Protecting Whereabouts and Whatabouts for Check-in Based Location-Based Services

Osman Abul[*,**], **Harun Gokce**[**]

[*]Department of Computer Science, University of Sharjah, United Arab Emirates.

[**]Department of Computer Engineering, TOBB University of Economics and Technology, Turkey.

E-mail: oabul@sharjah.ac.ae, hgokce@etu.edu.tr

Received 26 August 2023; received in revised form 2 June 2024; accepted 28 July 2024

Abstract

Using diverse location-based services (LBSs), especially through mobile smart phones, have become a daily routine for many people. Location privacy is an important concern with each and every LBS request. Besides whereabouts, location disclosure provides attackers with whatabouts as well. Since each individual's location privacy needs are not the same, most solutions enable individualized location privacy profiles. In this work, as attack models and trustfulness of LBS providers are different, in the context of location check-ins, we provide a framework offering a palette of location privacy protection methods to be picked for each LBS provider/attacker. Depending on what to protect per LBS provider, i.e., whereabouts and/or whatabouts, and attack model, i.e., weak or strong, we develop six privacy protection methods. A top-down location cloaking algorithm which is able to enforce the six protection methods is presented. An extensive experimental evaluation on two real datasets are performed.

1 Introduction

Today, various location based services (LBSs) are employed for different purposes by all age groups. Navigation services and point of interest querying services are two main kinds of LBSs. Navigation services give recommendations on the shortest route to a target location, and similarly point of interests services list nearby facilities. Location check-ins are another common use of LBS services. To consult such services the users need to submit their location data. Moreover, with the high bandwidth communication capable end-devices, the users stay connected with the service and share the location data constantly over a period of time, hence resulting in disclosure of the full trajectory.

In a typical mobile LBS interaction, the smart phone companion, called user, first detects his own location and sends LBS request/query, containing the user location, through a mobile application to the LBS provider. In this request/response mechanism, the LBS provider then returns with a response. The user identities are not anonymous with subscription based LBSs. Hence, the LBS provider knows the user identity, location, time and the requested service identifier. To protect the location privacy, the canonical solution is to create cloaking regions and share coarse location information so that the precise user location cannot be re-identified by the LBS provider while the service quality is still acceptable.

In a subscription based LBS, an LBS request typically contains the following information: (i) the user id, (ii) the request time, (iii) the user location, and (iv) the service identifier with possibly additional parameters. The user is already aware that "whenabouts" (i.e., privacy of request time), "whereabouts" (i.e., privacy of place), "whyabouts" (i.e., privacy of service type), "howabouts" (i.e., privacy of user condition) and "whatabouts" (i.e., privacy of user habits) can be inferred from the request. In our framework we are only interested in "whereabouts" and "whatabouts" kinds of inferences. In the "whereabouts", the users simply do not want to be pinpointed, whereas in the "whatabouts" the users do not want to be associated with a particular sensitive location category (e.g., night clubs). Yet, sometimes "whereabouts+whatabouts" together may reveal intrusive inferences.

Suppose Alice makes a check-in at an oncology hospital in the downtown and sends an LBS request. From the request, an attacker can easily conclude about her current whereabouts (i.e., being in the oncology hospital) and whatabouts (i.e., having cancer treatment). We consider that any or both of which may be sensitive for Alice depending on her privacy requirement. We would like to note that there are correlations as well, i.e., whereabouts leak whatabouts (i.e., being in an oncology hospital is a sign of suffering from cancer) and vice versa (i.e., suffering from cancer causes visits to oncology hospital). This suggests that we need location privacy solutions preventing intrusive inferences on only whereabouts, only whatabouts and whereabouts+whatabouts together.

Depending on what the location privacy is involved with, the LBS request should be cloaked so that attackers cannot surface the sensitive information with high confidence. In this paper, we adopt spatial cloaking framework as a mean for limiting whereabouts (through providing diversity) and whatabouts (through protecting sensitivity) disclosures for various location privacy preferences/requirements. There is an inherent tradeoff between the level of the location privacy and the service quality. Since with the coarser cloaking regions, the location privacy improves but the service quality degrades, and vice versa.

A typical spatial cloaking solution (e.g., PROBE [7]) creates a cloaking map per user profile. In our framework, however, several cloaking maps per user profile can be created to address different notions of privacy requirements. Depending on the trust levels and possible attack models by the LBS providers, our framework develops six location privacy protection formulations: (i) weak whereabouts, (ii) strong whereabouts, (iii) weak whatabouts, (iv) strong whatabouts, (v) weak whereabouts+whatabouts, (vi) strong whereabouts+whatabouts. Each of them has user specific parameters to individualize the location privacy requirements and to balance the service utility/location privacy tradeoff.

