents a walkthrough of the bottom-up dynamic expansion by example. In this example, we assume the user-defined k value is 20 and l value is 3. The cloaking request originates from the shaded cell with the mobile object count (number at top-left in each cell in Figure 3) of 6 and the still object count (number at bottom-right in each cell in Figure 3) of 1. It is located at the second row and the second column in the grid. We encode this candidate cloaking region by assigning the value of 2 to both *selectedRows* and *selectedCols* respectively. Clearly, this cell fails to meet both the k-anonymity and the l-diversity requirements. The algorithm starts the dynamic cell expansion process from the current cell. All neighboring cells of the shaded cell are considered. Given that the first row to the north increments the mobile object count to 12, the highest among all four neighboring cells, it is chosen to be the first cell for expansion. We add the row number 1 into the *selectedRows* to encode the new candidate cloaking region. Even though the total still object count in this candidate cloaking box is 3, satisfying the l-diversity requirement, the total mobile object count of 12 does not meet the user-specified k-anonymity requirement of 20. Thus the algorithm starts the next iteration of cell expansion. In this iteration, we choose one of the two neighboring columns of the candidate area to expand. We first consider the column to the left (first column in grid), which is not sufficient to meet the privacy requirements. Then we consider the column to the right (the third column in grid) which provides a cloaking area with the object count of k'=21, sufficient to meet the k-anonymity requirement. Thus the algorithm terminates and returns $selectedRows = \{1, 2\}$ and $selectedCols = \{2, 3\}$.

In fact, there are two ways in which cell expansion can be performed: (1) static cell expansion based on a pre-defined pattern, such as the quadtree-based grid expansion [18] or (2) dynamic cell expansion that opportunistically determines appropriate neighboring cells to expand the candidate cloaking region at run time. It is interesting to note that the static expansion approach promotes static cloaking following a pre-defined structure. For example, the pyramid approach in [18] uses the quadtree-based pyramid structure and some steps may expand the cloaking area to a pre-defined parent cell along the pyramid hierarchy, quadrupling the cloaked area, limiting the ability of the algorithm to explore all options of varying granularity. Though such a static cloaking approach is simple and fast, it suffers from a number of weaknesses. For example, pyramid cloaking expands the cloaking

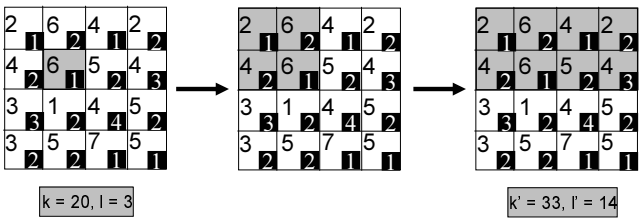


Figure 4: Pyramid Expansion Example

area to two or four times of the current size at each iteration, leading to a much bigger cloaking area and a much higher anonymity level than required, which hurts the QoS provided to the user and results in low anonymization success rate. In contrast to the static cloaking approach that selects the cloaking area using a pre-built cell composition structure, the dynamic expansion approach opportunistically expands the cloaking area, enabling the algorithm to quickly locate an ideal cloaking area that meets privacy and QoS requirements. Figure 4 shows pyramid expansion cloaking for the same example used in Figure 3. Clearly, pyramid expansion results in a much larger cloaking area with a much higher anonymity level than required. In contrast, the cloaking area produced by the dynamic bottom-up approach is much smaller as shown in Figure 3.

4.3 Top-Down Grid Cloaking

In some scenarios, a top-down cloaking approach performs anonymization faster compared to the bottom-up approach. For example, high k-anonymity and low maximal spatial resolution constraints may help the system quickly locate appropriate cloaking areas by using a top-down dynamic reduction approach as explained below. In PrivacyGrid, we design the top-down dynamic grid cloaking algorithm by utilizing the user-specified maximum spatial resolution. We first find the largest grid cell region within the user-specified maximum spatial resolution area, and encode the candidate cloaking area by a set of *selectedRows* and *selectedCols* in a similar manner as is done in the bottom-up approach. If the largest possible candidate cloaking box fails to meet the desired privacy requirements, the message cannot be cloaked using user-defined privacy and QoS requirements and the algorithm terminates. Otherwise, the top-down cloaking approach starts searching for the smallest possible cloaking box that meets the k-anonymity and l-diversity requirements by iteratively removing either an outermost row or column with the lowest object count from the candidate cloaking area. This iterative process shrinks the candidate cloaking box along one of the four directions and terminates when object counts in candidate cloaking area fall below the privacy requirement. Due to space constraints, we omit the detailed algorithm in this paper and refer the readers to our technical report [5] for further discussion.

