Hindawi
Mobile Information Systems
Volume 2019, Article ID 9086062, 11 pages
https://doi.org/10.1155/2019/9086062

Research Article

Batching Location Cloaking Techniques for Location Privacy and Safety Protection

Patricio Galdames,[1] **Claudio Gutierrez-Soto**,[1] **and Arturo Curiel**[2]

[1]*Departmento de Sistemas de Información, Universidad del Bío-Bío, Concepción 4051381, Chile*
[2]*Facultad de Estadística e Informática, Universidad Veracruzana, Veracruz, Mexico*

Correspondence should be addressed to Patricio Galdames; pgaldames@ubiobio.cl

Received 20 September 2018; Revised 15 November 2018; Accepted 10 December 2018; Published 2 January 2019

Guest Editor: Nelson Baloian

Location-based services (LBSs) have become a profitable market because they offer real-time and local information to their users. Although several benefits are obtained from the usage of LBSs, they have opened up many privacy and safety challenges because a user needs to release his/her location. To tackle these challenges, many location-cloaking techniques have been proposed. Even though these solutions are effective in protecting either location privacy or location safety, they do not provide unified protection. Furthermore, most of them do not address the potential bottleneck in the anonymity server as a high demand of location and safety protection is requested. Finally, they do not take into account the potential impact of processing a large amount of location-cloaked queries. This paper deals with the efficient construction of location-cloaking areas for many users, who have both privacy and safety requirements. To achieve this goal, the construction of location-cloaking areas is carried out in batches. The LBSs' batch processing takes advantage of users who are close to each other and who have similar requirements. Two batching techniques to build cloaking regions are analyzed using simulations. Empirical results show our techniques are able to balance the anonymizer workload, quality of location privacy and safety protection, and LBS workload.

1. Introduction

This paper is an extension of a work in progress published at CODASSCA [1] in the context of location-based services (LBSs). An LBS is a geographic information system connected to the Internet, whose main goal is to track the location of their users within a wireless network. These users report their exact position when a service is required, by using their location-enabled mobile devices. Having received users' location, the LBS offers them real-time information about other users who are close; for example, a user who suffered a serious car accident should submit his/her current position as soon as possible to get prompt medical support.

On the other hand, when LBS users reveal their locations, they could endanger their safety and privacy integrity. An adversary, listening to this information, could not only determine their identities but also track them to any place they go to. Moreover, the same LBS providers should keep the data confidential and should not release this information to unknown third parties. All of these issues have motivated a series of research on location-cloaking techniques.

The key idea is to limit location resolution to achieve a desired level of protection. When requesting for an LBS, users report a cloaking region instead of their exact positions. A cloaking region needs to contain a user's current position and also encloses other locations in which the user could be located. Most of the approaches [2–11] are based on a trusted third party, called *the anonymizer*, which is responsible for selecting these additional locations depending on what type of protection a user is demanding. Other techniques such as [12–16] assume that the same users, collaborating with other peers, can compute their own cloaking regions. In addition, a few articles have proposed a hybrid approach, in which an anonymizer and users collaborate to create cloaking regions [17–19].

The approaches proposed in [2–4, 6, 12] support anonymous use of an LBS. An adversary will not know the identity of the user located at each location even if he/she manages to identify all these users by matching the cloaking region with public information available in white and yellow pages. In contrast, the techniques in [1, 7, 8, 14, 15] ensure that each cloaking region contains locations that have been visited by at least K different users. Because these users visited the region at different periods of time, it prevents an adversary from identifying the user who was in that region at the moment the LBS was requested. Thus, the user's location privacy is protected from the time dimension.

Other techniques [20, 21] have been proposed to protect location safety. Their goal is also to build a cloaking region containing a user's position, but they want to prevent an adversary from combing the entire region to locate and destroy a target user and every other user located within that region. The idea behind this concept is that the target user and the other users located nearby could have some common purpose. For example, let us consider a set of wireless sensors deployed in some area and working together to detect or track specific objects, like a tank. In this case, the adversary is not concerned about finding the identity of each sensor but simply wants to locate and destroy each one of them.

Reducing location resolution limits privacy and safety risks but adds more workload on the LBS server and the anonymizer. First, on the LBS server, a precise location is more convenient because a query result is only computed with respect to a specific position. However, when a user location is cloaked, (i.e., the user's real location is mixed with other possible locations) the LBS server also needs to compute the responses for these other locations. We will refer to a query in which the location has been cloaked as a *Location-Cloaked Query* (LCQ). In a system with a large number of users, the processing of LCQs can be overwhelming to the LBS server and could bring it down. This is especially problematic (in terms of runtimes) when the server has to deal with large cloaking regions, which happens when users request a high level of protection (e.g., high values of K are requested by users).

Secondly, the anonymizer can also be problematic. If the anonymizer has high demand with the aim of finding an optimal (size) cloaking region for a large amount of clients, the anonymizer becomes a bottleneck and therefore clients experience high response times. This is undesirable if the LBS wants to support real-time applications. A solution for this issue is to build the smallest amount of cloaking regions that satisfy the privacy and safety requirements of every user. However, this approach can end up returning cloaking regions larger than those needed. Thus, an approach that balances the LBS and anonymizer workloads is needed.

Thirdly, in [1], we consider the problem of building cloaking regions for users demanding only location privacy protection. However, in this paper, we aim to satisfy both location privacy and location safety requirements, which is more challenging. A cloaking region for location privacy must prevent an adversary from distinguishing between a subject demanding protection and others located within this region. The more users present within that area, the better it is for the subject. However, location safety is the opposite; if this region is highly dense, it can become quite attractive for an adversary to comb such a region, localize all users within that area, and destroy them. In this paper, we address the problem of building a set of cloaking regions (CRs) for a large number of users having both location privacy and location safety requirements. Our idea is to build CRs in a large-scale system as long as the anonymizer has processing resources available. In such a real-time processing model, a CR is computed upon its request arrival without any latency when the anonymizer is underloaded. However, when the anonymizer is overloaded, the incoming requests for CRs are queued. These requests are processed in a batch as soon as the anonymizer has hardware resources available. Our research focuses first on how to build a cloaking region satisfying both location privacy and safety requirements. Then, we address the scalability dilemma between the anonymizer and the LBS.

