Hindawi
Mobile Information Systems
Volume 2022, Article ID 5191041, 22 pages
https://doi.org/10.1155/2022/5191041

Research Article

Location-Query-Privacy and Safety Cloaking Schemes for Continuous Location-Based Services

Claudio Gutiérrez-Soto,[1,2] **Patricio Galdames**,[1,2] **Carlos Faúndez**,[2,3] **and Cristian Durán-Faúndez**[4]

[1]*Dept. de Sistemas de Información, Universidad del Bío-Bío, Concepción 4051381, Chile*
[2]*Group of Smart Industries and Complex Systems (gISCOM), Universidad del Bío-Bío, Concepción 4051381, Chile*
[3]*Magíster en Ciencias de la Computación, Universidad del Bío-Bío, Concepción 4051381, Chile*
[4]*Dept. de Ingeniería Eléctrica y Electrónica, Universidad del Bío-Bío, Concepción 4051381, Chile*

Correspondence should be addressed to Claudio Gutiérrez-Soto; cogutier@ubiobio.cl

Received 29 July 2021; Revised 3 February 2022; Accepted 27 March 2022; Published 17 May 2022

Academic Editor: Peter Brida

Nowadays, people can access a wide range of applications and services for mobile device users. Among them, location-based services (LBS), where the application needs the user's position to provide the service. Some examples of these applications are Uber and Waze. Nevertheless, the repetitive use of an LBS can reveal confidential user information; thus, behavior patterns—such as daily routes—could be deduced by some dishonest LBS. Furthermore, a query's keywords could provide information about a user's health status or future position when it inquires about hospitals or hotels. Therefore, an adversary can use this information for unethical purposes, and users need mechanisms that protect their privacy. At present, several approaches separately tackle location privacy, location security, and query privacy. To the best of our knowledge, no previous work deals with all these mentioned aspects simultaneously especially when users demand continuous protection when moving and accessing an LBS. This paper proposes two batch techniques to provide location privacy, location safety, and query privacy in an environment that considers a continuous LBS. These techniques apply *l*-diversity (query privacy) in a context that contemplates query semantics, as well as a diverse set of users' paths. Extensive experimentation shows that both techniques are cost-effective and scalable solutions that offer unified location privacy, query privacy, and location safety protection for many mobile users.

1. Introduction

The explosive technological development of sensors and mobile devices and their overcrowding in the markets provide users with a wide range of choices to acquire some of these devices at affordable prices. Furthermore, the processing and storage capacity of these devices has led to the development of useful applications for users, including the development and use of location-based services (LBS) [1]. Users highly demand these latter services; for example, to order food, request transportation, or meet other users, they must release their locations to the LBS server. Nevertheless, continuous delivery of our position can reveal essential information about our behavior. For instance, whether a user frequently visits a clinic or hospital, it could indicate that the user has a disease. Moreover, an adversary could use the user's behavior patterns for different purposes, including the malicious use of such information, so if a security truck regularly follows particular routes, someone can use this information to carry out a robbery (e.g., using sophisticated statistical techniques to determine the actual locations of users [2]). From this baseline, one way to forestall an adversary—who uses such information [3, 4]—is to blur our real position (i.e., this is well-known as location privacy [5–7]).

A user can hide their real position in an LBS by reporting a cloaking region. A cloaking region consists of several locations in which the user could be, ensuring that they are

present in one of them. Approaches such as in [8–18] give support to the cloaking region. Overall, these approaches deal with a trusted third party, named the anonymizer, which is responsible for choosing other locations according to the type of protection that the user is requesting. Approaches such as in [19–21] suppose that users can collaborate to provide cloaking regions. Hybrid approaches can be found in [22–24], where both the anonymizer and the user work together to build cloaking regions. Another thread is the use of the LBS anonymously, such as in [8–10, 12, 19]; according to the authors, an adversary cannot reveal users' identifications who are in the service area. In turn, other researches [3, 6, 13, 21] guarantee that each cloaking region includes locations visited by at least K different users at different points in time. Therefore, it is problematic for an adversary to identify a user at the right time, which requests a service to the LBS. By doing this, the user safeguards their location through the time dimension.

On the other hand, another branch of research is aimed at safeguarding users' physical integrity, named as location safety [5, 25, 26]. Location safety intends to avoid the physical damage that someone could receive in a specific geographical area. For instance, suppose an adversary wishes to do as much harm as possible through an attack; they should prefer to threaten densely populated areas because location safety is related to the size of the area in which users are. More precisely, location safety requires a server to satisfy a safety level Θ relating to the size of a geographic area and the number of users located there.

Another aspect that can reveal important information on behavior patterns is related to users' submitted queries to the LBS. For example, a user is looking for information about the availability of lodges or hotels in February in Villarrica city for his whole family. This request could indicate that his home could be without residents in that month. Therefore, someone unscrupulous can use this information to perpetrate a misdeed. Consequently, query privacy gains importance, especially when the query terms can divulge sensitive user information [27]. A technique used to hide a user's query is l-diversity, which hides this query by selecting and submitting to the LBS a set of l-distinct queries, one of which corresponds to the user's actual query [4, 28].

On the other side, in the literature, it is workable to find different schemes and approaches that deal with location safety [5, 25]. Meantime, in [4, 7, 28–31], query privacy is addressed. In addition, the researches presented in [2–4] and [16] deal with queries in continuous location-based service (cLBS) (i.e., a cLBS is regularly receiving users' queries, who frequently change their locations). There is a wide range of research that addresses one or two particular issues (location privacy, location safety, and query privacy), and to the best of our knowledge, no previous work deals with all the services mentioned above simultaneously in a cLBS.

In this work, we consider a traditional privacy-aware architecture (as shown in Figure 1). Here, a trusted third party called the "Location Anonymizer" must efficiently and timely build privacy and safety protections for many users accessing cLBS. This anonymizer acts as the middleware between the LBS and users, protecting users' privacy from the adversary. Keep in mind that the anonymizer must consider LBS efficiency, taking into account the simultaneous occurrence of the following requirements:

(i) To build a cloaking region required by a single user, the anonymizer must select K feasible locations to fulfill the user's requirement

(ii) Suppose now a user additionally requires location safety; in this case; the anonymizer must guarantee that a cloaking region complies with the demanded safety level. To achieve this goal, the anonymizer may need to increase the size of a cloaking region, degrading the LBS's efficiency even more

(iii) This processing at the LBS becomes even more challenging when users also demand l-diversity. Here, the anonymizer must provide $l-1$ additional distinct queries. So, in this way, the LBS has to process l queries to respond to the user's actual query

With this in mind and considering a scenario of high demand where many users frequently change their locations (i.e., cLBS), at the same time that they submit a considerable quantity of queries, counting with efficient algorithms to avoid bottleneck is a crucial factor in performance terms.

1.1. Contribution. The main contribution of this paper can be itemized as follows:

(1) Two techniques have been proposed; the first one is named Diversity Bottom-Up (DBU), while the second technique is denoted as Diversity Top-Down (DTD) to work in a cLBS where users can change their locations according to the time. Some users follow routes; meantime, other users move freely. In both cases, all users have requirements of location privacy, location safety, and l-diversity

(2) The techniques' efficiency is underlined beforehand to process users' requirements; the techniques corroborate if each user is in any restricted space. Using this rationale, it is unnecessary to compute users' requirements when they are not in a restricted space. A restricted space can be seen as a geographical area where an adversary could determine the exact location

(3) A new notion of geographical semantic is proposed. This notion is based on ontologies, which contain specific places related (i.e., place names are in the ontology) to physical distance. We assume that an essential proportion of queries submitted by users is related to the geographical zone where the user is

(4) Aiming to tackle the l-diversity, we consider that sensitive terms exist (which can be previously determined). In this manner, before computing l-diversity for a user, our techniques decide whether it is necessary to calculate l-diversity. If the query comprises

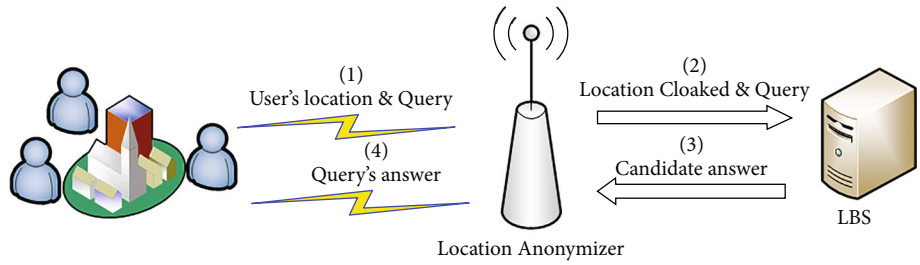


Figure 1: Traditional privacy-aware LBS architecture.

terms that do not provide confidential information, then the *l*-diversity is omitted

(5) Extensive experimentation using simulations has been carried out to evaluate the techniques, considering different metrics such as cloaking regions, computational saving, entropy, and size of cloaking regions. In simulations, we have incorporated the concept of the timestamp (i.e., a simple timestamp can be seen as a picture of objects distributed in the space for a particular point in time). Therefore, the time in which users are in motion can be modeled as a set of timestamps

This paper is an extended version of a work-in-progress article presented at CODASSCA 2020 [32]. The remainder of this paper is organized as follows: In Section 2, the related work is presented. In Section 3, the methodological description is exposed. In Section 4, both techniques DBU and DTD are presented. Section 5 exposes the security analysis for both techniques. In Section 6, the experimental environment and the empirical results are given. Finally, Section 7 provides some perspectives on future work along with the conclusions.

2. Related Work

Aiming to organize the related work, we have considered those approaches that involve more than one service or requirement.

In [4], a novel query-perturbation-based scheme that protects query privacy in continuous LBS is presented. The scheme named DUMMY-Q provides dummy queries, which consider query context and user motion models. According to the researchers, this scheme does not need the existence of a trusted third party. Aiming to reduce computational capacity requirements and storage, DUMMY-Q uses a quadtree. In order to consider the context, historical query logs are used at the same time the road density is utilized to carry out simulations. Specifically, the road density information was obtained from the second edition of the Topologically Integrated Geographic Encoding and Referencing (TIGER) 2 system published by the US Census Bureau in 2006. The service attribute values were obtained using Uniform and Zipf distribution, while Brinkhoff's Generator gives the snapshots. Two performance metrics were evaluated both for privacy protection: Query Success Rate (QSR), which deals with the number of query dummies, and the Average Size of the Candidate Set that captures the average number of service attributes. Empirical results assert that the problem of ensuring both utility and privacy for dummy insertion is NP-complete; in the meantime, their approach based on heuristics can provide the dummy queries.

In [15], with the aim to enhance the response times of queries without decreasing the degree of privacy protection, an online method that relies on privacy region replacement (PRR) is proposed. To achieve this goal, a privacy region is built at the beginning, taking into consideration the people density and privacy requirements. Thereafter, the privacy region is changed to an anonymous region grounded on people distribution in the privacy region. Subsequently, the coverage degree between the user query and the anonymous region is determined, so the new query region is employed to submit the online query. Simulations were used to evaluate people's densities, while the dataset employed was Thomas Brinkhoff (network-based Generator of Moving Objects). The evaluated metrics correspond to anonymous degree K, query response time, and people density. Empirical results are compared to Casper, Fragment, and ARC. The final results indicate a reduction of at least 17.33% for PRR in comparison with the other approaches.

In [33], a Cache-Based Privacy-Preserving (CBPP) is exposed. From the researchers' point of view, the use of this approach diminishes the probability of knowing the real position of users because only in some cases are the real positions provided to the LBS. This approach deals with both location privacy and query privacy while avoiding the issue of a trusted third-party (TTP) server through the collaboration among users in a mobile peer-to-peer (P2P) environment. In this approach, each user holds a buffer in its mobile device to store service data, which works as a micro-TTP server. Roughly, the user only interacts with the LBS when its neighbors do not have the data that it is looking for. Accordingly, less interaction among users and LBS implies less risk to the users' privacy. In the experiments, 1000 user mobiles were spread out in a P2P network. To provide query privacy, the *l*-diversity is used. The evaluated metrics comprise the effects of *l*-diversity on cache use, also considering the time. CBPP is contrasted with the schemes DLS, Mobicache, and MobiCrowd. Empirical results show better results for CBPP regarding other schemes.