Figure 5 displays an example walkthrough of the top-down dynamic cloaking algorithm. Recall the previous example where a mobile user in the cell at the intersection of the second row and second column issued a location anonymization request. The shaded area in the leftmost figure displays the largest possible cloaking area computed based on the user-specified maximum spatial resolution. Given that the mobile object count is 35 and the still object count is 18, cell reduction is performed repeatedly by first removing the third row (lowest mobile object count) and then removing the first column. The final cloaking box consists of the four cells marked by the first two rows and the second and third columns, with k'=21 and l'=6.

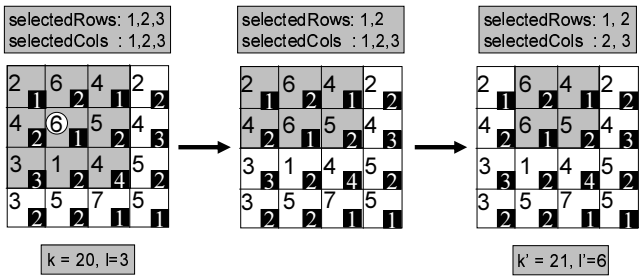


Figure 5: Top-Down Dynamic Reduction Example

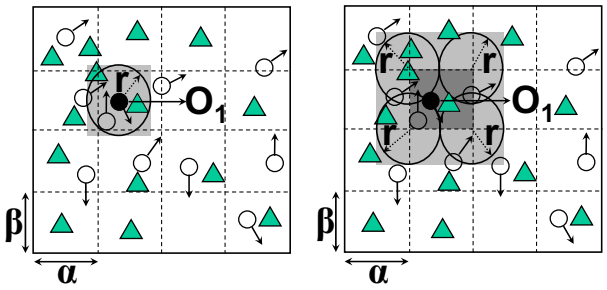
5. HYBRID CLOAKING

An obvious enhancement to bottom-up and top-down cloaking algorithms is the hybrid approach that takes advantage of the strengths of both approaches to produce a cloaking algorithm that *runs faster* than either of them. There are several ways to combine the bottom-up and top-down methods. In the first prototype of PrivacyGrid we adopt a most straightforward approach. The main idea is to provide guidelines on how to appropriately decide whether to proceed in a bottom-up or a top-down manner upon receiving a message cloaking request. For lower k-anonymity level and higher maximum spatial resolution values, the algorithm will benefit by proceeding in a bottom-up manner. On the other hand, for higher k-anonymity level and lower maximum spatial resolution values, the top-down approach clearly works faster than the bottom-up approach for finding the ideal cloaking box. Hence, the ability of the hybrid approach to identify whether it should proceed in a bottom-up or top-down manner upon receiving a cloaking request is crucial to its effectiveness. We provide some guidelines through a formal analysis of the hybrid cloaking algorithm in [5].

6. PROCESSING ANONYMOUS QUERIES

In PrivacyGrid, each location query will be sanitized through the location anonymization server before proceeding to the relevant LBS provider. The location anonymization engine will transform a raw location query into two components: anonymized query and privacy sensitive filter. The anonymized query can be submitted to the LBS providers by either the mobile user who issued the original query or by the anonymizer. However, the privacy-sensitive filter will be kept either at the location anonymization engine or on the mobile client side. Upon receiving an anonymous query, the LBS provider will invoke the anonymous location query processing engine residing at the LBS provider. Based on processing logic, we divide anonymous location queries into two classes: location anonymous queries over static objects (public location data) and location anonymous queries over moving objects (privacy sensitive location data). In either case, instead of exact answers, the anonymous location query processor will return approximate query answers that include the exact answer. The exact answer will be computed over the minimal approximate answer either at the mobile client or at the location anonymizer, which then forwards the exact answer to the mobile client.