The paper makes the following contributions: we propose a unified approach for building CRs demanding both location privacy and safety requirements. To achieve this goal, we propose two algorithms to batch build a set of cloaking regions. To the best of the authors' knowledge, this problem has not been addressed before in any literature. By using our algorithms we tackle the problem of improving scalability on the anonymizer without potentially compromising the LBS workload, which also, to our knowledge, has not been properly addressed. To measure the effectiveness of these solutions, we have simulated different scenarios (not addressed in [1]) in which users have similar and different location privacy and safety requirements and are disseminated nonuniformly in different places of the service area.

The remainder of this paper is organized as follows: in the section *Related Work*, an extensive review of articles related to the problem of building cloaking regions is presented. Then, we explain the background and basic concepts in the section *System Overview*, and our scalable location privacy and safety protection techniques are presented in the section *Proposed Batching Techniques*. The empirical results are reported in the section *Results and Discussion*. Then, this paper concludes in the section *Conclusions*.

2. Related Work

A wide range of approaches deals with location cloaking techniques. Some of these techniques can be categorized in different ways. In [22], the techniques can be classified as spatial anonymization, obfuscation, and private retrieval methods. Another classification is proposed in [23], where the different methods are categorized as dummy-based, K-anonymity, differential privacy, and cryptography. Unlike the previous classifications, we present the related work according to performance for a single user and a batch of users.

Several approaches present their performances for a single user. Among them, in [24], a scalable fog server architecture with a bus-based edge device was implemented. It is based on the topology of roads: the authors optimize the

allocation of roadside *cloudlets* to better offload the computation tasks in the moving fog servers. The data set reflects the actual movement of the buses with time. A genetic algorithm is used to address the problem. Two metrics are used: the total cost of the roadside cloudlet over the number of buses and the performance comparison between service offloading and non-offloading. The data set used corresponds with collected traces of the fleet of city buses in Seattle. A sample of the data set was chosen under uniform distribution. The authors do not consider privacy issues, rather assuming that the LBS servers themselves are trusted 3rd parties.

In [25], the authors present three dynamic grid-based spatial cloaking algorithms to provide location k-anonymity and location l-diversity in a mobile environment. These algorithms rely on a semitrusted third party to give spatiotemporal cloaking. In the worst case, their method has to consider the entire search space through iteration to create a covering for each user. The metrics considered in this work centre on the algorithm's effectiveness in terms of privacy, quality, time complexity, and scalability. The PrivacyGrid framework and Zipf distribution with parameter 0.6 were used to provide the K values. The authors point out that their algorithms are highly efficient in terms of both time complexity and update cost.

In [26], the automatic generation of cost-effective dummy locations in the clients is presented with the aim to obfuscate the user's real location without a trusted third party. The main metric used here is the number of dummies. According to the authors, three distributions of users are used; however, these are not detailed. The empirical results show that the cost rapidly escalates if a high value of dummies is required: all of them are recalculated individually, on each query. Therefore, response efficiency is necessarily limited by the computing power of the clients.

The cloaking algorithm presented in [3] enables the user to specify the level of anonymity, by specifying restrictions such as the geographical size of the covering. The algorithm takes into consideration the distribution of all users on the map along with their previous cloaking requests. The experiments show the performance under several conditions by using realistic workloads synthetically generated from real road maps and traffic volume data. The empirical results are expressed in terms of success rate, relative anonymity level, and relative spatial/temporal resolution, where Zipf distribution is used to spread out users. Another component studied was the scalability of the extreme cases in terms of the runtime performance. This method works well when the distribution of users is uniform across the entire space but may fail to anonymize effectively when small covering regions are used in low density spots.

Niu et al. [14] consider the decentralized creation of well-crafted dummy locations, intended to maximize both entropy and the covered spatial region. The authors propose a novel Caching-aware Dummy Selection Algorithm (CaDSA). The main evaluated metric incorporates the effect of caching on privacy, which describes the quantitative relation between cache hit ratio and the achieved privacy area on the map of New York City. In the simulations, users follow the Levy walk mobility model. Two caching-aware dummy selection algorithms to improve user's location privacy are proposed. The first algorithm, CaDSA, achieves K-anonymity effectively by selecting some candidate cells with similar query probabilities. The second algorithm, enhanced-CaDSA, considers distance normalization and data freshness. Enhanced-CaDSA improves caching hit ratio along with the overall privacy. However, the authors do not consider efficiency; calculations are performed redundantly in the clients, without analyzing how this might affect communication and storage costs, and it is assumed that users have complete and trustworthy information of other users.

In [27], the authors further improve the work in [14] by caching the dummy locations that contribute to maximizing entropy. In this work, two privacy metrics to measure location privacy are defined. One of these metrics measures the privacy degree achieved for a user when he/she sends a query to the LBS server. The other metric considers the effect of caching and measures the overall privacy achieved for the system. This work uses dummy locations to achieve K-anonymity even when the LBS server has side information. Aiming to evaluate the performance of proposed algorithms, several simulations are carried out, where every 1 minute, 10 users issue a request for LBS service. The query probability of the cell is considered to assess the user probability in order to be chosen. However, contrary to our work, their gains cannot guarantee a response time below some constant φ, which is desired to ensure quality of service (QoS).

The authors in [28] present an algorithm that preserves both query and location privacy, by creating a set of dummy locations that maximize entropy in cells with similar query probability. The entropy-based metric is used to quantify location privacy. Two dummy schemes are considered, optimal and random. Simulation is used to obtain experimental results on a New York map. The Levy walk model is used to generate synthetic data. Final results improve privacy; however, contrary to our method, all calculations have to be performed for each incoming request.

In a recent work, the authors in [29] use a *function generator* of service, which is based on Hilbert curves. It allows anonymization of both the queries and their respective responses. The function generator encodes users' queries into an alternate representation, which is sent to a third-party anonymization service without danger of identification. The anonymizer passes the queries through to the LBS, encoding its responses in the same code given by the function generator. This enables the clients to decode the response themselves upon return, protecting them in case the anonymization server gets compromised. Their experiments used (randomly generated) uniform data sets. The metrics evaluated here correspond to the computational cost on the user side with the aim of building the areas. This method increases the overall computation costs by requiring additional coding/decoding operations around the anonymization procedure. In this sense, it will be as effective as the one used by the anonymization server, with the addition of the codification overhead.