In [34], a dummy generation scheme, which contemplates the hierarchical structure of the address (DGS-HSA),

is described. DGS-HSA provides a novel meshing method splitting the historical location dataset taking into consideration the administrative region division. In an attempt to protect the user's location, query privacy, and organization information, dummy selections took place from the historical location dataset considering a two-level grid structure. Aiming to adjust the privacy protection level and system overhead, a multiobjective optimization problem is solved to prove the scheme's convenience. This scheme involves sending continuous queries to the LBS by the user on-site. To this end, K locations from historical locations are chosen as dummies. DGS-HSA works like this: firstly, DGS-HSA splits a city area where users are at different grids with two-level structures. In that regard, each grid exemplifies an organization, and all historical locations are gathered into these grids according to their organizations. Regarding the experimental environment, the Geolife Trajectories 1.3 dataset and the POI dataset of AMAP were used. Studied metrics are related to the balance between privacy protection and system overhead. Despite the fact to reach a good balance, the authors claim that the final results depend fundamentally if the historical location dataset considers enough areas as scarcely populated or inaccessible areas.

A theoretical model for location obfuscation is presented in [16]; this model allows the specification of different obfuscation levels. To clarify the proposed model's effectiveness, the authors analyze a popular square grid-based obfuscation mechanism, focusing mainly on obfuscation-based privacy-preserving mechanisms. To model obfuscation, two approaches, local and global, are presented. Broadly speaking, the difference between both approaches is given by the use or the nonuse of other users' actual locations. Considering a scenario where continuous queries take place, the local approach is expressed using a formal model. Aiming to quantify the privacy of obfuscation, a probability is defined considering the square grid-based location. According to the authors, an advantage of techniques based on obfuscation is that the user's real location is mapped to an inaccurate location, and therefore, it is not necessary to modify the original LBS architecture. Besides, the researchers point out that their theoretical results can be extended to other privacy-preserving techniques.

In summary, all works mentioned above deal mainly with two services, location privacy and query privacy, considering a context where users are in constant motion. Some simulations are based on city maps; nevertheless, just one approach uses different distribution probabilities to cover a broader spectrum of empirical conclusions.

Unlike previous work, in [5], the authors propose two scalable algorithms that deal with two services for a sporadic LBS: location privacy and location safety. Towards that end, the algorithms consider users who are close among them, processing their requirements in batch. The algorithms are named bottom-up and top-down according to their form to work. The bottom-up algorithm firstly seeks a small candidate cloaking region, which fulfills users' service requirements. By contrast, the top-down algorithm supposes that the entire network is the initial candidate while the cloaking region is pruned when the users' requirements are reached. Extensive experiments evaluated four metrics, computational cost, size of a cloaking region, number of cloaking regions built, and entropy of a cloaking region. The experiments were carried out using simulation considering Zipf, exponential, and uniform distribution to spread out users in the region. The bottom-up algorithm displays a good balance quality of a cloaking region and its size. Otherwise, the top-down algorithm provides advisable results when the quality and the number of built cloaking regions are considered despite the high computational cost. Incidentally, note that this work does not contemplate a cLBS, where users are in motion.

To sum up, none of the aforementioned works address more than two services. Therefore, it is not feasible to evaluate how different services affect LBS performance directly. In addition, except for [5], few initial configurations of users in the network are studied. Taking into consideration those mentioned previously, in this research, we consider three services—over a cLBS—using different initial configurations of users in the network in an environment where users are constantly in motion.

3. Methodology

3.1. Basic Concepts. When a user needs to contact an LBS server, it must submit a location-based query, including the user's exact location and several terms that describe its query. Without loss of generality, we assume a location-based query, or simply, a query is a range query anchored to the user's exact location and demands information about all points of interest (POI) matching the query terms. For instance, a user may be interested in discovering all restaurants or dining places located nearby or all health centers located nearby as well. Since the location and the terms declaring the type of POI searched can compromise a user's privacy (i.e., location privacy and query privacy) and safety (i.e., location safety), we aim to protect them.

In order to provide location privacy for a user, we use the traditional scheme based on K-anonymity (i.e., the adversary cannot distinguish the real user's location from other $K-1$ locations submitted to the LBS [35]). Here, we define a user's cloaking region (CR) as a geographic area that includes their real location (i.e., exact location), and any other position is equally likely to be the real one.

Besides, to protect a user's query, we use the geographical l -diversity notion. This concept is aimed at protecting query privacy by selecting other $l-1$ distinct queries that are as popular as the actual user's query in a specific location [28]. When a user demands l-diversity and K-anonymity simultaneously, the anonymizer must select $K-1$ distinct locations and $l-1$ different queries regarding the query submitted by the user.

Aiming to determine whether two queries are similar or not, we assume a set of ontologies. In this way, an LBS contains different ontologies to classify queries. To illustrate this, consider a user that has just arrived at a new city and submits several travel queries to the LBS [36]. For instance, a category (i.e., an ontology) can be queries related to lodging service, whereby this category should be confirmed by

terms such as hotel, hostel, and bed-and-breakfast to mention a few. In turn, another query category could have terms related to dining queries, by which terms such as restaurants, cafeterias, or bars should be part of this category. It should be noted that when the l-diversity is applied, the sensitive terms of the submitted query (i.e., these terms can provide relevant information to the adversary) are changed by terms of other categories to hide the original semantic of the submitted query.

Finally, we will take into account the concept of location safety. This notion implies "identifying an area whose safety level is below some safety threshold (θ)." Here, the safety level corresponds to "the ratio of its area, and the number of nodes inside it" [25], and therefore, the safety threshold corresponds to the minimum safety level demanded by a user.

Our goal is to build CRs on-demand, considering the location privacy, location query, and query privacy needed by many mobile users during their journey. Since this task can overwhelm an anonymizer, we want to efficiently and timely build many CRs for many users but only when this protection is beneficial. Yang and Cai [7] show that some types of locations need cloaking protection but not others. For instance, if a user discloses their location within a residential or a workplace place, this side information can help the LBS conclude a customer's identity. However, other locations like highways do not leak any detail about a user. The authors call the former location as restricted spaces, and based on this concept, they propose a technique called restricted space cloaking to limit the number of cloaking regions that an anonymizer has to build for a mobile user. In this work, we also limit the usage of the l-diversity, only when the terms used to specify a query can release private details about its owner.

In the cLBS context, we assume that there are two types of adversaries, a passive and an active. A passive adversary is any user that can monitor and eavesdrop on the wireless traffic or compromise any other user to obtain their private data. An active adversary is any user who can perform an attack on the LBS server to obtain its customers' sensitive data. The LBS can be seen as a potential adversary in this latter definition. The LBS can use the collected information from its customers to infer more private details about them, like their behavioral patterns and locations. Finally, we assume these adversaries can perform the following attacks: inference attack, colluding attack, and accessibility attack.

Given two consecutive cloaking regions CR_1 and CR_2 built for a known user at time t_1 and t_2, where $t_1 < t_2$, we define the following:

(i) *Inference attack*: this attack is successful when a single adversary can narrow down the location of the user by collecting and correlating several locations reported by this user with known public information. Similarly, we say this attack is successful when an adversary can conclude which queries were issued by some user in some period.

(ii) *Colluding attack*: this attack is successful when many adversaries collaborate and exchange locations and queries from users to perform inference attacks.

(iii) *Accessibility attack*: this attack can be successful in two cases. The first case arises when the adversary finds out that there is no path in the service area to travel from CR_1 to a particular point in CR_2. The second case can arise even though such a path exists, but it is not feasible to travel from CR_1 to some point in CR_2 in a period t_2–t_1. Here, we assume the adversary knows an upper bound of the speed of the users.

3.2. System Overview. We assume our system architecture ($\mathbb{S}$) uses a single anonymity server to manage location privacy, query privacy, and location safety requirements demanded by many mobile users, as shown in Figure 1. Thus, a user $u(C_u, t)$ is determined by a spatial location C_u (it can be represented by the points X_u and Y_u or latitude and longitude) and a point in time t. Following this reasoning, a timestamp τ_i corresponds to the same point in time t_i for all users of $\mathbb{S}$ (i.e., $\tau_i = \{u_j(C_{u_j}, t_i), \quad \forall u_j \in \mathbb{S}\}$). In this way, users' motions through time can be represented by a sequence of timestamps $\tau_1, \cdots, \tau_n$, such as $\tau_i < \tau_{i+1}$. In order to efficiently process each request for privacy and safety protection, the entire network area is partitioned into a set of $n \times n$ disjoint cells of equal size as shown in Figure 2. While a user u is moving, it can submit anonymizer protection requests for location and query anonymization. These requests consist of the user's current and exact location represented as a 2D point$(X_u ; Y_u)$, its location-based query, its demanded K_u -anonymity for location privacy, (l_u)-diversity of query privacy, and its desired location safety (θ_u).

The anonymizer enqueues all anonymization requests in a waiting list denoted as U. Finally, our system returns for each user u in U a cloaking region denoted as CR_u, consisting of a set of selected cells that conform to each individual's privacy and safety requirements.

To protect location privacy, we build a cloaking region for location privacy following a similar approach stated in several articles [5, 6, 20]. Our system chooses as a cloaking region a set of at least K_u cells that maximizes the entropy. To do so, when users report their current location-based queries, the anonymizer maintains a query frequency table that counts how frequently a query (or type of query) comes from a given cell in a similar way as proposed by [3]. Here, we assume that the anonymizer classifies each query in only one of many distinct classes or types according to their semantic similarity (we will explain this issue in the next section). Our server will keep a data structure for each cell to estimate the probability of issuing a particular type of query from a cell. Specifically, we say and define the query probability as

$$q_i = \frac{\text{Number of times a given (type)query is originated from cell } i}{\text{Number (\#) of requests coming from the network area}}, \tag{1}$$

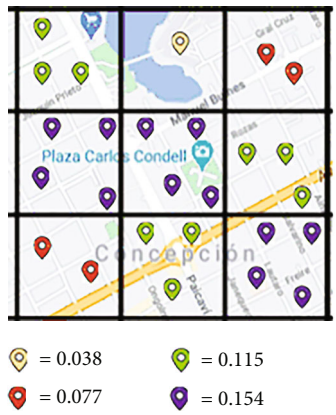


Figure 2: Grid partition of the network area and query frequencies per grid.

where $\sum_{i=1}^{n^2} q_i = q_1 + \cdots + q_n + \cdots + q_{n^2} = 1$, for all $i = 1, .., n^2$.

As a user u located in cell C_u demands query privacy, our system needs to take into account as well their query diversity (l_u), and therefore, it must select at least $l_u - 1$ different queries whose query probability are similar to the real query.

When a user also demands K-anonymity, our system selects other $K - 1$ cells. To select those cells, we introduce the notion of entropy of a cloaking region CR, denoted as $H(\mathrm{CR})$, which is computed as

$$H(\mathrm{CR}) = -\sum_{K}^{j=1} p_j \log_2\left(p_j\right), \tag{2}$$

where p_j represents the normalized probability of requesting the real user's query from each specific cell $c'_j \in CR$. This latter probability is computed as $p_j = q_j / \sum_{l=1}^{K} q_l$. The higher the entropy of a CR is, the better the location privacy protection offered.

On the other hand, a novelty aspect considered in this paper corresponds to the geographical semantic notion. Since many applications use georeferenced data, we assume that many queries are related to users' geographical areas. By these means, places' ontologies associated with nearby spaces are used. For instance, an ontology could be related to the downtown of Concepción city, where it can contain names (i.e., it can be seen as terms) of banks, restaurants, hotels, drugstores, and so on (i.e., indeed, an ontology can be seen as a topic or type). Following this trend, a whole space should be composed of several ontologies. Note that each cell (see Figure 2) has an associated probability of submitting a query to the LBS in our architecture. Thus, ontologies are essential in building the queries and carrying out the l-diversity. Therefore, a query is formed by several terms, mainly belonging to the ontology associated with its geographical space and other terms related to other ontologies.