Anonymous location query processing poses two unique challenges. First, we must produce the minimal set of approximate answers, aiming at minimizing the amount of additional communication and computation cost due to the location privacy support. Second, with anonymous location queries, the exact query result must be delivered to the mobile user. This may be done through the trusted location anonymizer, which performs the post-processing and



(a) Actual Query (b) Anonymized Query

Figure 6: Anonymous Query Processing

forwards the exact answer to the mobile user. Alternatively, the minimal approximate answer can be forwarded directly to the mobile client by the LBS provider through the subset of base stations that cover the region of the minimal query answers. The base stations may broadcast the set of approximate answers with the secure query identifier – the hash of $user_{id}$ and req_{id} (recall Section 3.4); only the mobile user who knows the *secret* identifier will be able to read the result set and perform the post-processing to produce the exact answer.

Most of the spatio-temporal query processing techniques developed in mobile data management field to date cannot be applied directly to anonymous query processing. We briefly describe below the anonymous query processing mechanisms required at the LBS server. In order to process range queries associated with cloaked spatial regions instead of spatial points and produce the minimal set of approximate answers for each anonymous query, the anonymous query processor needs to process each query in three steps: (1) determining the anonymous query minimum bounding rectangle (anonymous MBR) that contains the minimal set of approximate answers, (2) transforming the anonymous location query with anonymous MBR to an anonymity-aware range query, and (3) executing the range query using the traditional spatial query processor to produce the minimal set of approximate answers. Figure 6 illustrates this process using an example, where a moving object O_1 issues a query Q, requesting for some static objects (e.g. restaurants) within the distance r from its current position. Figure 6(a) shows the Minimum Bounding Rectangle (light grey rectangle) which forms the exact result set of the query Q using a traditional mobile query processor. Figure 6(b) shows the anonymous MBR (light grey rectangle) which produces the minimal set of approximate answers for the perturbed version of Q. The cloaked query region produced by the location anonymization server is shown as a dark grey rectangle contained inside the anonymous MBR. The mobile object O_1 could be present anywhere within the cloaked query region. Thus the computation of anonymous MBR will need to consider the four corner points of the cloaked rectangle and the entire region within a maximum distance r from each corner point to ensure that the anonymous MBR includes the exact answer of the query, as long as O_1 lies within the cloaked region. Different formulae will be required for computing anonymous MBR for different types of location queries. For example, [18] presents an efficient algorithm for privacy-aware processing of nearest neighbor queries (kNN). Due to space constraints, we omit the algorithms for computing the anonymous MBRs for range queries and other types of location queries and the correctness proof for these algorithms in terms of their minimality and inclusion of exact answer.

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

Table 1: Motion Parameters

7. EXPERIMENTAL EVALUATION

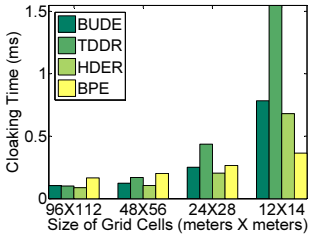
We divide the experimental evaluation of PrivacyGrid into two components: the effectiveness of our cloaking algorithms in terms of privacy and quality requirements and their performance in terms of time complexity and scalability. Before reporting our experimental results, we first describe the experimental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