A mixed approach is presented in [30]. In this work, both a semitrusted third party with caching capabilities and client-based anonymization are addressed. The user sends an encrypted query through the semitrusted third party—ignorant of the contents itself—which, in turn, passes them through to the LBS server as an enlarged region, obfuscating the point of origin. The results are returned to the clients through the semitrusted third party, so that they can locally select their points of interest themselves. To assess the approach, a set of moving objects on the real road map of Hennepin County, Minnesota (USA), are generated. Randomly, a vertex of the road map as the location is picked out. The mainly used metric corresponds to computation cost.

To summarize, from an experimental point of view, approaches that address the processing for a single user obtain the user data from a real map or a simulation framework. When a real map is used, all experiments are carried out considering only it. However, other real maps are not considered by which intuitively other results should be reported according to the initial distribution of users on these maps. In most cases, only Levy walk mobility distribution takes place in the experiments. On the other hand, when a simulation framework is used, only Zipf distribution with just one parameter is instantiated (generally, this parameter has a low value). Differing from this approach, in this paper, we use several probability distributions (Zipf, exponential, and uniform) in order to cover a wide range of possible results. Besides, several metrics are evaluated to assess the construction impact of location-cloaking areas on the anonymity server when it has high demand. It should be noted that our techniques deal with batch processing.

Other approaches use batch processing. In [31], a cloaking algorithm over a hierarchy-based representation of a road distribution is presented. The authors' procedure takes into account the spatial restrictions imposed by road systems in order to enhance privacy, both for a single user and batches of users. The proposed framework is evaluated from the aspects of privacy-preserving ability, quality of service, and system performance. A network-based generator of moving objects on the road map of Oldenburg was used to carry out the experiments (a sample of the road map was chosen randomly). Some empirical results to obtain the local privacy were processed in batches. According to the authors, the amount of needed calculation decreases when the batch processing takes place. Contrary to their method, our techniques always consider users in batches, rather than choosing to memorize previous batch responses for future queries.

In [16], the authors present an incentive-based batch algorithm to build a K-anonymity covering with $K-1$ willing participants. In this work, a probability threshold is suggested to indicate a user's reputation on a framework based on fuzzy logic. Batch processing is used to verify the certificates. The main metrics used here are the cost in order to build the areas and number of certificates. Final results show a reduction in processing time and energy consumption. Moreover, their solution needs to continuously calculate new covering regions, at least one per session. However, the distribution of users by space is not mentioned in this paper.

In summary, approaches that deal with batch processing acquire the user data from a real map or from a uniform data set. Similar to the approaches based on the processing for a single user, all experiments are carried out considering the same real map. For this case, we believe that other different data to the same map should be considered. On the other hand, the approach that uses uniform data does not use other distributions. Different to our approach, several probability distributions are used to expand the results, considering other metrics as the entropy at the same time.

Finally, to our knowledge, none of the aforementioned works and those presented in the introduction have properly addressed the efficient building of a large number of cloaking regions for users having heterogeneous privacy and location safety requirements.

3. System Overview

Without loss of generality, we assume that a single anonymity server is used to manage all users, as shown in Figure 1. In order to efficiently process each request for a CR, the entire network area is partitioned into a set of $n \times n$ disjoint cells of equal size as shown in Figure 2. Each user u submits a protection request including his/her current location represented as a 2D point $(X_u; Y_u)$, location-based query, and location privacy (K_u) and location safety (θ_u) requirements. We also assume that our system receives a large set of queries and requirements for location privacy and safety protection, which are queued in a waiting list, denoted by U. Finally, our system returns for each user u in a cloaking region, denoted by CR_u, which conforms to the requirements of privacy and safety given by users.

To protect location privacy, we follow a similar approach as shown in [1, 14]. Our system chooses at least K_u cells to maximize the entropy. To do so, when users report their current locations, the anonymizer maintains a count of how frequently a request comes from a given cell. Based on this information, we define the *query probability* as

$$q_i = \frac{\text{Number of requests originated from cell } i}{\text{Number of requests coming from the network area}}, \tag{1}$$

where $\sum_{i=1}^{n^2} q_i = 1$, for all $i = 1, \ldots, n^2$. Besides, the entropy of a given region CR, denoted as $H(\mathrm{CR})$, 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 request probability of cell c_j. This latter probability is computed as $p_j = q_j / \sum_{l=1}^{K} q_l$. The higher the entropy of a CR, the better the location privacy protection offers.

To protect location safety, we follow a similar approach as shown by Xu and Cai [20]. These authors define 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 a CR).

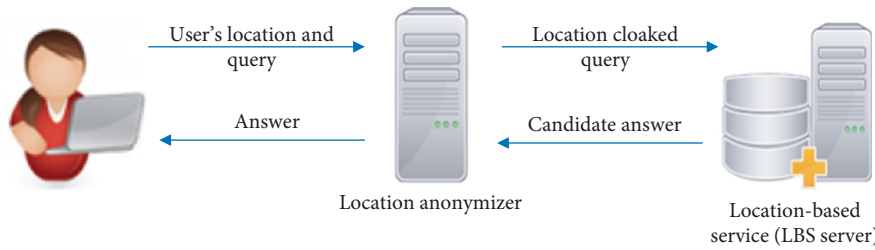

Figure 1: Traditional architecture of an LBS.

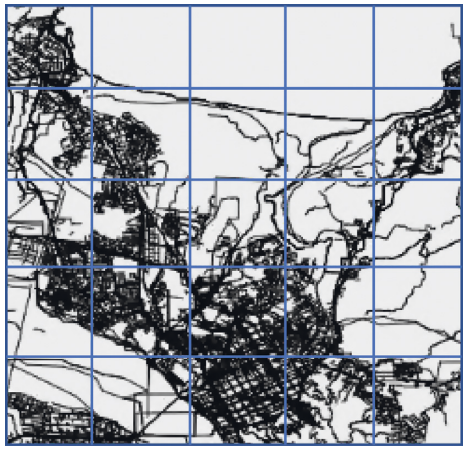

Figure 2: Partitioning of the network area.