To achieve this goal, different probability distributions are used to choose the specific ontology firstly and, subsequently, the particular term. In so doing, the query will be formed mainly by terms of its ontology when the distribution probability is not uniform as the case of exponential and Zipf distribution (i.e., from the semantic point of view, when most of the terms belong to an ontology, we could say they belong to a semantic type).

Once the ontology has been chosen, the uniform distribution is utilized to select a term. Keep in mind that a query can have between 5 and 8 terms. It means that a query can have terms from different ontologies or types, such as occurs in the real world. Recall that a query could have sensitive terms, which eventually could be used by an adversary. Thus, when the l-diversity must be computed, first, the identification of sensitive terms and the semantic type of query must be determined. Later on, a similar cell regarding the probability of submitting the query should be found. Besides, this similar cell should be a different semantic type. It should be noted that each query is associated with a specific cell in our approach. Hence, to carry out l-diversity, the substitute query should have at least the same quantity of sensitive terms, prioritizing the same quantity of terms (i.e., sensitive and nonsensitive terms).

To facilitate our approach's presentation, let us consider the example shown in Figure 3. Here, user u, located at cell 1, needs to find out all nearby health centers. In this figure, we can see that there are four types of queries as indicated by the legend. Since the query considers a sensitive term, assume u demands 3-diversity, and 2-anonymity. First, the anonymizer chooses a 3-diversity set that groups queries of similar frequency. We can observe the query frequency table for cell 1; the anonymizer can select from several alternatives. A first option can be the set as hospital, hotel, and restaurant, but another set can be hospital, hotel, and ATM. The anonymizer chooses randomly a set whose frequencies for the other queries is closer to the real query, i.e., hospital. Now, let us assume the anonymizer chooses as a diversity set: hospital, hotel, and restaurant.

Now, the anonymizer searches for another cell to build a 2-anonymity set. Here, it chooses the cell that makes the entropy of the anonymity set the highest one. In this example, the anonymizer selects those cells with a similar query frequency about hospitals, which are cells 2 and 4, but not cell 3 since it shows a smaller query frequency for hospitals and therefore a smaller entropy.

However, the anonymizer must also take into account the chosen 3-diversity set. In this case, it chooses cell 4 since this cell has a query frequency for restaurant closer than the one counted for cell 2.

In case the same user demands protection of its location safety, we follow a similar approach, as shown by Xu and Cai [26]. These authors define the concept of the safety level of a cloaking region CR as $\mathrm{SL}(\mathrm{CR}) = A(\mathrm{CR})/N(\mathrm{CR})$, where $A(\mathrm{CR})$ denotes the area of a CR and $N(\mathrm{CR})$ denotes the population of CR (i.e., the number of wireless users moving within region CR).

Thus, given a user u located within a CR and demanding a location safety requirement θ_u, then CR protects the location safety of the user u if $\mathrm{SL}(\mathrm{CR}) \geq \theta_u$.

As a general remark, Xu and Cai [25] assume that a CR is a convex region, which is not our case since our CR is a set of fragmented areas or separate cells. Now having this

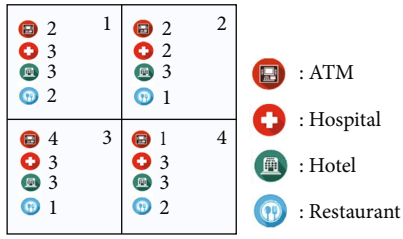


FIGURE 3: Example of our basic idea.

description in mind, we define a *cloaking region* for a user u, CR_u, as a set of K-disjoint cells (c_i) of the network area, and we propose to compute a safety level of CR as follows:

$$\mathrm{SL}(\mathrm{CR}) = \frac{\sum_{i=1}^{K} A_i}{\sum_{i\_1}^{K} \#\text{of wireless users moving within } c_i}. \quad (3)$$

Since all cells have the same area (A), we can simplify equation (3) as $A/(1/K)\sum_{i=1}^{K}\#\text{of users in } c_i$. The higher the safety level of a CR is, the better its location safety protection.

4. Proposed Batching Techniques

Here, we briefly describe the notation used to describe our proposed algorithms:

(i) Let $\mathbb{C}_{\mathbb{N}}$ be the set of all cells in the network area sorted in ascending order according to their probabilities

(ii) Let U be the current set of users requesting location privacy and safety protection. Then, given a user u in U, CR_u is the user u's cloaking region

(iii) Given a user u, K_u corresponds to its location privacy, while θ_u determines its location safety protection

(iv) Let $\#(\mathrm{CR})$ be the cardinality (size) of a cloaking region CR defined as the number of cells c'_j making up this region

(v) Let $\mathbb{C}(u, q_i, r)$ be a subset of $\mathbb{C}_{\mathbb{N}}$, which consists of those "r" neighbor cells at the right and at the left of C_u in $\mathbb{C}_{\mathbb{N}}$. These cells have similar probabilities regarding q_i. Thus $\#(\mathbb{C}(u, q_i, r)) = 2 \times r + 1$

(vi) Given a cell C_i, the occupancy of C_i ($O_c(C_i)$) corresponds to the number of mobile users inside C_i.

(vii) Let $\theta_{\max}$ be the maximum location safety requirement a user can demand

(viii) Let Q be the set of semantic queries that can be submitted to the LBS within the service area. Then, the size of Q is denoted as $|Q|$

(ix) Let l_u be the diversity value demanded by user u. Then, L_u corresponds to a set of l_u-semantically distinct queries, including u's query

(x) Let p_{ij} be the probability associated with the cell j, from which a user in C_j can submit a query i

The Algorithms 1 and 2 are aimed at finding a set of dummy queries having similar probabilities to the user's actual query q_j. These algorithms use Q, such that there are subsets that have similar semantic queries, but each subset is different regarding others. Specifically, Algorithm 1 selects $l_u - 1$ queries whose query probabilities are similar to those of q_j. Here, the algorithm computes for each query q_i ($i : 1, .., |Q|$) the ratio between p_i and p_j. The anonymizer then chooses the $l_u - 1$ distinct semantic queries at random with a probability proportional to its corresponding ratio.

Given L_u and the user's query q_u, Algorithm 2 finds all cells (C_k) that have similar l-diversity to C_u. Then, C_k and C_u have similar l-diversity regarding a set L_u if cells' probabilities C_k and C_u (note that the probability of a cell is associated with the likelihood of submitting a query) are similar considering the following notion of a norm:

$$\mathrm{Norm}(C_k, C_u)_{L_u} = \max_{L_u}\left\{\left|p_i^k - p_i^u\right|\right\}, \quad \forall q_i \in L_u. \quad (4)$$

Here, we consider that p_i^k is the query probability for the query i in L_u in cell k, and p_i^u is the query probability for the query i in L_u in user's cell C_u. Then, cell C_k is chosen as a candidate cell only if the value for equation (4) is smaller than a system threshold, denoted as s.

We developed two batching techniques based on the above algorithms to compute efficiently cloaking regions for many mobile users. The first one (DBU) follows a bottom-up approach since it first finds a small candidate cloaking region satisfying a location and query privacy requirements. Then, it tries to enlarge this region until it achieves the location safety requirement. Conversely, the second technique (DTD) follows a top-down approach assuming that the entire network is the initial candidate for a cloaking region, and then, it attempts to reduce its size. For both techniques, users' requirements, along with the prevention of accessibility attacks, are all constantly checked. A common characteristic in both techniques is that a cloaking region comprises similar users' requirements, specifically similar location privacy, location query, and safety.

4.1. Diversity Bottom-Up Technique. This bottom-up approach is based on Algorithm 3. Given a user u, this algorithm is aimed at computing a cloaking region CR_u. In line 3, a candidate set S of cells offering similar l-diversity as the user's cell (C_u) is chosen. Then, it finds a candidate cloaking set, composed of K_u cells having the highest entropy (lines 8-9). Lines 10 and 1 check whether this candidate cloaking region is susceptible to being compromised by an accessibility attack, assuming the knowledge of the user's current cloaking region ($\mathrm{CR}_{\max}$). Finally (lines 15 and 16), it

Data: user u located at cell C_u and having an actual query q_j
Results: a l-diversity set (L_C) for user u
1 $i \longleftarrow 0$;
2 $R_i \longleftarrow \varnothing$;
3 **for** ($i < |Q|$ and $i \neq j$) **do**
4 $\quad R_i \longleftarrow \min_{\text{cell}C_u}\{(p_i/p_j), (p_j/p_i)\}$;
5 $\quad i \longleftarrow i + 1$;
6 **end**
7 i $\longleftarrow 0$;
8 $L_C \longleftarrow \{q_j\}$;
9 **while** $i < l$ **do**
10 $\quad$ From Q chooses a query q_i with probability proportional to R_i;
11 $\quad L_C \longleftarrow L \cup \{q_i\}$;
12 $\quad$ i $\longleftarrow$ i + 1;
13 **end**
14 Return L_C

Algorithm 1: Computing a l-diversity set (L_C) for a user located at cell C_u and having an actual query q_j.

Data: l-diversity set L_u, actual query q_u, user's cell C_u, anonymity degree K
Results: a candidate set of cells (S) for l diversity and K-anonymity
1 $i \longleftarrow 0$;
2 $S \longleftarrow \mathbb{C}(u, q_u, r)$;
3 **for** each $c' \in \mathbb{C}(u, q_u, r)$ and $c' \neq C_u$ and $|S| \geq K$ **do**
4 $\quad$ **if** norm $(c', C_u)_{L_u} \geq s$ (see equation (4)) **then**
5 $\quad\quad$ Remove c from S;
6 $\quad$ **end**
7 **end**
8 Return S

Algorithm 2: A candidate set of cells satisfying l-diversity and K-anonymity for a user located at cell C_u and asking query q_u.

Data: user u, q_j, m, s, $\text{CR}_{u,\text{old}}$
Results: a new cloaking region (CR_u) for user u satisfying l_u for a query q_j
1 $i \longleftarrow 0$;
2 $L_u \longleftarrow$ call Algorithm 1 (q_j, C_u, l_u);
3 $S \longleftarrow$ call Algorithm 2 (L_u, C_u, K_u, $4K_u$);
4 **if** $\text{CR}_{\max}$ is null **then**
5 $\quad \text{CR}_{\max} \longleftarrow C_u$
6 **end**
7 **for** $i < m$ **do**
8 $\quad \text{CR}' \longleftarrow K_u$ cells at random with equal probability from S;
9 $\quad \text{CR}_{\text{aux}} \longleftarrow \text{CR}'$ only if CR' has the highest entropy computed with respect to L_u;
10 $\quad$ **if** there exist feasible paths from $\text{CR}_{\max}$ to CR_{aux} and $E(\text{CR}_{\text{aux}}) > E(\text{CR}_{\max})$ **then**
11 $\quad\quad \text{CR}_{\max} \longleftarrow \text{CR}_{\text{aux}}$;
12 $\quad$ **end**
13 $\quad i \longleftarrow i + 1$;
14 **end**
15 $\text{CR}_u \longleftarrow \text{CR}_{\max}$;
16 Return CR_u, L_u;

Algorithm 3: Computing cloaking region for a user u.

returns the chosen set CR_{max} as the new cloaking region for user u.

Algorithm 4 is the proper batching procedure whose goal is to build in batch several cloaking regions for all users in U (those users demanding privacy and safety protection). The idea is to build a candidate CR for the user having the largest location privacy requirement (u_v). The anonymizer verifies if CR_v needs to be increased to satisfy user u_v's location safety requirement.