The contributions of the paper are given as follows.

- Six location privacy problem formulations, in the context of location check-ins, are provided to address different needs against various trust levels and attack models of the LBS providers/attackers. All of the formulations are parametric to meet individual location privacy needs.
- Depending on the formulation picked, our framework is able to guard whereabouts, whatabouts and these two together.
- An efficient top-down space partitioning algorithm is developed. The algorithm is able to generate cloaking maps for any of the six formulations within the same framework. The algorithm exploits the monotonicity property of cloaking regions to do recursive partitioning.
- Several cloaking map quality metrics are defined and an experimental evaluation on two real-world check-in datasets has been conducted.

The paper is organized as follows. Section 2 presents the related work on location cloaking and locates our work within the literature. The framework is presented in Section 3. The framework develops six location privacy problem formulations regarding various senses of whereabouts and whatabouts. The nice notion of our framework is that these problem formulations can be uniformly solved in a top-down region refinement algorithm, which is given in Section 4. Section 5 presents an extensive experimental evaluation on two real datasets. Finally, Section 6 concludes.

2 Related Work

Privacy issues due to the disclosure of user-specific sensitive microdata can be broadly studied under two categories: (i) *offline*, and (ii) *online*. In the former, the microdata is already stored by the server and is going to be shared with the third parties. In the latter, however, the microdata is still being accumulated at the server while the user interacts with it. The user can actively control the disclosure limitation in the latter. For this reason, the user in the *online* category have the main responsibility to exploit privacy enhancing technologies. The privacy problem we studied here falls in the *online* category, but the cloaking map generation is usually an *offline* process, i.e., the cloaking map must be ready before the LBS requests begin.

2.1 Offline privacy

Sweeney [24] introduced the k-anonymity privacy principle as a mean of limiting the disclosure of sensitive information from tabular data. Typically the data is perturbed by generalization, e.g., ages to age groups. The principle ensures that each subject is indistinguishable from at least $k-1$ others, and hence any attack on record linkage (based on quasi-identifiers) cannot succeed with more than $1/k$ probability [22]. Other privacy principles like l-diversity[15] and its refinement t-closeness [14] are proposed to strengthen k-anonymity model. Our whereabouts protection exploits the k-anonymity and l-diversity principles to provide diversity of the features within the cloaking regions.

Originally developed for tabular data publishing, k-anonymity model is extended for data mining results publishing [4] and trajectory data publishing [1, 18] too. Optimal k-anonymization is proven to be NP-hard [2, 1].

2.2 Online privacy

The location protection and identity protection are the main privacy concerns of anonymous LBSs and subscription based LBSs, respectively. In the former case, the classical solution relies on the concept of *location k-anonymity* [11, 9]. Similar to classical k-anonymity, location k-anonymity requires at least $k-1$ other service requesters from the same coarse location, called mix-zone. Service handling needs to be delayed most of the time as service requests from at least k users from the same mix-zone rarely happen. The size of spatial cloaking and temporal cloaking are the two performance parameters.

In subscription based LBSs, the typical approach is to obfuscate the true location of the user by sending fake positions [3, 12, 19]. In Kido et al. [12], the user sends one or more fake positions in addition to the true location. This way the LBS provider is confused about the true location of the user, but it has the overhead of increased traffic. Moreover, the server can extract the trajectory in case the user makes multiple requests along his trajectory. In these exact location sharing (no cloaking) solutions, the locations need not to have associated semantics and are not assigned sensitivity levels. In our whatabouts formulation, however, the locations have semantics with varying degrees of sensitivities.

Private information retrieval (PIR) [10] provides strong privacy guarantees by running an encryption based protocol. However, the communication/computation overhead is very high and, more importantly, the approach is suitable only for pre-recorded static locations. The typical solution in this case is to employ a variant of k nearest neighbor (kNN) query processing, e.g., SpaceTwist [27]. Location perturbation methods based on Bayesian statistics [23] and differential privacy have been proposed too [8].