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 $\mathrm{SL}(\mathrm{CR}) \geq \theta_u$.

Also, Xu and Cai [8] assume that a CR is a convex region, which is not our case, since a CR is a set of fragmented areas or cells. Now, let us consider a cloaking region, CR, 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, the better its location safety protection.

4. Proposed Batching Techniques

First, we define the following notation to describe our location cloaking techniques:

(i) Let C_N be the set of all cells in the network area sorted in ascending order of their request probability.

(ii) Let U be the current set of users requesting location privacy and safety protection. Given a user u in U, we say c_u is the current cell containing u's exact location and CR_u is the user u's cloaking region.

(iii) Given a user u, we say K_u is the location privacy protection demanded by user u, and similarly, θ_u is the location safety protection demanded by user u.

(iv) Let $\#(\mathrm{CR})$ be the cardinality of a cloaking region CR as the number of cells c_j making up this region.

(v) Let $C(u, r)$ be a subset of C_N, which consists of those "r" neighbor cells at the right and at the left of c_u in C_N. These cells are the ones whose request probability is close to c_u's probability. Thus $\#(C(r)) = 2 * r + 1$.

(vi) Given two cells c_i and c_j in C_N, the distance between these two cells is $|i - j|$.

(vii) Given a cell c_i, we say the occupancy of c_i is the number of mobile users currently located within c_i.

(viii) Let $\theta_{\max}$ is the maximum location safety requirement a user can demand.

We developed two batching techniques to compute cloaking regions at once. The first one, denoted as BU, follows a bottom-up approach because it first finds out a small candidate cloaking region satisfying a given location privacy requirement. Then, it tries to enlarge this region until the location safety requirement is satisfied. The second technique, denoted as TD, follows a top-down approach and works on the contrary. It assumes the entire network is an initial candidate for a cloaking region, and then it attempts to reduce its size while the location safety and location privacy requirements are both satisfied. In both techniques, users having similar location privacy and safety requirements may share a computed cloaking region.

4.1. Bottom-Up Technique. Our bottom-up approach is based on two algorithms denoted by Algorithms 1 and 2. The goal of the first algorithm is to build a candidate cloaking region satisfying the location privacy requirement demanded by a user u. To achieve this goal, this algorithm first finds a candidate set of size $2\dot{K}_u$ cells with the highest entropy (lines 4–6). Finally (line 8), it chooses a set of K_u cells from the previous candidate set at random with a probability inversely proportional to that of the occupancy of a cell. This is done in order to prioritize cells having smaller density of nodes.

Algorithm 2 is a proper batching procedure, whose goal is to build several cloaking regions at once for all pending users in U. The idea is to first build a candidate CR for the user having the largest location privacy requirement (u_l).

Data: user u, m
Result: A Cloaking Region (CR_u) for user u satisfying K_u
(1) $i \longleftarrow 0$;
(2) $C_{\max} \longleftarrow \varnothing$;
(3) **for** $i < m$ **do**
(4) $C \longleftarrow 2K_u$ cells at random with equal probability from $C(u, 2K_u)$;
(5) $C_{\max} \longleftarrow C$ only if C has the highest entropy;
(6) $i \longleftarrow i + 1$;
(7) **end**
(8) $\mathrm{CR}_u \longleftarrow$ Select K_u cells from $C_{\max}$ with a probability $\propto$ (1/cell' soccupancy);
(9) Return CR_u;

Algorithm 1: Computing Cloaking Region for a user u.

Data: set U
Result: A set of cloaking regions for every user u in U satisfying its respective K_u and θ_u
(1) $l \longleftarrow$ chooses a user with the highest K from U (denoted as K_l). If there are many of them, chooses one with the highest θ in U;
(2) $\mathrm{CR}_l \longleftarrow$ call Algorithm 1 (l, $2K_l + 1$);
(3) **repeat**
(4) **if** $\mathrm{SL}(\mathrm{CR}_l) \geq \theta_l$ **then**
(5) **for** any user u located in CR_l **do**
(6) $\mathrm{CR}_u \longleftarrow \mathrm{CR}_l$ if $K_l - \Delta \leq K_u \leq K_l$ and $\theta_u \leq \theta_l$;
(7) Remove u from U only if CR_u was set as CR_l;
(8) **end**
(9) **end**
(10) $c \longleftarrow$ from $C_N \backslash \mathrm{CR}_l$ with a probability inversely $\propto$ distance (c_l, c_u) $\times$ C_u's occupancy;
(11) $\mathrm{CR}_l \longleftarrow \mathrm{CR}_l \cup \{c\}$;
(12) **until** $U = \varnothing$ or $\mathrm{CR}_l = C_N$;

Algorithm 2: Bottom-up cloaking batching algorithm (BU).

This CR is then checked whether it needs to be extended to satisfy user u_l's location safety requirement.

Specifically, Algorithm 2 chooses first, the user l, with the highest location privacy requirement (K_l, line 1). Then, it calls the bottom-up technique algorithm to obtain a candidate CR for this chosen user. Now, it verifies whether this CR_l satisfies the safety level (location safety) demanded by *user l* (line 4). If this is the case, then it finds out what other users may share in this cloaking region (lines 5-6). Otherwise, it randomly chooses a cell having a low occupancy but a high request probability (line 10) and adds it to CR_l (line 11). Again, it verifies whether this new CR (line 11) satisfies θ_l (line 4), otherwise, another cell is chosen randomly (line 10) until $\mathrm{SL}(\mathrm{CR}_l)$ is greater or equal to θ_l (line 4). This algorithm finishes when either U becomes empty or $\mathrm{CR}_l = C_N$ (line 12).