7.1 Experimental Setup

We extend the simulator from [10] to evaluate the effectiveness and performance of PrivacyGrid cloaking algorithms. The simulator generates a trace of cars moving on a real-world road network, obtained from maps available at the National Mapping Division of the USGS [3], and generates requests based on the position information from the trace. The simulator extracts the road network based on three types of roads − *expressway*, *arterial* and *collector* roads. Our experimentation uses a map from the Chamblee region of Georgia, which covers an area of approximately 168 km^2, to generate traces for a two hour duration. Traffic volume data from [12] is used, generating a set of 10,000 cars on the road network for Chamblee. Table 1 lists mean speeds, standard deviation and traffic volume values for each road type. Cars are randomly placed on the road network according to the traffic densities, start moving on the roads and proceed in a random direction at the intersections. The simulator attempts to keep the number of cars on each type of road constant with time. Each car generates a set of messages during the simulation. By default, each message specifies an anonymity level k from the range $[2, 150]$ using a Zipf parameter of 0.6 favoring higher k values. Our experiments do not consider l-diversity requirements, but can be easily extended to measure relevant values. The maximum spatial resolution (or spatial tolerance) values of the messages are selected independently using normal distribution with default mean spatial resolution of $600m$ and 5% standard deviation. Though all parameters take their default values if not stated otherwise, the settings of many parameters are changed in different experiments to show the impact of these parameters on the effectiveness and efficiency of the algorithms.

7.2 Experimental Results

Our experimental evaluation of the PrivacyGrid algorithms consists of two parts. First, we evaluate the effectiveness of the location anonymization algorithms by measuring success rate, relative anonymity level, relative spatial resolution, average cloaking time, user location distribution and observe how these parameters behave when we vary the settings of a number of parameters, such as grid cell size, anonymity level k and 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. We briefly describe the impact of incorporating temporal cloaking on the anonymization success rate of our dynamic approaches. Our results show that the PrivacyGrid dynamic grid cloaking algorithms are fast, effective, scalable and outperform other location cloaking approaches in terms of both anonymization success rate and cloaking QoS in the presence of a larger range of k values.



(a) Anonymization Time

	Cell Size (meters)	Success Rate	RAL	RSR
BUDE	24 X 28	93.9%	1.0001	3.276
	48X 56	92.8%	1.0009	3.291
TDDR	24 X 28	93.9%	1.001	3.296
	48X 56	92.9%	1.004	3.243
HDER	24 X 28	93.9%	1.0001	3.305
	48X 56	92.9%	1.001	3.306
BPE	24 X 28	40.9%	1.17	2.366
	48X 56	40.9%	1.176	2.327

(b) Other Metrics

Figure 7: Results with Varying Size of Grid Cells

7.2.1 Varying Size of Grid Cells

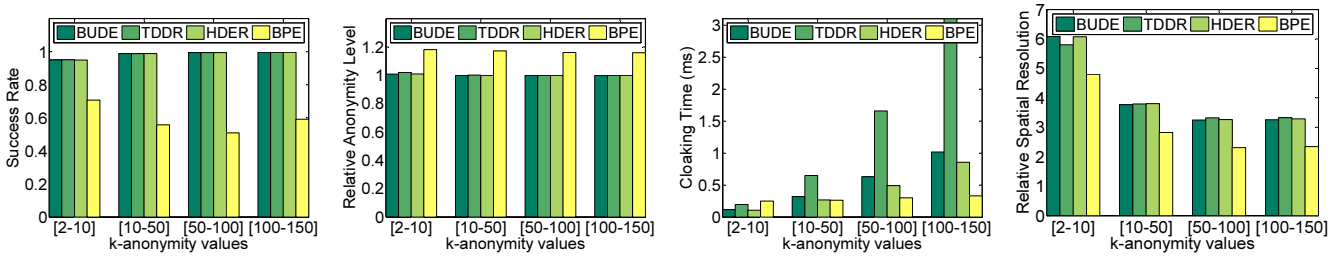
This set of experiments aims at measuring cloaking performance obtained by using different settings of grid cell size. Figure 7 shows the results measured for different settings of grid cell size which are equivalent to different settings for the grid size, ranging from 128×128 cells to 1024×1024 cells. The user-defined anonymity levels for this set of experiments are chosen in the range $[10 - 50]$ with a Zipf distribution using parameter 0.6 favoring higher k values.