Specifically, Algorithm 4 chooses first the user v demanding the highest K-anonymity (K_v, line 2). Then, it calls Algorithm 3 to obtain a candidate CR_v that also satisfies its l-diversity. Now, it verifies whether this CR_v satisfies the safety level (location safety) demanded by user v (line 5). If this is not the case, our algorithm randomly chooses a cell having a similar l-diversity but a low occupancy (lines 6-7), and then, it verifies whether this cell is reachable from the current user u's CR (line 8). When this latter case is true, our algorithm adds the chosen cell into CR_v (line 9).

When the safety level is achieved (line 12), our algorithm finds out what other users (labeled as user u) may share CR_v (line 13). Here, three restrictions need to be all satisfied (lines 14-15). The first one prevents the accessibility attack by finding at least a feasible path from any cell in user u's CR to any other cell in this new candidate CR_v. The second one checks whether the K-anonymity degree demanded by user u is also satisfied by the candidate CR_v. Finally, the safety level demanded by the user u is also checked. Note that the selected user u must have similar l-diversity as user v (line 13). Finally, the process is finished when U becomes empty or $CR_v = \mathbb{C}_N$ (lines 20-21).

4.2. Diversity Top-Down Technique. DTD (see Algorithm 5) is aimed at computing an initial CR for a chosen user and checking whether other users can share it as well. To do so, it selects any user v in U (line 3) having the largest θ, denoted as θ_v. In line 4, it filters out any cell not having a feasible path to either CR_v or C_v. Then, in line 6, it chooses a candidate CR_v satisfying the demanded diversity set L_v (computed in line 5).

From now on (lines 7 to 22), it tries to reduce the size of CR_v (lines 11-17) as long as the cardinality of $CR_v \geq K_v$ and $SL(CR_v) \geq \theta_v$ (lines 22 and 12). To do that, it finds out whether removing a randomly chosen cell c (line 12) from CR_v could achieve the lowest reduction of the entropy of CR_v (line 14). After "m attempts" (line 10), only one cell is chosen and removed from CR_v (lines 15 and 20). Note that the state variable E_{max} becomes nonzero (line 19) only when lines 12 and 13 are satisfied. Thus, the anonymizer removes cell c' from CR_v (line 15). In lines 23-27, this technique verifies whether another user can share the same cloaking region CR_v by checking K-anonymity, l-diversity, location safety, and feasibility of having an accessibility attack. Finally, the algorithm stops when the first repeat-end statement finishes (lines 2 and 29). This latter happens when all pending requests have been successfully attended.

4.3. Time Complexities. This section presents the time complexities for all algorithms implied in our techniques.

4.3.1. Time Complexity for Algorithms 1 and 2. As stated previously, Algorithm 1 has as purpose of computing the l-diversity. Consequently, it returns a similar semantic query's set ($l-1$ queries from the set). The time complexity to provide the l-diversity for a user comprises $O(|Q|)$ (lines 3-6) and $O(\mathrm{lR})$ (lines 9-13). Notably, $|R| = |Q| - \{q_j\}$, by which the latter can be seen as $O(l|Q|)$. Finally, the algorithm implies $O(l|Q|)$ for a single user.

Regarding the second algorithm, it provides both l-diversity and K-anonymity. To achieve this goal, a subset of cells $\mathbb{C}_N$ with similar probabilities to the q_i's cell is used. The algorithm's invariant simultaneously depends on l, K, and s. Thus, the time complexity corresponds $O((\max(l, K, s)))$; however, s is much lower than K and l.

4.3.2. Time Complexity of Algorithm 3. Algorithm 3 provides the CRs; for this purpose, Algorithms 1 and 2 are used (lines 2-3). The highest entropy takes $O(m^2)$ (lines 7-9), in such a way that there are m candidate sets from which the set with the highest entropy is chosen. Following the same trend presented in [20], the candidate cells for each set are extracted from a list of cells, which contains $n \times n$ cells (note that it corresponds to our network domain and that these cells are sorted according to their probabilities, using Merge Sort). To this end, $4K_u$ (line 3) cells are selected from the list (e.g., if C_u is at the middle of the list, $2K_u$ can be chosen from the left side and $2K_u$ from the right side regarding C_u; in particular, $m < 4K_u$). Checking the existence of a path from CR_{max} to a cell (lines 10-12) takes $O(M)$, which in the worst case implies checking 7 cells (M) (it occurs when C_u is in the middle of its neighboring cells). At all events, M is a constant, which can be omitted. Finally, the time complexity for this algorithm is $O(\max(\text{Algorithm 1}, \text{Algorithm 2}, m^2 + M)) = O(\max(l|Q|, (\max(l, K, s))))$, whether $m^2 <$ Algorithm 1 and $m^2 <$ Algorithm 2.

4.3.3. Time Complexity of DBU Algorithm. The DBU algorithm gives a set of CRs for each user, which comply with their requirements. Towards that goal, both Algorithms 2 and 3 are used. Without loss of generality, in the DBU's worst-case, DBU takes $O(|U|(\text{Algorithm 3 Algorithm 2 } M))$ (lines 2, 3, and 6). Following the same trend as Algorithm 3, checking a path's existence implies corroborating M cells (lines 8-10 and 14-16), which is a constant that can be discarded. Note that the loop in line 13 implies checking all users in CR_v, such that there is at least one $l_u \leq l_v$. Depending on the user distribution over the network domain, the number of users CR_v should be lower than $|U|$. If the number of users is $|U|$, this loop takes $O(\text{Algorithm 2})$. In this way, the time complexity for BDU is $O(|U|(\max(l|Q|, (\max(l, K, s)))(\max(l, K, s))))$.

4.3.4. Time Complexity of DTD Algorithm. The DTD algorithm provides a set of CRs for all users in U. Note that Algorithms 1 and 2 are employed (lines 5 and 6). Subsequently, the highest entropy is calculated for each user taking into consideration its K requirement. As a consequence, this procedure takes $O(\max(CR_v, m^2))$ (lines 7-22), such that CR_v is the maximum value considering all

Data: set U
Results: a set of cloaking regions for every user u in U satisfying its respective K_u and θ_u

1 **repeat**
2 $v \longleftarrow$ chooses a user with the highest K from U (denoted as K_v). If there are many of them, choose one with the highest θ in U
3 $\mathrm{CR}_v, L_v \longleftarrow$ call Algorithm 3 (v, q_v, $2K_v + 1$, CR_v);
4 **repeat**
5 **if** $\mathrm{SL}(\mathrm{CR}_v) \geq \theta_v$ **then**
6 $S_u \longleftarrow$ call Algorithm 2 (L_v, C_u, K_u, s);
7 $C'_u \longleftarrow$ from $S_u \setminus CR_v$ with a probability inversely $\propto$ distance(C_v, C_u) $\times$ $O_C(C_u)$ (occupancy);
8 **if** there exist feasible paths from any cell in CR_u to cell C'_u **then**
9 $\mathrm{CR}_v \longleftarrow \mathrm{CR}_v \cup \{C'_u\}$;
10 **end**
11 **end**
12 **else**
13 **for** any user u located in CR_v having a query type $q_u \in L_v$ and $l_u \leq l_v$ **do**
14 **if** there exist feasible paths from CR_u to CR_v **then**
15 $\mathrm{CR}_u \longleftarrow \mathrm{CR}_v$ if $K_v - \Delta \leq K_u \leq K_v$ and $\theta_u \leq \theta_v$;
16 Remove u from U only if CR_u was set as CR_v;
17 **end**
18 **end**
19 **end**
20 **until** $\mathrm{SL}(\mathrm{CR}_v) < \theta_v$;
21 **until** $U = \varnothing$ or $\mathrm{CR}_v = \mathbb{C}_N$;

Algorithm 4: Diversity bottom-up cloaking batching algorithm (DBU).

users that belong to U. In this way, the time complexity between lines 3 and 22 implies $O(|U|(\text{Algorithm 1} + \text{Algorithm 2} + \max(\mathrm{CR}_v, m^2))$. Note that the last cycle (lines 23-28) is similar to the last cycle in DBU, whereby it takes $O((\text{Algorithm 2})$. Finally, the time complexity for DTD is $O(|U|(\text{Algorithm 1} + \text{Algorithm 2} + \max(\mathrm{CR}_v, m^2))($ Algorithm 2)). Therefore, it can be seen as

$$O\left(|U|\left(l|Q|(\max(l, K, s)) + \max\left(CR_v, m^2\right)\right)((\max(l, K, s)))\right). \tag{5}$$

To sum up, DTD's time complexity is more expensive than DBU (see Table 1). An important point to note is that the time complexity of Algorithm 1 depends on the number of queries, which can be high. However, several strategies could be implemented with the aim of improving the efficiency of algorithms; between them, we avoid the increase in the number of queries using some mechanism of caching or the reduction of the K and m parameters in those algorithms that use them.

5. Security Analysis

Section 2 compares our techniques with several previous works. It is clearly shown that those works do not protect the privacy of the user's location, the physical security, and the privacy of the queries simultaneously for a continuous LBS, while our techniques (DBU and DTD) tackle at the same time those requirements.

5.1. Resistance to Collusion Attack. Adversaries can collude with other users to obtain additional information from legitimate users.

Theorem 1. *Our scheme is resistant to collusion attacks.*

Proof. The collusion attack can be carried out under a model of honest but curious users. Here, it is assumed that users can collaborate to broaden their knowledge and improve the likelihood of learning more information from other legitimate users. In our scheme, users release their sensitive information only to our anonymizer and in no case is the information of each user disclosed to other users. Therefore, our schemes are resistant to collusive attacks.

In an extreme case, a passive adversary could collude with the LBS server or directly compromise it to obtain all the users' information. In this scenario, this attacker becomes an active adversary who could perform an inference attack.

5.2. Resistance to Inference Attacks

Theorem 2. *Our schemes are resistant to inference attacks.*

Proof. In this case, it is assumed that an active adversary knows precisely the details of our schemes, as well as the historical information collected from all users. In this sense, the

Data: set U
Results: a set of cloaking regions for every user in U
if SF("Service Area") $\geq \theta_{\max}$ **then**
  repeat
    $v \longleftarrow$ a user from U with the largest θ. If many, chooses the one with the largest K, (K_v);
    $\mathbb{C}'_v \longleftarrow$ set of cells of the service area having feasible paths to reach CR_v;
    $L_v \longleftarrow$ call Algorithm 1 (q_v, C_v, l_v);
    $\mathrm{CR}_v \longleftarrow$ call Algorithm 2 (L_v, C_v, K_v, $\mathbb{C}'_v$);
    repeat
      $E_{\max} \longleftarrow 0$;
      $i \longleftarrow 0$;
      for $i < m$ **do**
        $c' \longleftarrow$ from $CR_v \setminus \{C_v\}$ with a probability $\propto$ cell's occupancy;
        if $\mathrm{SL}(\mathrm{CR}_v - \{c'\}) > \theta_v$ **then**
          if $E(\mathrm{CR}_v) > E_{\max}$ **then**
            $E_{\max} \longleftarrow E(\mathrm{CR}_v - \{c'\})$;
            $\mathrm{CR}_t \longleftarrow \mathrm{CR}_v - \{c'\}$;
          end
        end
      end
      if $E_{\max} \neq 0$ **then**
        $\mathrm{CR}_v \longleftarrow \mathrm{CR}_t$
      end
    until $(\#(\mathrm{CR}_v) = K_v)$ or $(E_{\max} = 0)$;
    for every user u in CR_v **do**
      if there exist feasible paths from CR_u to CR_v **then**
        Set CR_v for user v only if $K_v - \Delta \leq K_u \leq K_v$ and $\theta_u \leq \theta_v$ and $l_u \leq l_v$ and query type $q_u \in L_v$;
        Update set U removing those users whose cloaking region is CR_v;
      end
    end
  until $U == \varnothing$;
end

Algorithm 5: Diversity top-down cloaking batching algorithm (DTD).

active adversary could at most possess the same information as the LBS server.

In this scenario, our DBU and DTD algorithms mitigate the consequences of an inference attack. First, our algorithms guarantee that all the queries chosen by applying l-diversity have similar query probabilities, and the active adversary cannot distinguish the actual query from the set of l-diversity, even when it executes our schemes several times. Similarly, applying K-anonymity guarantees the protection of the privacy of the user's location. Second, since the cells that make up a CR are chosen randomly, a user's actual location can be blurred in some candidate cells. Therefore, the adversary cannot infer any helpful information from the users.