4.2. Top-Down Technique. Our top-down technique is described by Algorithm 3. The idea of this procedure is to compute an initial CR for a chosen u and see if other users can share it. To do so, it chooses any user l in U (line 4) having the largest θ, denoted as θ_l, and sets C_N as a candidate CR_l (line 5). From now on (lines 9 to 21), it tries to reduce the size of C_l (lines 10–12) as long as the cardinality of $\mathrm{CR}_l \geq K_l$ and $\mathrm{SL}(\mathrm{CR}_l) \geq \theta_l$ (lines 6–21). For doing that, it finds out whether the removal of a randomly chosen cell c (line 11) from CR_l could achieve the lowest reduction of the entropy of CR_l (line 12). After "m attempts" (line 9), only one cell is definitely chosen and removed from CR_l (lines 18 y 19). Note that the state variable $E_{\max}$ becomes no zero (line 18) only when lines 11 and 12 are satisfied. This means there exists a candidate cell c_i to be removed (line 14). In lines 22 and 23, this technique verifies whether other users can share the same cloaking region CR_l. Finally, the algorithm stops when the first repeat-end statement finishes (lines 6 and 24). The latter happens when all pending requests have been attended successfully.

5. Results and Discussion

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

(i) *Computational Cost*. The average total amount of work (complexity time) incurred on building a set of cloaking regions.

(ii) *Size of a Cloaking Region*. The average number of cells conforming to a cloaking region. This size can be equal to or higher than the degree of location privacy protection (K) demanded by a user.

Data: set U
Result: A set of cloaking regions for every user in U
(1) $C_N \longleftarrow$ all cells in the network area;
(2) **if** $\mathrm{SF}(C_N) \geq \theta_{\max}$ **then**
(3) **repeat**
(4) $l \longleftarrow$ a user from U with the largest θ. If many, chooses the one with the largest K, (K_l);
(5) $\mathrm{CR}_l \longleftarrow C_N$;
(6) **repeat**
(7) $E_{\max} \longleftarrow 0$;
(8) $i \longleftarrow 0$;
(9) **for** $i < m$ **do**
(10) $c \longleftarrow$ from $CR_l \backslash \{c_l\}$ with a probability $\propto$ cell's occupancy;
(11) **if** $\mathrm{SL}(\mathrm{CR}_l - \{c\}) > \theta_l$ **then**
(12) **if** $E(\mathrm{CR}_l) > E_{\max}$ **then**
(13) $E_{\max} \longleftarrow E(\mathrm{CR}_l - \{c\})$;
(14) $\mathrm{CR}_t \longleftarrow \mathrm{CR}_l - \{c\}$;
(15) **end**
(16) **end**
(17) **end**
(18) **if** $E_{\max} \neq 0$ **then**
(19) $\mathrm{CR}_l \longleftarrow \mathrm{CR}_t$
(20) **end**
(21) **until** $(\#(\mathrm{CR}_l) = K_l)$ or $(E_{\max} = 0)$;
(22) Set CR_l for user l and for every other user u in CR_l having $K_l - \Delta \leq K_u \leq K_l$ and $\theta_u \leq \theta_l$;
(23) Update set U removing those users whose cloaking region is CR_l;
(24) **until** $U == \varnothing$;
(25) **end**

Algorithm 3: Top-down cloaking batching algorithm (TD).

(iii) *Number of Cloaking Regions Built*. The number of CR built by the anonymizer. The minimum value is one, because only CR can be built to protect all users at once. The maximum value corresponds to the number of users deployed in the network area, because for each of them a CR can specifically be built.

(iv) *Entropy of a Cloaking Region*. We apply equation (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.

We developed a C-based simulation, in which you can set the location cloaking technique and the network area. As a network area, we consider a medium-size city, as shown in Figure 1. We generate a network domain of 21×21 which is equally partitioned in cells of size 3×3. We disseminate a fixed number of users in this area in a range of $[200, 800]$. These users are disseminated based on three probability distributions: uniform (UNI) and two other nonuniform distributions as the exponential (EXP, 0.5) and the Zipf (ZIP, 2.0). With these two latter distributions, we want to simulate a scenario in which there are a large proportion of users and requests coming from some specific zones of the network area.

We also generate a frequency of requests for cloaking regions per cell based on the aforementioned distributions. To simplify our experiments, we use the same distribution and parameters to set both the location of the users in the network area and the frequency of requests per cell.

The value of K ranges from 2 to 12, and the value of θ ranges from 0.018 to 0.882 ($\theta_{\max}$). We are mainly interested in comparing how the anonymizer performance is impacted with the quality of the computed cloaking regions when we run our two batching techniques (denoted as BU and TD) independently and two baseline techniques that compute all CRs one by one (IND-BU and IND-TD). The first baseline approach, IND-BU, is our bottom-up approach used to compute a CR for every user from scratch, and it does not verify whether this computed CR can be assigned to other users as well. Similarly, the second baseline approach, IND-TD, is our top-down approach used to compute a CR for each user independently from scratch as well. Finally, we set $\Delta = 0$ and $m = 2 * K_u + 1$ when we run all techniques.

5.1. Effect of the Number of Users. We vary the number of users in the range of 100 to 800 users. We fixed K to 7 and θ to 0.45.

Figure 3(a) shows the computational costs incurred by all techniques. We observe that all techniques based on our *TD* approach show larger costs; because these techniques are set as an initial CR for the entire network region, they conservatively check whether it is possible to remove a cell from this region without affecting the required θ and K. We