Figure 7(a) shows that the basic pyramid expansion (BPE) is fast in terms of cloaking time and the cloaking time does not increase significantly with the decrease in the size of grid cells. We implement the basic location anonymizer using the pyramid approach as described in [18]. Due to the fine granularity of the grid structure and consequently small cell sizes, the adaptive location anonymizer [18] does not work well due to frequent splitting and merging of cells and experiences inferior performance compared to the basic anonymizer. Except for the smallest grid size, both bottom-up dynamic expansion (BUDE) and top-down dynamic reduction (TDDR) almost match the performance of BPE. More rows (or columns) need to be added (or removed) to obtain ideal cloaking regions but maintenance of data structures is less expensive for these approaches. Interesting to note is that the actual cloaking time of all dynamic approaches is still below 1.5 ms in all cases and such low delays are hardly perceivable. Hybrid dynamic expansion reduction (HDER) performs better than both bottom-up and top-down approach, adapting appropriately to each message, by deciding whether to proceed in a bottom-up or top-down fashion.

From Figure 7(b) we observe two interesting results. First, the success 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 BPE with the three dynamic grid cloaking approaches. BPE, though marginally faster for the smallest grid cell sizes (recall Figure 7(a)), has only 41% of the messages being anonymized successfully when QoS measures are considered, while all the dynamic approaches have similar but much higher rate of success ($> 92.8\%$). All the dynamic approaches give low relative anonymity levels, which are close to one, whereas the BPE approach has about 17% 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 approaches when compared to BPE. We use grid cells of size $[24m \times 28m]$ for further evaluation.

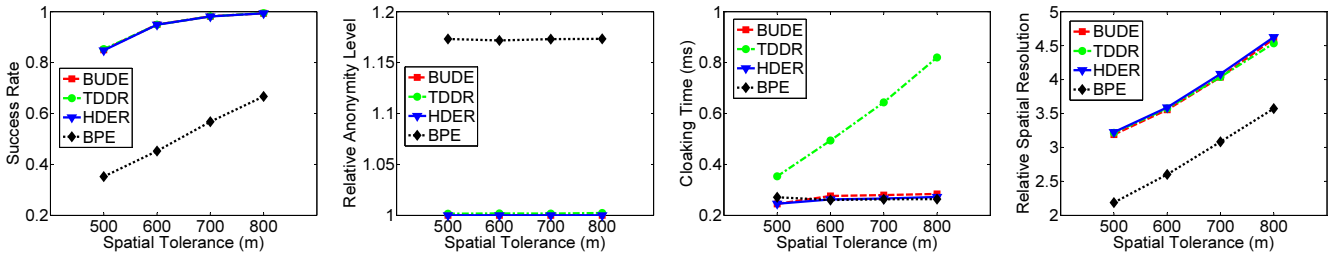
7.2.2 Varying User-defined Anonymity Level k

This set of experiments evaluates cloaking performance with varying anonymity level k for various ranges: [2–10],



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

Figure 8: Results with Varying Anonymity Levels



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

Figure 9: Results with Varying Maximum Spatial Resolution

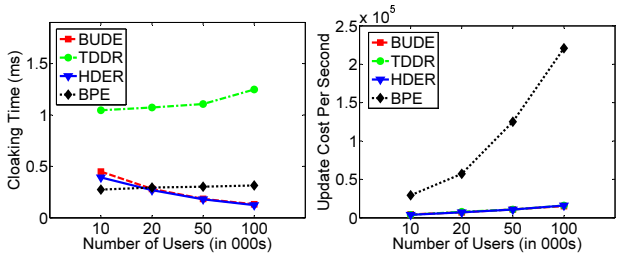
[10–50], [50–100] and [100–150]. Maximum spatial resolution values for the anonymity ranges are 400 m, 800 m, 1200 m and 1600 m (mean values with 5% standard deviation) respectively and are chosen to be large enough to theoretically allow cloaking of a large fraction of the messages. Figure 8 shows that BPE is able to cloak only around 70% of the messages with anonymity level k set in the range of [2–10] and the success rate falls further to 50-60% with increasing k values. In contrast, the dynamic approaches cloak 95-99.6% of the messages within user-defined maximum spatial resolution values (Figure 8(a)). From Figure 8(b), we see that BPE results in higher relative anonymity level but all dynamic cloaking approaches have relative anonymity levels close to one, indicating that the anonymity levels obtained for all perturbed messages are very close to the user-defined k.