In order to meet the location safety requirement, it is required that the ratio between the area enclosed by a CR and the actual number of users who are in it is greater than the security level demanded by each user. When this requirement is not met, DTD and DBU add new cells to the K-anonymity set to accomplish it. However, an attacker can attempt to drop these new cells. The conditions of K-anonymity and l-diversity are also applied to these new selections to prevent this situation.

5.3. Resistance to Accessibility Attack

Theorem 3. *Our schemes are resistant to accessibility attack.*

Proof. Here, a passive adversary is assumed that has information on all the CRs requested by a user and can perform the following analyses. Consider two consecutive cloaking regions CR_1 computed at timestamp τ_1 (recall that t_i is a point of time for the timestamp in τ_i in $\mathbb{S}$) and CR_2 computed at timestamp τ_2 for a given user, where $\tau_1 < \tau_2$. The attacker computes the service area's topology graph and verifies if all points in CR_2 are reachable from CR_1. This attack is successful when this adversary finds no-reachable points in CR_2. Moreover, assuming that the maximum speed in the network area is known, the adversary can reduce the size of CR_2 by checking whether it is feasible to achieve any point in CR_2 from any point in CR_1 in a period $t_2 - t_1$ at the maximum speed.

Table 1: Summary of the time complexity of our proposed algorithms.

Algorithm	Time complexity
Algorithm 1	$O(l\|Q\|)$
Algorithm 2	$O((\max(l, K, s)))$
Algorithm 3	$O(\max(l\|Q\|, (\max(l, K, s)))$
Algorithm DBU	$O(\|U\|(\max(l\|Q\|, (\max(l, K, s)))(\max(l, K, s)))$
Algorithm DTD	$O\left(\|U\|\left(l\|Q\|(\max(l, K, s)) + \max\left(\mathrm{CR}_v, m^2\right)\right)((\max(l, K, s)))\right)$

Both DTD and DBU verify if the candidate cells composing a new CR are workable to be reached from the user's current CR, and therefore, this attack can also be prevented.

6. Experimental Environment

We developed a C-based simulator intending to assess our approaches considering the three service requirements (i.e., location privacy, query privacy, and location safety). Aiming to simulate different configurations considering users' number, initial locations of users, restricted spaces, motion users, and service requirements of users, a network domain of 21 $\times$ 21 (similar to Figure 4), which is equally partitioned in cells of size 3×3, was set up. We disseminate a number of users in this area in a range of [200,800]. 25% of these users are stationary, and the remaining ones are mobile users. The stationary users are disseminated based on three probability distributions: uniform (Uniform), exponential (Exponential, 0.5), and Zipf (Zipf, 2.0). We want to simulate scenarios where users are moderately and highly concentrated with these two latter distributions. For example, a few cells have many users regarding Exponential distribution, while others have a few users. Conversely, Zipf has fewer populated cells than the Exponential distribution, but those cells are much more populated. Each mobile user follows a predefined trajectory in the service area, and Figure 4 shows an example of these trajectories. The value for θ is varied between $0.018 = 3 \times 3/500$, assuming all users are located within a small area, and $0.882 = 21 \times 21/500$ when users are equally distributed in the network area.

Aiming to evaluate DBU and DTD in a context where the restricted spaces change (i.e., the LBS has to provide the services to the users in these spaces), six restricted space scenarios are analyzed. In the first scenario, there are no restricted spaces; in the second scenario, 10.8% of the total service area corresponds to a restricted space. The third scenario contemplates 18.1% of the service area; meantime, the fourth scenario covers 43.5%. Regarding the fifth scenario, it involves 69.3%. Finally, the sixth scenario implies the total area.

The variables in all experiments were instantiated as follows: the number of users is 500, mobile users correspond to 375, users with fixed routes are 50, θ is 0.45 (except for Figure 5), the number of ontologies is 16, the number of sensitive terms is 194, K takes the value 7, and the values used in all figures consider the averages of 6 timestamps as well as the minimum and maximum values. Finally, we set $\Delta = 0$ and $m = 4 \times K_u + 1$ when we run all techniques.

We evaluated the performance of our batching techniques using simulations. Four performance metrics are used, including the following:

(i) *Number of cycles*: the average total amount of work (it is related to the time complexity) incurred on building a set of cloaking regions.

(ii) *Size of a cloaking region over K*: the average number of cells conforming to a cloaking region regarding K-anonymity degree demanded by users. The best value is when this ratio equals one, i.e., the size is equal to K, but in general, this metric will be larger than one.

(iii) *Number of cloaking regions*: the number of CRs built by the anonymizer. The minimum value is one since only one CR can be built to protect all users at once. The maximum value corresponds to the number of users deployed in the network area since a CR can specifically be built for each user.

(iv) *Entropy of a cloaking region*: we apply formula (2) to compute the entropy of a CR and then to obtain the average entropy of many computed CRs. With this metric, we want to evaluate the quality of the location privacy protection offered by a CR. The higher the entropy, the better the quality is.

(v) *Safety level*: we apply formula (3) to compute the safety level of a CR. Then, we obtain the average safety level of many computed CRs. With this metric in place, we want to evaluate the quality of the location safety protection offered by a CR, which must be larger than the threshold (θ) established by each user.

We are mainly interested in comparing how the anonymizer performance is impacted and the quality of the computed cloaking regions when we run independently our two batching techniques (denoted as DBU and DTD) and also the baseline. We chose as a baseline, a state-of-the-art approach called by their authors as "*T-Dummy*" [3]. In this work, like our proposals, a sequence of cloaking regions is built for each user for continuous access to a LBS. However, a cloaking region is built for each user independently and only considers K-anonymity restrictions. *T-Dummy* does not consider any l-diversity and location safety restrictions demanded by users. Moreover, *T-Dummy* is not aware of restricted areas, and therefore, it builds a cloaking region

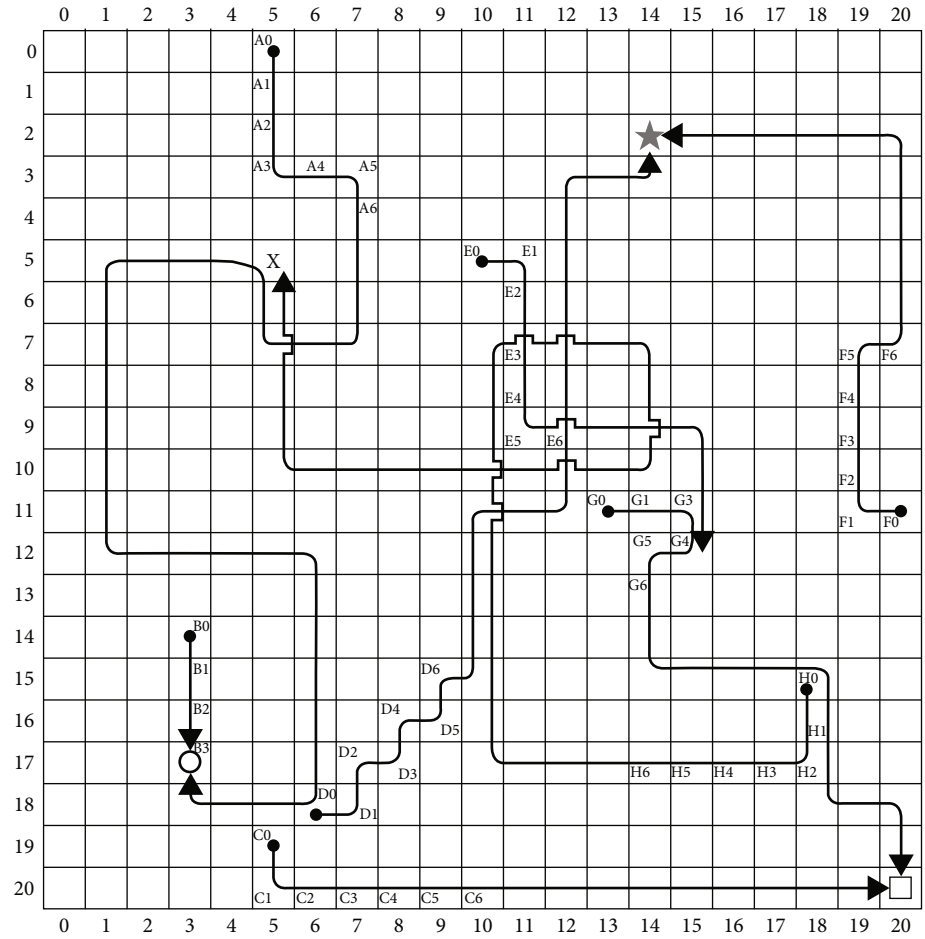


Figure 4: Example of the trajectories of the mobile users in the service area.

every time it is requested. In our results, instead of referring to *T-Dummy* we simply mentioned it as *baseline*.

6.1. Experimental Results. From Figures 6(a) and 6(b), it is workable to appreciate the performance (i.e., the performance is expressed in terms of cycles) for both techniques, when l-diversity is required by users considering three different distributions. The proposed techniques (specifically averages) in both figures outperform the baselines, considering all distributions. These baselines do not consider the restricted spaces; therefore, l-diversity is computed in all timestamps. Roughly, l-diversity is carried out similarly to how K-anonymity is obtained. In Figure 6(a), DTD's behavior remains constant considering the three distributions; however, the technique executes more cycles when the distribution is Uniform, followed by the Zipf distribution and subsequently by the Exponential distribution. When the distribution is Uniform, the technique runs more cycles since it should discard those cells that do not hold the location safety considering at the same time that it should satisfy the user's requirements; this latter can be seen as an exhaustive search since all users are distributed in cells with similar probabilities. Conversely, the Zipf distribution provides fewer cells occupied by users, such that most of the cells have a single user while the rest contain a high concentration of users. DTD provides better performance when the distribution is exponential since more occupied cells exist than the Zipf distribution. Finally, it is worth noting that the probability of submitting a query for some user is linked to the cell probability where the user is.

On the other hand, in Figure 6(b), DBU's performance is displayed, which follows a similar DTD behavior. Nevertheless, when the Exponential distribution occurs, the technique chooses $4K$ and $2K$ (i.e., this is due to the invariant of the technique) cells with similar odds to build the cloaking region. In this case, the role of maximum entropy in the technique's invariant is vital considering the users' distributions. In this way, the technique's performance is more efficient in cycles when there are few cells populated by users (i.e., unlike the Uniform distribution, the Zipf distribution presents few occupied cells, and very few of them have a high concentration of users). Broadly speaking, DBU is more sensitive to the users' distributions than DTD; however, DBU is more efficient in terms of executed cycle numbers. This latter is in line with the time complexities of algorithms, where DTD presents a more expensive complexity than DBU.

In either Figures 6(a) or 6(b), when our techniques are compared to the baseline approach, we can observe that our techniques incur less computational cost since several users share a cloaking region.