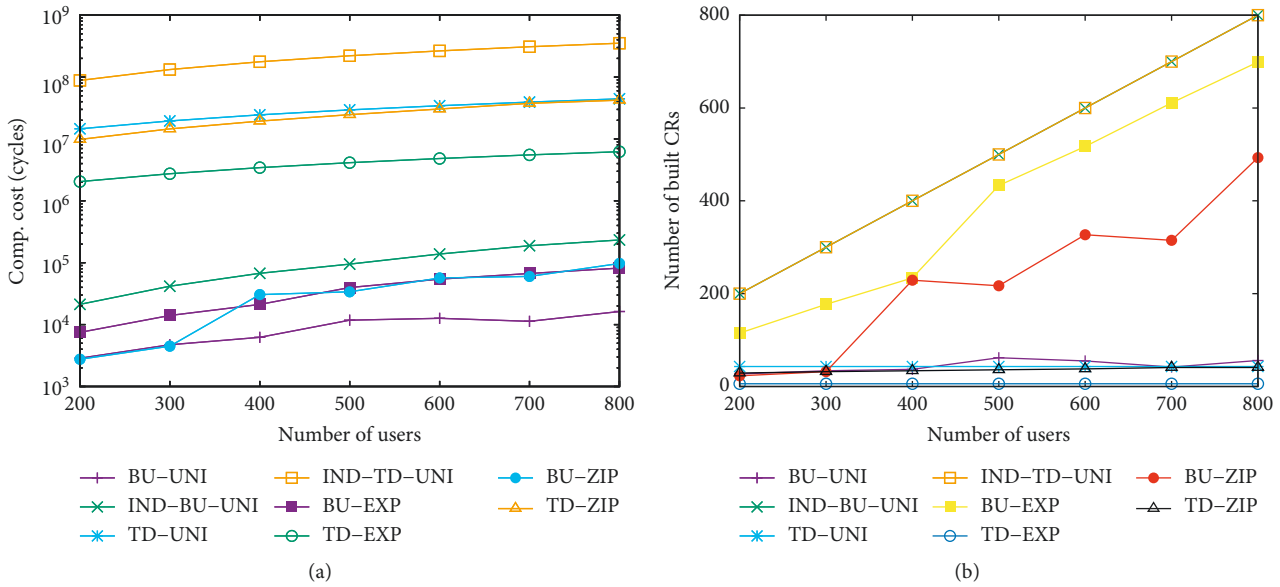

Figure 3: Effect of the number of users.

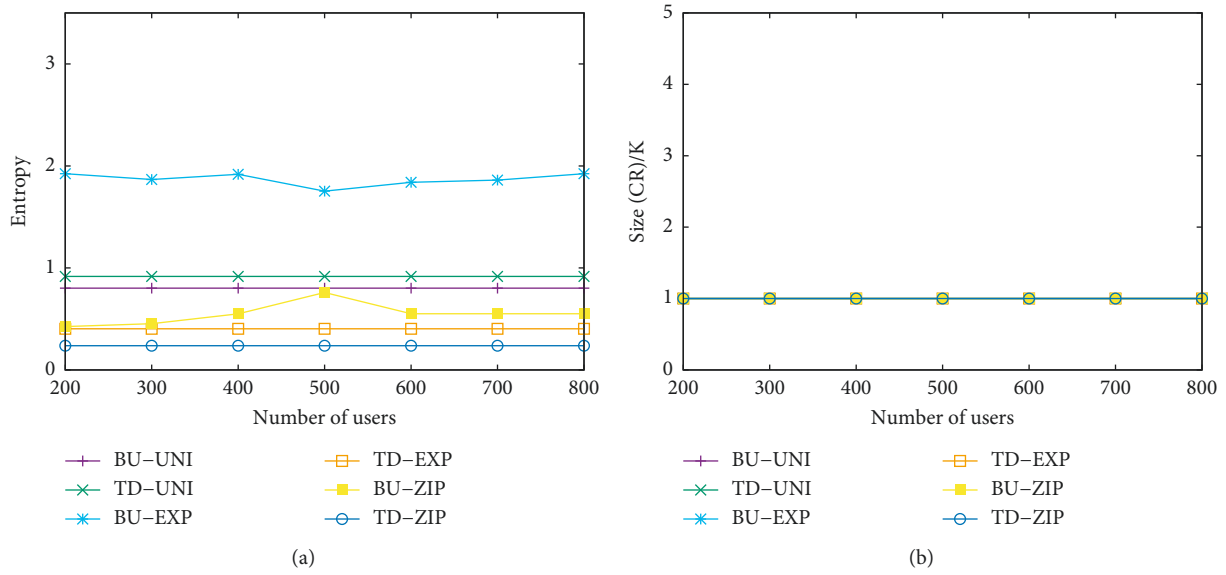

Figure 4: Effect of the number of users.

also observe that our approaches take advantage of the locality of the requests when users are preferably located in certain zones (EXP and ZIP) because they exhibit smaller costs than their similar versions running on a uniform distribution of users and requests.

Figure 3(b) shows the number of built cloaking regions by all techniques. We can observe that techniques based on TD, except IND – TD, build a smaller amount of cloaking regions. This is not surprising because TD-based approaches begin with a large cloaking region (network area) and refine this solution until it is not possible to satisfy the demanded location privacy and safety requirements.

Figure 4(a) shows the average entropy of all techniques for several distribution of users. We observe that the quality (entropy) of the cloaking regions provided by TD- and BU-based approaches is similar when the same user distribution is applied.

Figure 4(b) shows the ratio between the size of cloaking region (number of chosen cells) and the value K demanded. We observe that all techniques show similar and the best performance, which is close to 1. This is because when we consider BU-based approaches, we observe that the candidate cloaking regions returned by the bottom-up technique algorithm also satisfy the demanded location safety requirement. For TD-based approaches, the size of the initial candidate cloaking region (the entire network area) is reduced to a size equal to the location privacy requirement demanded and also satisfying the location safety required.