Figure 8(c) shows the impact of varying k on the cloaking time of all algorithms. BPE 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 with increasing k values, the amount of increase in cloaking time for BUDE and HDER is lower when compared to TDDR. It is important to note that the cloaking time for the worst case (where the top down approach is used) is still below 3.5 ms for k values in [100–150].

Figure 8(d) displays the impact of changing k values on relative spatial resolution (RSR) obtained for the perturbed messages. Clearly, the dynamic cloaking algorithms have considerably higher RSR (25-35%) than BPE approach for all k values, though RSR values decrease as the k values become larger.

7.2.3 *Varying Maximum Spatial Resolution Values*

This set of experiments examines the performance of the algorithms by varying the maximum spatial resolution settings; 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 500 m to 800 m (mean values with 5% standard deviation) and examine the effect



(a) Anonymization Time (b) Update Costs

Figure 10: Results with Varying Number of Users

of different settings of maximum spatial resolution on the effectiveness of the different approaches. Figure 9 displays the results. The dynamic approaches are able to cloak all messages which can be theoretically cloaked for each maximum spatial resolution value, whereas BPE fails to cloak a large number of messages (50-60% less as shown in Figure 9(a)). Figure 9(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 9(c) shows that only the top-down cloaking algorithm experiences an increase in cloaking time as the maximum spatial resolution values increase, while other cloaking algorithms are not very time-sensitive to maximum spatial resolution. Finally, Figure 9(d) shows that with increasing maximum spatial resolution, the relative spatial resolution (RSR) for all cloaking algorithms will increase proportionally, with a close to constant gap between BPE and the dynamic grid algorithms.

7.2.4 *Scalability*

We now 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 algorithms to generally decrease as messages will be anonymized more easily, but the update costs for the grid-based structures will also increase. We use a similar setup to that in Section 7.2.3 with the mean spatial resolution fixed at 800 m with 5% stan-

dard deviation. We vary the number of users from 10K to 100K and observe the effect on the cloaking time and update cost. From Figure 10(a) we observe some interesting results. First, the amount of difference in cloaking time among the algorithms changes slightly with the increase in the number of mobile users. Second, TDDR shows a modest 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. However, BUDE displays a reverse trend – the cloaking time decreases as the number of users increases. This is because a higher density of mobile users per grid cell will enable it to find the smallest cloaking box faster. Finally, we observe that HDER adapts well to the increase in the number of users, offering similar performance as BUDE in terms of cloaking time. Figure 10(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 user provides a location update to the system after moving a distance of 100 m. We observe that BPE requires a large number of updates as the number of users increases. A nine-level pyramid is used in this experiment, requiring an average of 8 to 9 updates per location update request. In contrast, the dynamic cloaking approaches use the flat grid index, requiring only 2 updates for each location update request, which is significantly lower than the BPE approach [18].

7.2.5 Distribution of User Location within Cloaked Areas

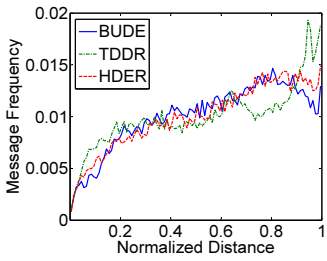


Figure 11: User Location Distribution

This experiment is designed to study the distribution of the user location within the cloaking area. Figure 11 displays the user location distribution (ULD) for the different cloaking algorithms. Plotting the ULD allows us to observe the distribution of the normalized distance from the center of cloaking area to the actual user position for each of the algorithms. The more uniform the distribution of user locations is within the cloaking areas, the harder it is for an adversary to guess the actual location of the user within the cloaking area. We observe that all dynamic cloaking algorithms provide a rather uniform distribution of user location within the cloaking area. None of the approaches reveal any significantly skewed ULD patterns.