Spatial cloaking, in which exact coordinates are replaced with more coarse region identifiers, is another popular approach employed in subscription based LBSs. The spatial cloaking approach requires the area of operation to be divided into a number of uncertainty regions [5, 6, 7]. The collection of these regions is called cloaking map. The cloaking map is typically pre-computed offline based on the location privacy preferences of the user, and shared with the LBS provider. During online requests, the user reports the region identifier where his true coordinate falls in. The service quality depends on the average size of the cloaking regions. Being easily applicable, this approach should be patched when there are multiple requests from the same user over his trajectory. More concretely, the LBS provider can exploit maximum velocity to constrain the cloaking regions. The canonical solution to this case is either employing *postdating* or *time delaying* [5, 7, 26, 17] to guard against velocity attacks. In a recent study [21], it has been shown that given the location history, the current location can be estimated even if the location is obfuscated. Recently, Zheng et al. [28] proposed a semantic based obfuscation method protecting against the "whyabouts"/"howabouts" kinds of inferences. Monreale et al. [16] provides a comprehensive survey on location privacy issues and privacy-preserving strategies in mobility data publishing.

PROBE [7] like spatial cloaking approaches run in two stages: (i) offline cloaking map generation, and (ii) online location transformation which includes mechanisms to prevent against velocity attacks. The nice feature of this separation is that these stages are decoupled, i.e., the second stage can run on any cloaking map generated for different privacy needs. Indeed our current work develops location privacy solutions to generate alternative cloaking maps across several location privacy requirements. In PROBE, a user is assigned the same privacy profile across all LBSs, while in this paper, each user may have multiple privacy profiles depending on the kind of LBSs.

Our framework has three main differences from PROBE. First, PROBE's location privacy problem formulation protects from a particular attack scenario but our framework develops six targeted location privacy problem formulations to protect against a wide range of attack scenarios. Second, PROBE needs features' spatial extensions to be represented on a gridded space while our formulations are based on points. Third, PROBE operates in a bottom-up manner (i.e., uniting smaller regions), while our algorithm is top-down (i.e., dividing larger regions).

3 The Framework of Whereabouts/Whatabouts Protection

In our framework, the operating region of LBSs is fixed to a region R which is a rectangular bounded two dimensional spatial area containing a set of geo-referenced features F, e.g., check-in places. Each feature $f \in F$ is an institution/business/facility which is identified with its spatial coverage $Coverage(f) \subseteq R$. Moreover, each feature is assigned with a name (e.g., Johnson's Steak House) and a semantic annotation to express its category (e.g., restaurant).

Due to the location privacy requirements and sensitivity of some features (e.g., bars and night clubs) the users prefer their locations to be cloaked before shared with LBSs. To this end, our framework creates a set of coarse spatial cloaking regions each with a unique identifier. The set of the cloaking regions is called cloaking map CM and shared between the LBS and the user. Each user is able to locally find the cloaking region $cr \in CM$ in which the current exact location falls in, and shares the identifier of the cr with the LBS.

Since peoples' location privacy requirements are different across LBS servers and cultures, they have to be provided with a wide variety of location privacy protection solutions, the main focus in this paper. Broadly speaking, people may declare as sensitive the (i) whereabouts, (ii) whatabouts, or (iii) whereabouts+whatabouts together. In the sequel, we define parametric location privacy protection schemes for each of the three requirements.

3.1 Protecting whereabouts

Classical table anonymization involves creating equivalence groups w.r.t. quasi-identifier values. Classical k-anonymity requires that each equivalence group has at least k records. This way, the attacker cannot reidentify the correct record within the group, i.e., his success of correct record linkage is not greater than $1/k$. However, the sensitive values within an equivalence group need not to be distinct/diverse. Classical l-diversity principle insists that sensitive values in every group is diverse enough so that each sensitive value is "well-represented". The l-diversity simply ensures that none of the sensitive values can be assigned to a screened participant with high probability.

By an analogy, we can define k-anonymous and l-diverse cloaking regions as given next.

Definition 1 (k-anonymous location cloaking region/map)**.** A cloaking region cr is k-anonymous if it contains at least k features. A cloaking map CM is k-anonymous if every cloaking region $cr \in CM$ is k-anonymous.

The definition simply states that each cloaking region $cr \in CM$ (usually a rectangular region) contains at least k number of features (check-in places) so that when the respective identifier is shared with LBS provider, the user's check-in location is at least k anonymous within the respective cloaking region. There are certain efficiency (e.g., runtime) and effectiveness (e.g., mean spatial coverage of all cloaking regions) metrics to cloaking map generation process and its results.

Problem 1 (k-anonymous location cloaking region privacy)**.** Given a set of geo-referenced feature set F and anonymity level k, find a cloaking map CM which is k-anonymous.