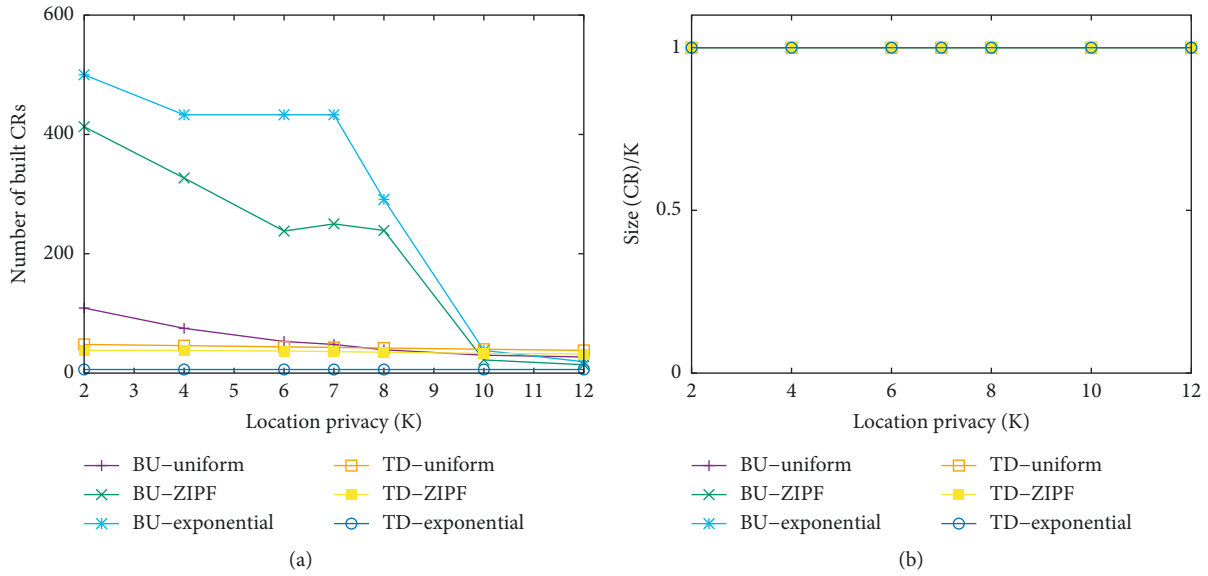

Figure 5: Effect of the location privacy (K).

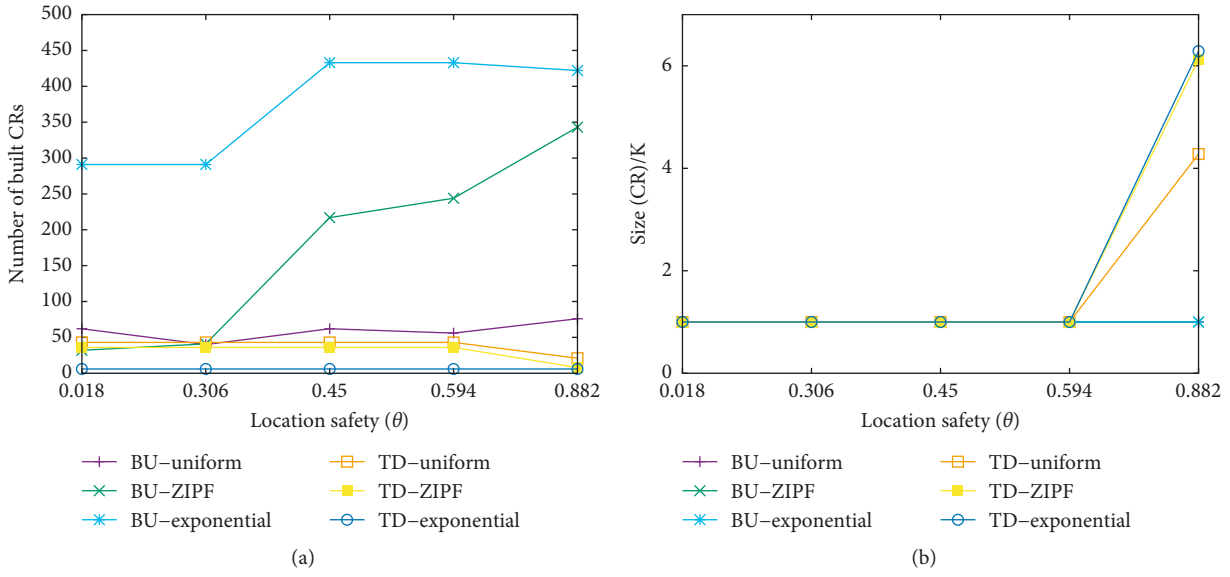

Figure 6: Effect of the location safety (θ).

5.2. Effect of the Location Privacy (K). We vary the value for K between 2 and 12. The (X, Y) coordinates of users and the frequency of the cloaking requests per cell are set according to either uniform (UNI), exponential (EXP,0.5) or Zipf (ZIP, 2.0). We fixed the location safety requirement (θ) at a value of 0.45, and the number of users is set to 500.

Figure 5(a) shows the average number of built cloaking regions. We observe that when K becomes higher, more cells are demanded, and therefore it is highly probable that the cloaking region might have a large proportion of area overlapping. As a consequence, we observe a reduced number of cloaking regions being built. Specifically, TD-based approaches exhibit the smaller amounts of cloaking regions since they initially propose the entire network area as a candidate cloaking region and they attempt to reduce its size.

Figure 5(b) shows the ratio between the size of a cloaking region and K. All techniques exhibit the best result, i.e., 1.0, which means the size the of a CR is equal to the demanded location privacy requirement (K).

5.3. Effect of the Location Safety (θ). We vary the value for θ between $0.018 = 3 \times 3/500$ and $0.882 = 21 \times 21/500$. The (X, Y)-coordinates of users and the frequency of the cloaking requests per cell are set according to either uniform (UNI), exponential (EXP, 0.5) or Zipf (ZIP, 2.0). We fixed the location privacy requirement (K) at a value of 7, and the number of users is set to 500.

Figure 6(a) shows most of the techniques based on BU build more cloaking regions when θ is increased. On the

contrary, techniques based on TD build a number of cloaking regions almost independently of θ.

Figure 6(b) shows the ratio between the size of the cloaking region and K. All techniques exhibit a value equal to one, except when θ achieves a larger value (0.882). This is because a higher θ value demands larger areas with low user occupancy.

6. Conclusions

This paper introduced two novel batching techniques to build cloaking regions for a large number of users having diverse location privacy and location safety requirements. Our proposed techniques attempt to balance computational cost of the anonymizer and the location-based service. Our techniques take advantage of building efficient cloaking regions of users having similar location privacy and safety requirements and who are located close to each other.

From the results, our techniques offer cost-effective solutions for the anonymizer side to build location privacy and safety protections. Our bottom-up approach shows a good balance between quality of a cloaking region, its size (which measures the impact at the LBS), and its computational cost for the anonymizer. Our top-down approach shows good results for the quality and the number of built cloaking regions at the expense of computational cost. This is because the latter approach is quite conservative, and there is space to make it more efficient.