7.2.6 Effectiveness of Spatio-Temporal Cloaking

We also examined the effect of introducing temporal cloaking in the message anonymization process. Message cloaking may be delayed according to user-specified maximum temporal resolution values. Again we use the same experimental setup as in Section 7.2.3 to measure the success rate by varying both maximum temporal resolution (or temporal tolerance) d_t from 15 seconds to 60 seconds (mean values with 5% standard deviation) and the maximum spatial resolution from 500 m to 800 m. The use of maximum temporal resolution helps increase the fraction of messages being cloaked by 10–20%; detailed results are omitted due to space constraints. In our experiments, the dynamic approaches were able to cloak 99.9% of the message anonymization requests successfully using spatio-temporal cloaking in most scenarios.

8. CONCLUSION

We described the PrivacyGrid framework which allows users to express their privacy requirements in terms of location hiding and QoS measures to control query processing overheads. Three dynamic grid-based spatial cloaking algorithms are developed for providing location k-anonymity and location l-diversity in a mobile environment. A brief discussion of the PrivacyGrid mechanisms for processing anonymous location queries is provided. We report our extensive experimental evaluation results and show that compared to existing grid cloaking approaches such as [18], our dynamic grid cloaking algorithms provide much higher anonymization success rate and yet are highly efficient in terms of both time complexity and update cost.

Acknowledgements

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.

9. REFERENCES

[1] Anonymous Web Surfing. `http://www.anonymizer.com`.
[2] Platform for Privacy Preferences (P3P) Project. `http://www.w3.org/P3P/`.
[3] U.S. Geological Survey. `http://www.usgs.gov`.
[4] C. Aggarwal. On k-Anonymity and the Curse of Dimensionality. In *VLDB*, 2005.
[5] B. Bamba and L. Liu. PrivacyGrid: Supporting Anonymous Location Queries in Mobile Environments. Technical report, Georgia Tech., 2007.
[6] R. Bayardo and R. Agrawal. Data Privacy Through Optimal k-Anonymization. In *ICDE*, 2005.
[7] A. Beresford and F. Stajano. Location Privacy in Pervasive Computing. *Pervasive Computing, IEEE*, 2003.
[8] Computer Science and Telecommunications Board. IT Roadmap to a Geospatial Future. *The National Academics Press*, 2003.
[9] M. Duckham and L. Kulik. A Formal Model of Obfuscation and Negotiation for Location Privacy. In *Pervasive*, pages 152–170, 2005.
[10] B. Gedik and L. Liu. Location Privacy in Mobile Systems: A Personalized Anonymization Model. In *ICDCS*, 2005.
[11] G. Ghinita, P. Kalnis, and S. Skiadopoulos. PRIVE: Anonymous Location-Based Queries in Distributed Mobile Systems. In *WWW*, 2007.
[12] M. Gruteser and D. Grunwald. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *MobiSys*, 2003.
[13] M. Gruteser and D. Grunwald. Enhancing Location Privacy in Wireless LAN Through Disposable Interface Identifiers: A Quantitative Analysis. *Mobile Networks and Applications*, 2005.
[14] J. Hong and J. Landay. An Architecture for Privacy-Sensitive Ubiquitous Computing. In *Mobisys*, pages 177–189, 2004.
[15] K. LeFevre, D. DeWitt, and R. Ramakrishnan. Incognito: Efficient Full-Domain K-Anonymity. In *SIGMOD*, 2005.
[16] N. Li, T. Li, and S. Venkatasubramanian. t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. In *ICDE*, 2007.
[17] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-Diversity: Privacy Beyond k-Anonymity. In *ICDE*, 2006.
[18] M. Mokbel, C. Chow, and W. Aref. The New Casper: Query Processing for Location Services without Compromising Privacy. In *VLDB*, 2006.
[19] L. Sweeney. Achieving k-Anonymity Privacy Protection Using Generalization and Suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 2002.
[20] X. Xiao and Y. Tao. Personalized Privacy Preservation. In *SIGMOD*, 2006.
[21] X. Xiao and Y. Tao. m-Invariance: Towards Privacy Preserving Re-publication of Dynamic Datasets. In *SIGMOD*, 2007.