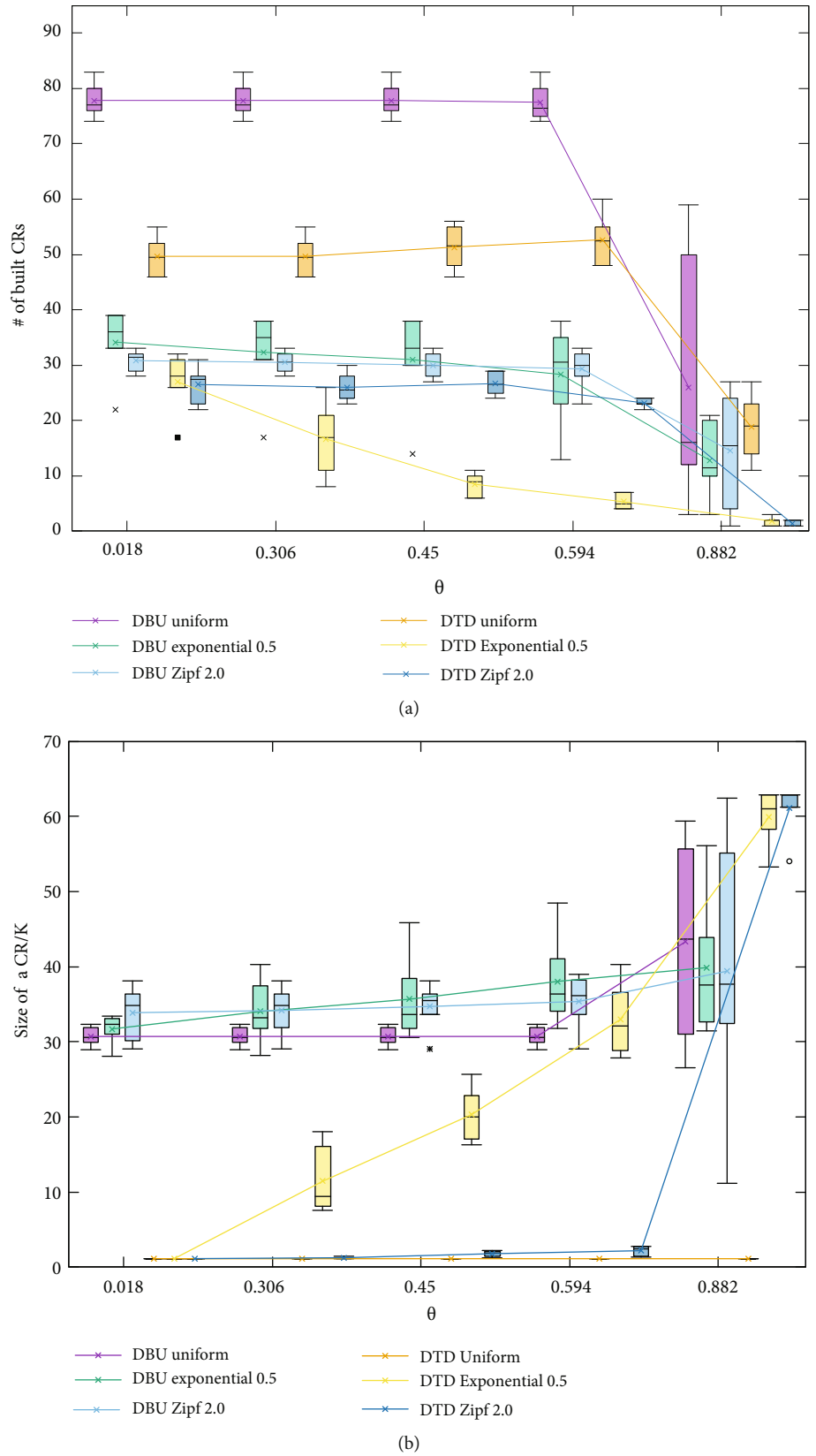


Figure 5: Effect of the safety threshold (θ).

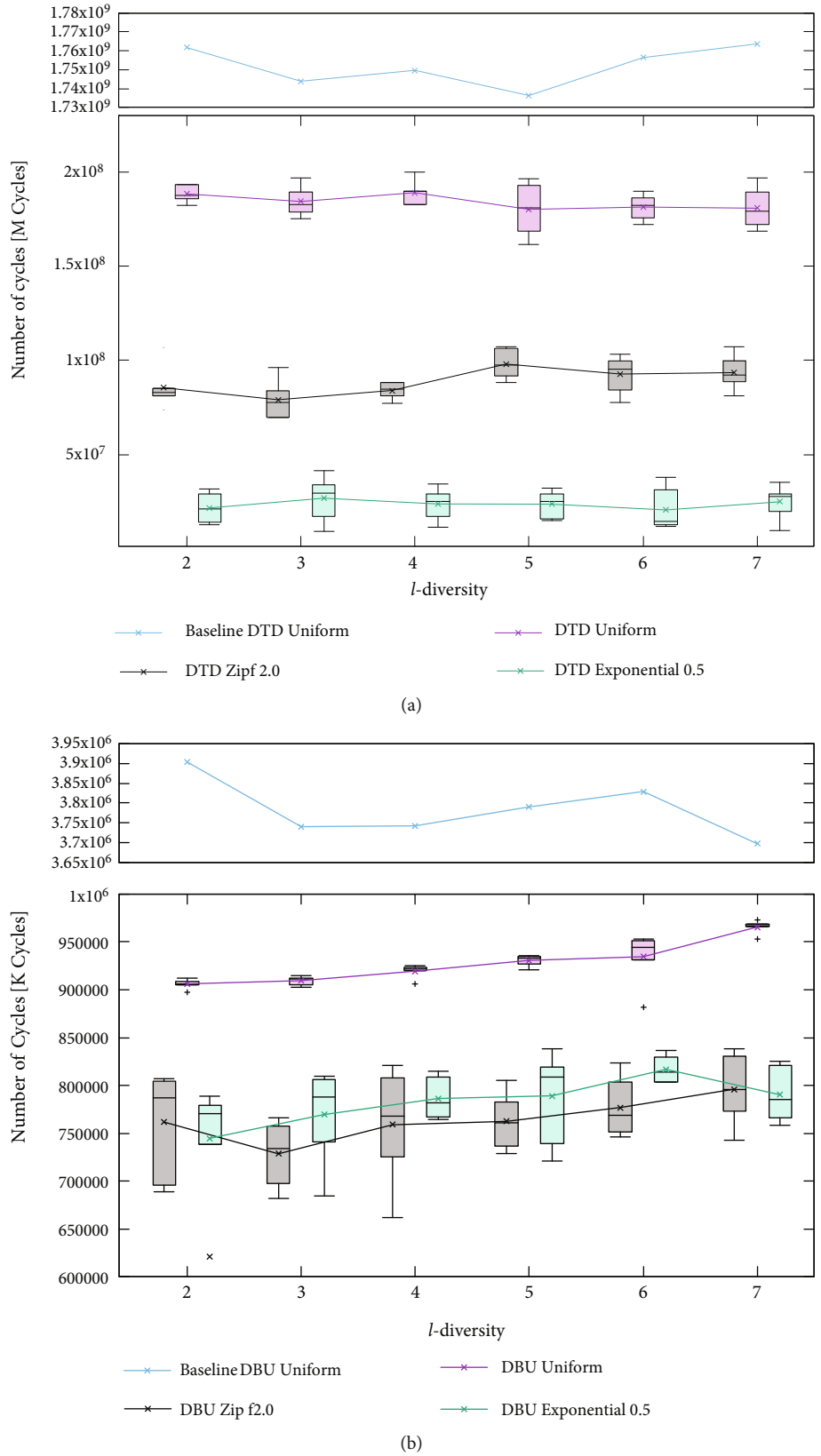
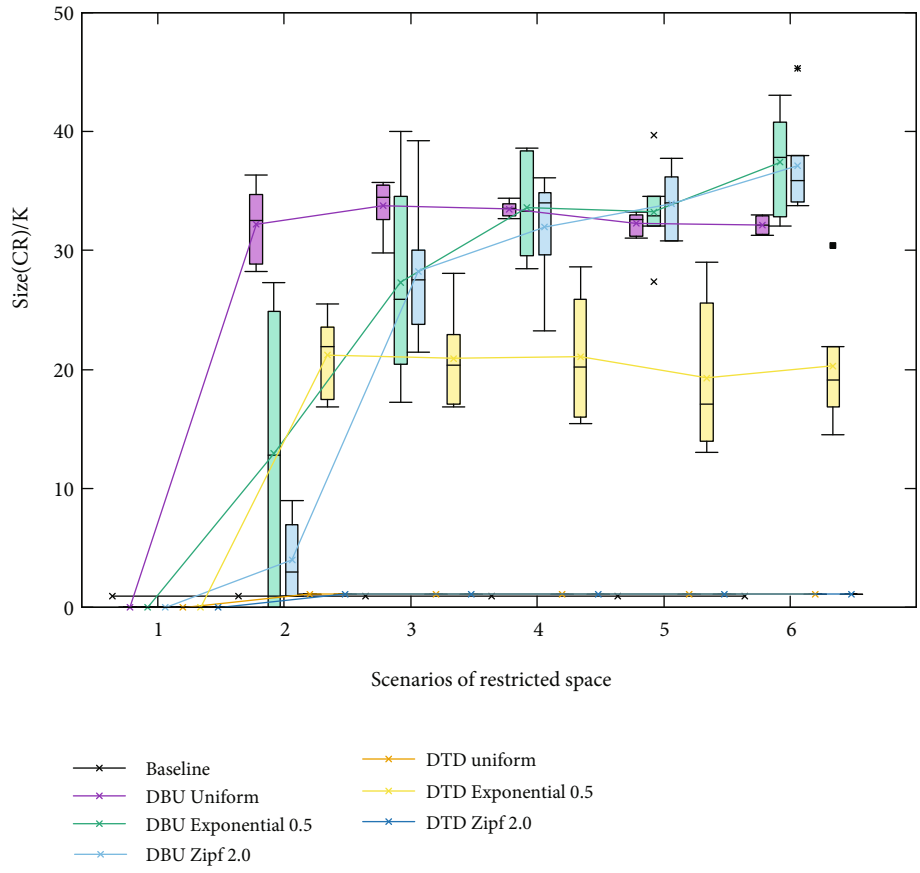


Figure 6: Effect of the query privacy (l-diversity).



(a)

Figure 7: Continued.

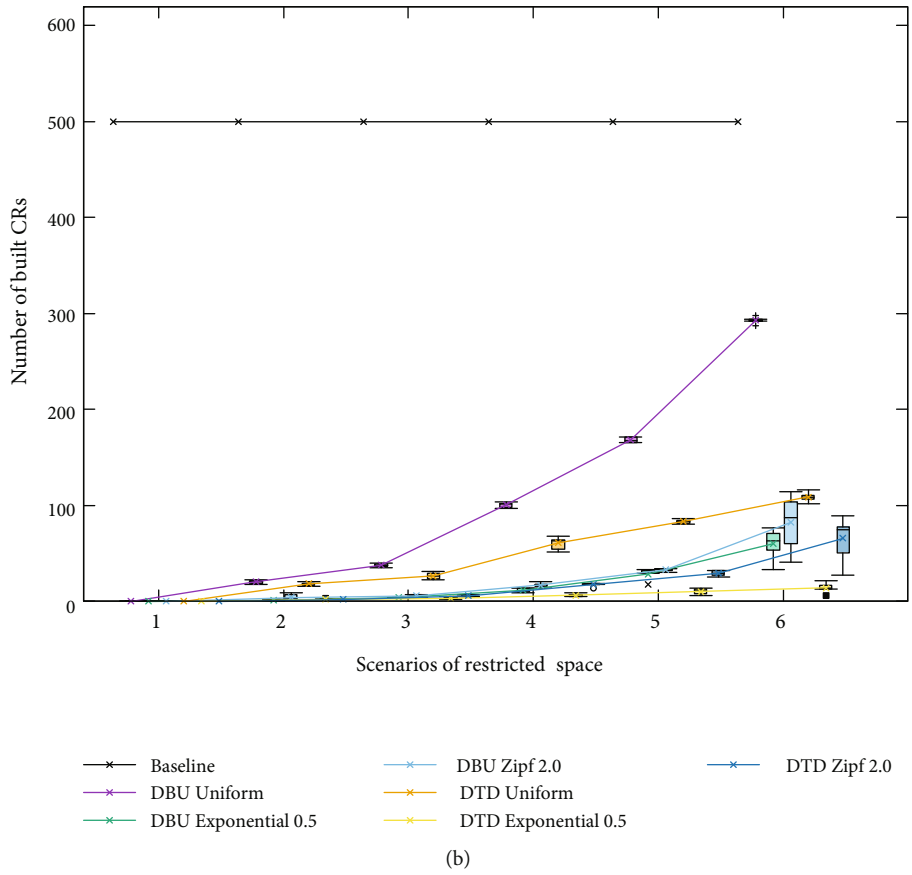


Figure 7: Size and number of built CRs.