Our results are preliminary yet promising. We are planning to test more diverse scenarios and to find optimal values for some system parameters such as m and Δ. In addition, we would like to extend our techniques to support continuous LBS. In this service, users periodically request location privacy and safety protection and either an LBS server or a third party adversary can attempt to correlate these cloaking regions to narrow down the location of one or many target users. Thus, the anonymizer must take into account the cloaking regions released to a user before returning a new one.

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 are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported in part by the Universidad del Bío-Bío (under grants DIUBB GI 150115/EF, DIUBB 173315 3/RS, and DIUBB 184615 1/I). We thank David Cáceres and Pablo Torres, students of the University of Bío-Bío, for performing the simulations and collecting the data presented in this article.

References

[1] G. Tobar, P. Galdames, C. Gutierrez-Soto, and P. Rodriguez-Moreno, "A batching location cloaking algorithm for location privacy protection," in *2018 Proceedings of CODASCCA*, pp. 26–36, Yerevan, Armenia, September 2018.

[2] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services. MobiSys'03*, pp. 31–42, San Francisco, CA, USA, May 2003.

[3] 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.

[4] M. Mokbel, C. Chow, and W. Aref, "The new casper: query processing for location services without compromising privacy," in *Proceedigs of ACM International Conference on Very Large Databases (VLDB'06)*, pp. 763–774, Seoul, Republic of Korea, September 2006.

[5] 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, Cambridge, UK, 2006.

[6] T. Xu 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, GIS'07*, pp. 39:1–39:8, Seattle, WA, USA, 2007.

[7] T. Xu and Y. Cai, "Exploring historical location data for anonymity preservation in location-based services," in *Proceedings of IEEE INFOCOM 2008-The 27th Conference on Computer Communications*, Hangzhou, China, April 2008.

[8] T. Xu and Y. Cai, "Feeling-based location privacy protection in location-based services," in *Proceedings of ACM International Conference on Computer and Communications Security (CCS'09)*, pp. 348–357, Chicago, IL, USA, September 2009.

[9] Q. Xie and L. Wang, "Privacy-preserving location-based service scheme for mobile sensing data," *Sensors*, vol. 16, no. 12, p. 1993, 2016.

[10] A.-D. Lahe and P. Kulkarni, "Location privacy preserving using semi-TTP server for LBS users," in *Proceedings of 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, pp. 605–610, Piscataway, NJ, USA, May 2017.

[11] H. Zhang, C. Wu, Z. Chen, Z. Liu, and Y. Zhu, "A novel online spatial-temporal k-anonymity method for location privacy protection from sequence rules-based inference attacks," *PLoS One*, vol. 12, no. 8, Article ID e0182232, 2017.

[12] C.-Y. Chow, M. Mokbel, and X. Liu, "A peer-to-peer spatial cloaking algorithm for anonymous location-based service," in *Proceedings of ACM International Siymposium on Advances in Geographic Information Systems (GIS'06)*, pp. 171–178, Arlington, VA, USA, November 2006.

[13] W.-S. Ku, R. Zimmermann, C. N. Wan, and H. Wang, "MAPLE: a mobile scalable P2P nearest neighbor query model for location-based services," in *Proceedings of the 22nd International Conference on Data Engineering (ICDE'06)*, pp. 182–222, Atlanta, GA, USA, 2006.

[14] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Achieving k-anonymity in privacy-aware location-based services," in

Proceedings of IEEE INFOCOM 2014-IEEE Conference on Computer Communications, pp. 754–762, Toronto, Canada, April 2014.

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

[16] 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.

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

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

[19] R. Gupta and U.-P. Rao, "A hybrid location privacy solution for mobile LBS," *Mobile Information Systems*, vol. 2017, Article ID 2189646, 11 pages, 2017.

[20] T. Xu and Y. Cai, "Location cloaking for safety protection of ad hoc networks," in *Proceedings of of IEEE International Conference on Computer Communications (INFOCOM'09)*, pp. 1944–1952, Rio de Janeiro, Brazil, April 2009.

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

[22] H. Kido, Y. Yanagisawa, and T. Satoh, "Protection of location privacy using dummies for location-based services," in *Proceedings of 21st International Conference on Data Engineering Workshops (ICDEW'05)*, p. 1248, Washington, DC, USA, April 2005.

[23] H. Liu, X. Li, H. Li, J. Ma, and X. Ma, "Spatiotemporal correlation-aware dummy-based privacy protection scheme for location-based services," in *Proceedings of 2017 IEEE Conference on Computer Communications, INFOCOM 2017*, pp. 1–9, Atlanta, GA, USA, May 2017.

[24] D. Ye, M. Wu, S. Tang, and R. Yu, "Scalable fog computing with service offloading in bus networks," in *Proceedings of 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud)*, pp. 247–251, Beijing, China, June 2016.

[25] B. Bamba, L. Liu, P. Pesti, and T. Wang, "Supporting anonymous location queries in mobile environments with privacygrid," in *Proceedings of the 17th International Conference on World Wide Web-WWW'08*, p. 237, ACM Press, Beijing, China, 2008.

[26] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *Proceedings of ICPS '05-International Conference on Pervasive Services*, pp. 88–97, Santorini, Greece, 2005.

[27] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Enhancing privacy through caching in location-based services," in *Proceedings of 2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 1017–1025, Kowloon, Hong Kong, April 2015.

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

[29] T. Peng, Q. Liu, and G. Wang, "Enhanced location privacy preserving scheme in location-based services," *IEEE Systems Journal*, vol. 11, no. 1, pp. 219–230, 2017.

[30] R. Schlegel, C.-Y. Chow, Q. Huang, and D. S. Wong, "User-defined privacy grid system for continuous location- based services," *IEEE Transactions on Mobile Computing*, vol. 14, no. 10, pp. 2158–2172, 2015.

[31] Y. Wang, Y. Xia, J. Hou, S.-M. Gao, X. Nie, and Q. Wang, "A fast privacy-preserving framework for continuous locationbased queries in road networks," *Journal of Network and Computer Applications*, vol. 53, pp. 57–73, 2015.