In Figure 7(a), the average number of cells used to build CRs is analyzed when restricted areas are increased. DTD has a similar behavior when the users' distributions are Uniform and Zipf. In simple words, most cells have a single user while the rest are highly populated (in particular for Zipf distribution), but this is enough to satisfy the three services simultaneously: location safety, location privacy, and query privacy. Nevertheless, when the distribution is Exponential, the technique tends to increase the average number of cells to build CRs (i.e., fulfilling the location safety and query privacy). On the other hand, DBU uses a greater cell average to build CRs than DTD considering the three distributions. When restricted spaces are greater than 43%, the cells' average does not suffer a substantial variation regarding the distributions. The main differences between algorithms are underlined, first, by how CRs are built while the user requirements are satisfied; second, this difference also depends on how users are spread at the service area. When our techniques are compared to the baseline approach, we can observe that our techniques compute a smaller amount of CRs since the baseline approach builds a CR for each user independently of the location privacy requirements of other users.

In Figure 7(b), CRs for each technique are displayed. Note that both techniques provide more cloaking regions when the Uniform distribution takes place according to the increment of restricted spaces. This behavior is in line with what is exposed in Figure 7(a), where the averages of cells used by both techniques are also higher. Note that the number of CRs built by DBU considering all distributions is similar among them. On the other side, from the fourth to the sixth restricted space, DTD provides the smallest number of CRs, such that there are some cells which contain many users (see Figure 7(a)).

The average entropy for both techniques considering the three distributions is exposed in Figure 8. The best average entropy for DBU is presented under the Uniform distribution, which makes sense since cells inside the service area have the same probabilities (i.e., entropy is maximized for service area users). Following the same trend, the best average entropy presented by DTD is obtained under Uniform distribution—which is very similar to the Zipf distribution for this technique—notably, the best entropy averages shown by both techniques are consistent with the results displayed in Figure 7(b). Here, there exist more cloaking regions according to the increasing of restricted spaces, but the entropy is more or less constant. In turn, the worst entropies for both algorithms are given under the Exponential distribution.

Some interesting conclusions can be extracted from Figures 7(b) and 8. First, in a service area where users are spread out with a probability distribution similar to a Uniform distribution, DBU would bring a hard task for an adversary who wants to determine the real user location. Nevertheless, we believe this situation is ideal and escapes from real-world scenarios. Note that although DTD's invariant does not search to maximize the entropy as a first

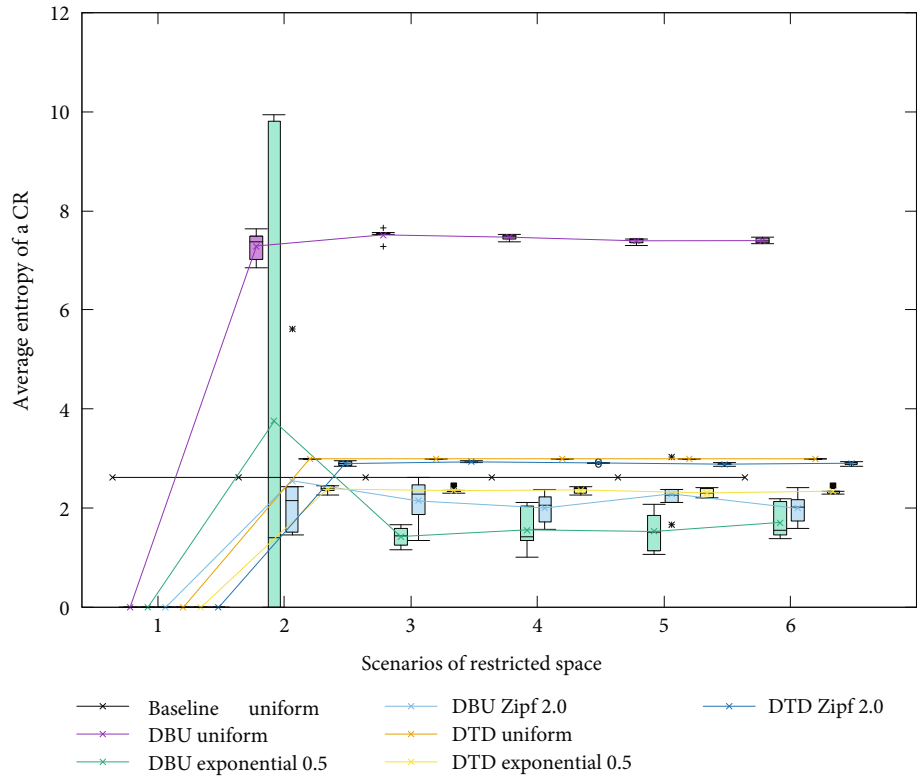


Figure 8: Averages of entropies over CRs.

priority, by bringing the location safety first, it converges to a better entropy when users are not evenly distributed, such as occurs in the Uniform distribution.

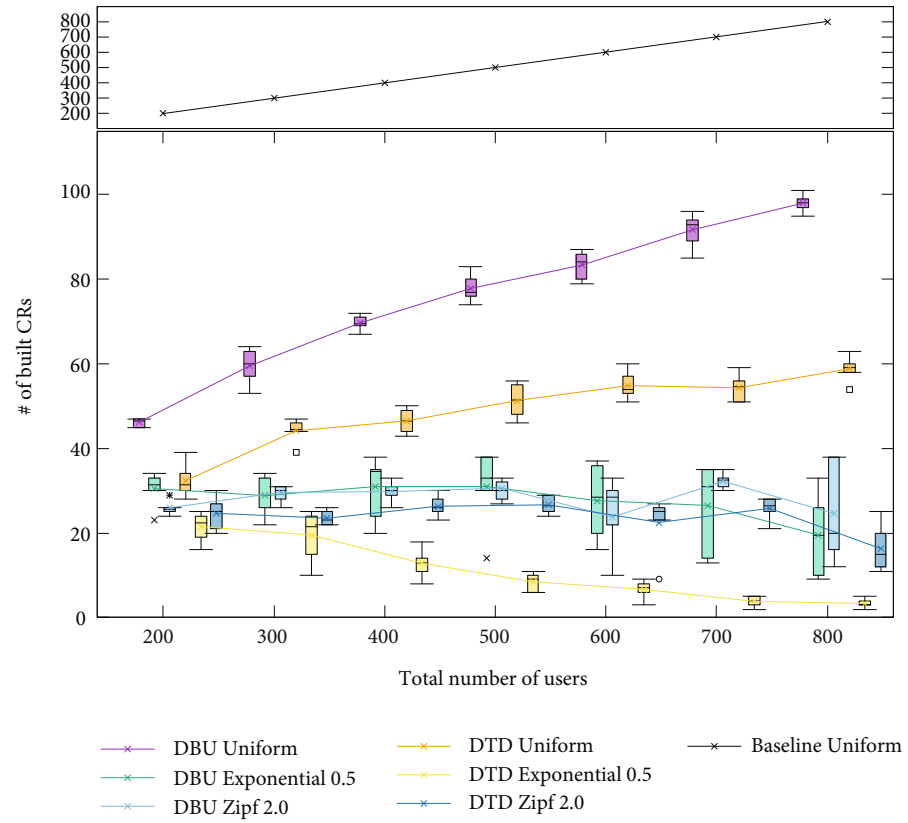
When our techniques are compared to the baseline approach, we can observe that our techniques compute cloaking regions whose entropy is equal or higher than for the baseline. This result is because some cloaking regions are overdimensioned (i.e., the size of this CR is larger than K) due to the location safety requirement demanded by users. When a cell is highly populated, our techniques will choose other less densely populated cells having similar l -diversity characteristics.

Figure 9(a) shows the number of CRs built by both techniques. In this graph, the number of CRs is the largest when users are distributed uniformly. This is because users are disseminated in a larger area, and a few are in the same cells. Furthermore, DTD tends to compute a smaller amount of CRs when compared to DBU. Additionally, when the distribution of users becomes more skewed, users tend to be located in the same cells, and many of them share the same CRs. As a result, there is a reduction of CRs built. On the other hand, independent of the users' distribution, DTD builds a smaller amount of CRs than DBU. This latter occurs because DTD begins the computation of a CR, setting the entire network area as an initial CR candidate. Therefore, a CR computed by DTD satisfies many more users at once than that by DBU. As we previously explained, it is not surprising that the baseline approach builds more cloaking regions than our approaches. The baseline does not verify whether other users can share the same cloaking region and builds cloaking regions even when users are in nonrestricted areas.

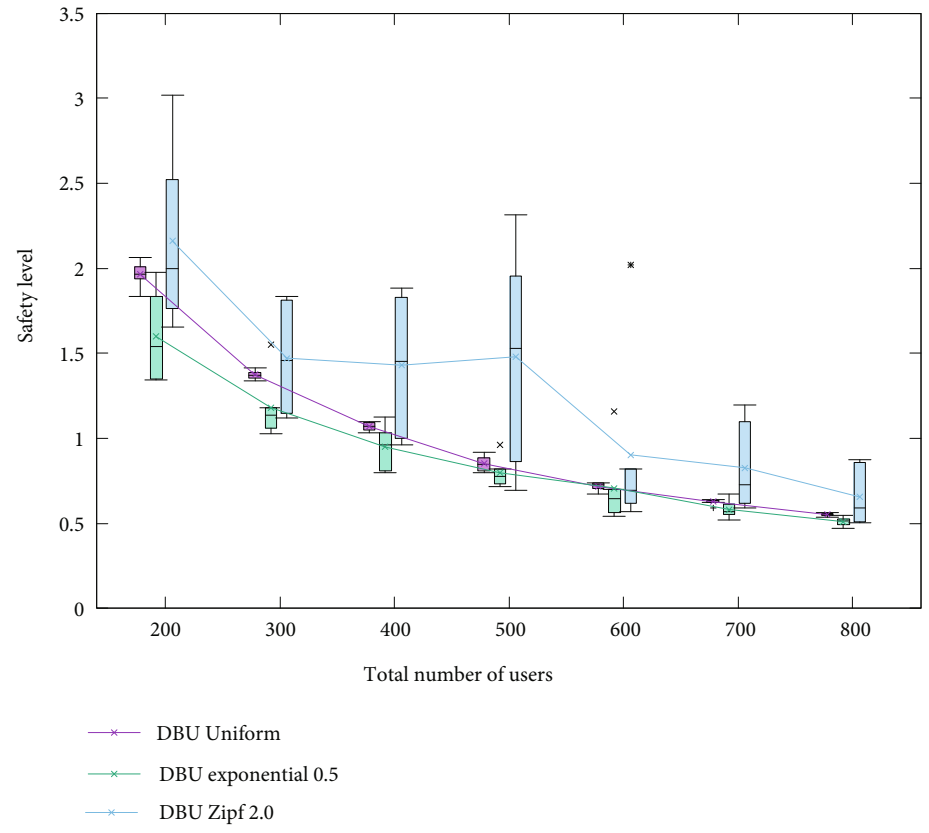
Figure 9(b) displays the safety level (SL) of CRs built by DBU considering different distributions of users within the service area. Note that SL decreases when the users' number grows. This result was expected since SL, by definition, declines when more users are located in the same geographic area. Furthermore, take into consideration that the highest SL is obtained when the distribution is Uniform. Finally, due to DTD providing similar results, we decided to omit its results.

On the contrary, a smaller SL can be observed when users are distributed nonuniformly (Zipf and Exponential). Furthermore, this result is coherent with the nonuniform case since many cells are overpopulated and have a lower SL than the Uniform distribution. Finally, and in conclusion, the computed SL is greater than the θ threshold for all studied scenarios.

Figure 5(a) shows the number of CRs built for all techniques. Unsurprisingly, there is a slight tendency of decreasing the number of built CRs when θ is increased. Consider that this result is more evident when θ grows from 0.594 to 0.882. Recall that this technique has to accomplish the θ required, considering at the same time that the number of users is fixed. To achieve this goal, the area covered by a CR had to be increased. In turn, when the distribution was Uniform, more CRs were built since users were evenly distributed in all cells that form the service area. On the one side, when the distribution becomes more skewed, it is possible to appreciate a decreasing number of CRs since users



(a)



(b)

Figure 9: Impact of the number of users.

are mostly located within a smaller area. On the other side, consider that BU builds more CRs than DTD since the former technique finds the minor CR that satisfies users' K-anonymity; thereafter, it begins to enlarge the area until θ is achieved. Lastly, note that DTD builds a smaller amount of CRs when it is compared to DBU.

Figure 5(b) corroborates the conclusions obtained before. Each time θ is increased, both techniques converge to build larger CRs since a greater θ demands a larger CR. Interestingly, DBU tends to build larger CRs than DTD, but when θ is increased from 0.594 to 0.882, this situation becomes the opposite, and DTD builds larger CRs. This result makes us think that it should be appropriate to use DTD when θ is more relaxed and then switch to DBU when θ is larger or demanding.

7. Conclusions and Future Work

This paper introduced two batching techniques to build cloaking regions for users with diverse requirements, such as diverse location privacy, query privacy, and location safety. Those requirements are solicited to cLBS, considering that users constantly change their locations when requesting a service. Our proposed techniques attempt to balance computational cost at the anonymizer and LBS. Toward that goal, both techniques take into consideration whether users are in a restricted space; from this baseline, when users are not in a restricted space, it is unnecessary to satisfy their requirements. Furthermore, aiming to provide query privacy, a new notion of geographical semantics based on ontologies is introduced. This notion considers sensitive terms in the query, which brings significant savings when the l-diversity is required, and the query does not have sensitive terms. Our techniques also consider several safeguards against three types of attacks (inference, collusion, and accessibility) that intend to reduce a user's location uncertainty through a cloaking region.

Extensive experimentation was carried out using simulation. Different scenarios were evaluated considering metrics such as cloaking regions, computational saving, entropy, and size of cloaking regions. In addition, the notion of the timestamp was incorporated with the purpose of emulating the users' motion in the space through time. Also, different configurations of restricted spaces were considered and evaluated, providing for their impact on the metrics over both techniques.

From empirical results, the following remarks were extracted:

(i) Regarding the l-diversity metric, DBU is more sensitive to the l-diversity than DTD. Nonetheless, DBU has a more efficient behavior in terms of executed cycle numbers

(ii) As regards the cell average used to build CRs when restricted areas are increased, DBU uses a greater cell average to build CRs than DTD considering the three distributions

(iii) Referred to CRs, both techniques provide more CRs when users are evenly spread out. DTD provides the smallest number of CRs, when users are not evenly distributed and restricted spaces increased

(iv) Empirical results align with the time complexities for both techniques, and these techniques are more efficient (in terms of executed cycles) than the presented baseline

(v) Concerning entropy, the worst results for both techniques are obtained under Exponential distribution; when the distribution is more biased like Zipf, both techniques provide similar results

(vi) DTD seems to show the right balance between size, K-anonymity, and location safety

(vii) Roughly, both techniques present stable behaviors (i.e., there is a tendency for every metric) whenever the users' number is increased. This latter suggests that both techniques are scalable

Further experiments should be carried out to obtain more precise values for all metrics while varying the safety location. Nevertheless, we believe that our results are preliminary yet promising. As a future work, we wish to test diverse scenarios and find optimal values for some system parameters, such as m and Δ. We would also like to simulate other scenarios because DBU seems to work better when users are distributed uniformly and DTD when the distribution is skewed and to increase the safety level demanded of users in order to determine feasible conditions to build a cloaking region satisfying location privacy, location safety, and query privacy restrictions.

Data Availability

The data used and the simulator from which this data was obtained to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

Dedicated to Pedro Rodríguez-Moreno, a former group researcher and friend, who coauthored and supported several research articles presented at several international conferences and journals. The Universidad del Bío-Bío, Campus Concepción in Chile, under grants DIUBB 184615 1/I and DIUBB 2130253 IF/R, and the Group of Smart Industries and Complex Systems (gISCOM), under grant DIUBB 195212 GI/EF, supported partially the work presented in this article.

References

[1] M. Decker, "Location privacy-an overview," in *2008 7th International Conference on Mobile Business*, pp. 221–230, Barcelona, Spain, July 2008.

[2] A. S. Saxena, M. Pundir, V. Goyal, and D. Bera, "Preserving location privacy for continuous queries on known route," in *Information Systems Security*, pp. 265–279, Springer, Berlin Heidelberg, 2011.

[3] B. Niu, S. Gao, F. Li, H. Li, and Z. Lu, "Protection of location privacy in continuous LBSs against adversaries with background information," in *2016 International Conference on Computing, Networking and Communications (ICNC)*, pp. 1–6, Kauai, HI, USA, February 2016.

[4] A. Pingley, N. Zhang, X. Fu, H. A. Choi, S. Subramaniam, and W. Zhao, "Protection of query privacy for continuous location based services," in *2011 Proceedings IEEE INFOCOM*, pp. 1710–1718, Shanghai, China, April 2011.

[5] P. Galdames, C. Gutierrez-Soto, and A. Curiel, "Batching location cloaking techniques for location privacy and safety protection," *Mobile Information Systems*, vol. 2019, Article ID 9086062, 11 pages, 2019.

[6] G. Tobar, P. Galdames, C. Gutierrez-Soto, and P. Rodriguez-Moreno, "A batching location cloaking algorithm for location privacy protection," in *Collaborative Technologies and Data Science in Smart City Applications*, pp. 26–36, 2018.

[7] G. Yang and Y. Cai, "Full location privacy protection through restricted space cloaking," *Journal of Information Processing*, vol. 25, pp. 756–765, 2017.

[8] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *ACM MobiSys'03*, pp. 31–42, 2003.

[9] B. Gedik and L. Liu, "Protecting location privacy with personalized k-anonymity: architecture and algorithms," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 1–18, 2008.

[10] M. F. Mokbel, C.-Y. Chow, and W. G. Aref, "The new Casper: query processing for location services without compromising privacy," in *Proceedings of the 32nd International Conference on Very Large Data Bases. VLDB '06*, pp. 763–774, Seoul, Korea, 2006.

[11] R. Cheng, Y. Zhang, E. Bertino, and S. Prabhakar, "Preserving user location privacy in mobile data management infrastructures," in *Proceedings of the 6th International Conference on Privacy Enhancing Technologies. PET'06*, pp. 393–412, 2006.

[12] X. Toby and Y. Cai, "Location anonymity in continuous location-based services," in *Proceedings of the 15th Annual ACM International Symposium on Advances in Geographic Information Systems*, pp. 1–8, 2007.

[13] T. Xu and Y. Cai, "Feeling-based location privacy protection for location-based services," in *Proceedings of the 16th ACM conference on Computer and communications security. CCS'09*, pp. 348–357, 2009.

[14] S. Zhang, K.-K. R. Choo, Q. Liu, and G. Wang, "Enhancing privacy through uniform grid and caching in location-based services," *Future Generation Computer Systems*, vol. 86, pp. 881–892, 2018.

[15] Y. Ji, R. Gui, X. Gui, D. Liao, and X. Lin, "Location privacy protection in online query based-on privacy region replacement," in *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0742–0747, Las Vegas, NV, USA, January 2020.

[16] A. S. Saxena, D. Bera, and V. Goyal, "Modeling location obfuscation for continuous query," *Journal of Information Security and Applications*, vol. 44, pp. 130–143, 2019.

[17] Y. Qiu, Y. Liu, and J. Chen, "A novel location privacy-preserving approach based on blockchain," *Sensors*, vol. 20, no. 12, p. 3519, 2020.

[18] L. C. Mutalemwa and S. Shin, "Achieving source location privacy protection in monitoring wireless sensor networks through proxy node routing," *Sensors*, vol. 19, p. 1037, 2019.

[19] C.-Y. Chow, M. F. Mokbel, and X. Liu, "A peer-to-peer spatial cloaking algorithm for anonymous location-based service," in *Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems. GIS'06*, pp. 171–178, 2006.

[20] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Achieving k-anonymity in privacy-aware location-based services," in *IEEE INFOCOM 2014- IEEE Conference on Computer Communications*, pp. 754–762, Toronto, ON, Canada, 2014.

[21] X. Li, M. Miao, H. Liu, J. Ma, and K.-C. Li, "An incentive mechanism for K-anonymity in LBS privacy protection based on credit mechanism," *Soft Computing*, vol. 21, no. 14, pp. 3907–3917, 2017.

[22] S. G. M. Koo, C. S. G. Lee, and K. Kannan, "A genetic-algorithm-based neighbor selection strategy for hybrid peer-to-peer networks," in *Proceedings. 13th International Conference on Computer Communications and Networks (IEEE Cat. No.04EX969)*, pp. 469–474, Chicago, IL, USA, 2004.

[23] M. R. Nosouhi, V. V. Pham, S. Yu, Y. Xiang, and M. Warren, "A hybrid location privacy protection scheme in big data environment," in *GLOBECOM 2017 IEEE Global Communications Conference*, pp. 1–6, Singapore, 2017.

[24] L. Li, Z. Lv, X. Tong, and R. Shi, "A location privacy protection scheme based on hybrid encryption," in *Proceedings of the 3rd International Conference on Computer Science and Application Engineering. CSAE 2019*, pp. 1–6, 2019.

[25] T. Xu and Y. Cai, "Location cloaking for safety protection of ad hoc networks," in *Proc. of IEEE Int'l Conf. on Computer Communications (INFOCOM'09)*, pp. 1944–1952, Rio de Janeiro, Brazil, 2009.

[26] X. Toby and Y. Cai, "Location safety protection in ad hoc networks," *Ad Hoc Networks*, vol. 7, no. 8, pp. 1551–1562, 2009.

[27] B. Niu, X. Zhu, X. Lei, W. Zhang, and H. Li, "EPS: encounter-based privacy-preserving scheme for location-based services," in *2013 IEEE Global Communications Conference (GLOBECOM)*, pp. 2139–2144, Atlanta, GA, USA, 2013.

[28] B. Niu, X. Zhu, W. Li, H. Li, Y. Wang, and Z. Lu, "A personalized two-tier cloaking scheme for privacy-aware location based services," in *2015 International Conference on Computing, Networking and Communications (ICNC)*, pp. 94–98, Garden Grove, CA, USA, 2015.

[29] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "L-diversity: privacy beyond k-anonymity," in *22nd International Conference on Data Engineering (ICDE'06)*, pp. 24–24, 2006.

[30] J. Wang, Z. Cai, Y. Li, D. Yang, J. Li, and H. Gao, "Protecting query privacy with differentially private k-anonymity in location-based services," *Personal and Ubiquitous Computing*, vol. 22, no. 3, pp. 453–469, 2018.

[31] A. Ye, Y. Li, and X. Li, "A novel location privacy-preserving scheme based on *l*-queries for continuous LBS," *Computer Communications*, vol. 98, pp. 1–10, 2017.

[32] C. Faúndez, P. Galdames, and C. Gutierrez-Soto, "A batching cloaking scheme for continuous location-based services," in *Proceedings of the Collaborative Technologies and Data Science in Artificial Intelligence Applications*, pp. 24–29, 2020.

[33] Y. Cui, F. Gao, W. Li et al., "Cache-based privacy preserving solution for location and content protection in location-based services," *Sensors*, vol. 20, no. 16, p. 4651, 2020.

[34] M. Li, Y. Wang, G. Yang et al., "DGS-HSA: a dummy generation scheme adopting hierarchical structure of the address," *Applied Sciences*, vol. 10, no. 2, p. 548, 2020.

[35] L. Sweeney, "K-anonymity: a model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[36] Z. Xiang and B. Pan, "Travel queries on cities in the United States: implications for search engine marketing for tourist destinations," *Tourism Management*, vol. 32, no. 1, pp. 88–97, 2011.