Suppose we have k-anonymous cloaking map CM and the user makes an LBS request at a $cr \in CM$. Since the cr contains at least k features, the attacker cannot identify the true source of the LBS request with probability more than $1/k$.

Although Definition 1 is conceptually very simple, we identify a serious drawback that the source of the request within the cr need not to be equally likely, e.g., one of the features may be highly frequented while the others are quite rarely. In such cases, we talk about pseudo k-anonymous cloaking regions and cloaking map. To this end, we need relative and normalized likelihoods of being frequented for features within the same cloaking region. Indeed, this defines a probability distribution within each cloaking region. On the other hand, Problem 1 is still relevant in case no probability distribution of the features are available apriori or the features are equally likely frequented.

Given a prior probability distribution P (which can be obtained from the frequencies of previous check-ins) over all features F (a set of check-in places), and the reported cloaking region cr in the LBS request, one can easily compute the posterior probability distribution of features $P(f|cr).\forall f \in cr$ since the equation $\sum_{f \in cr} P(f|cr) = 1$ holds by definition (as each location check-in is done at one of the places in F). As the cr is explicit in the LBS request then any feature out of the cr has probability zero, i.e., $P(f|cr) = 0.\forall f \notin cr$. Definition 2, which we call (entropy) l-diverse cloaking region, is the strong notion (i.e., employing the P) of the anonymity defined in Definition 1.

Definition 2 (l-diverse location cloaking region/map)**.** A cloaking region cr is l-diverse if $H(cr) = \sum_{f \in cr} -P(f|cr)\log(P(f|cr)) \geq \log(l)$ holds, where $H(cr)$ denotes the entropy of cr. A cloaking map CM is l-diverse if every $cr \in CM$ is l-diverse.

Problem 2 (l-diverse location cloaking privacy)**.** Given a set of geo-referenced feature set F, prior probability distribution P and diversity l, find a cloaking map CM which is l-diverse.

Theorem 1 (monotonicity of k-anonymity and l-diversity)**.** *1. Let $cr1 \in CM$ and $cr2 \in CM$ be k-anonymous cloaking regions, then $cr1 \bigcup cr2$ is k-anonymous too.*

2. Let $cr1 \in CM$ and $cr2 \in CM$ be l-diverse cloaking regions, then $cr1 \bigcup cr2$ is l-diverse too.

Proof.

1. *$|cr1| \geq k$ and $|cr2| \geq k$ together implies that $|cr1 \bigcup cr2| \geq k$.*

2. *This is simply due to the concavity of the entropy function. More formally, let α (resp. $1-\alpha$) be the probability of drawing a feature from $cr1$ (resp. $cr2$). Then, the two cloaking regions act as if they are disjoint mixtures. Hence, $H(cr1 \bigcup cr2) = \alpha H(cr1) + (1-\alpha)H(cr2) + H(\alpha)$ [20]. Clearly, for any α, $H(cr1 \bigcup cr2) \geq \log(l)$ provided that $H(cr1) \geq \log(l)$ and $H(cr2) \geq \log(l)$.*

The monotonicity property is an important property that enables us to stop looking solutions with finer granularity when a candidate cloaking is not safe. Since, all the of the six problem definitions are handled within the same framework we show this property for each of the problem definitions.

3.2 Protecting whatabouts

Suppose a cloaking region *cr* is diverse according to Definition 2. Unfortunately, diversity does not necessarily imply that an LBS request from the *cr* is safe. Just consider, for instance that, all of the features within the *cr* are sensitive for the user. Although the diversity is ensured, being at whichever feature within the *cr* is immaterial as far as the sensitivity is concerned. On the other hand, the same *cr* may be quite insensitive for another user as his sensitivity requirement might be different. This suggests that though cloaking region diversity is not user specific whereas the cloaking region sensitivity is. This in turn suggests that each user should have own sensitivity profile. Moreover, the features may have differing level of sensitivities.

To capture the sensitivity of the features, we define the sensitivity function S, i.e., $S : F \rightarrow [0..1]$, which assigns a sensitivity value between $[0..1]$ for each feature from the feature set F. Features with sensitivity values closer to 1 are very sensitive while the values closer to 0 are less sensitive. Indeed the function S is part of the user's feature sensitivity profile. Total, average and expected sensitivity of a cloaking region *cr* are defined next.

Definition 3 (Sensitivity of cloaking region)**.** Total sensitivity of cloaking region *cr* is defined as $TS_{cr} = \sum_{f \in cr} S(f)$, its average sensitivity as $AS_{cr} = \frac{TS_{cr}}{|cr|}$, and its expected sensitivity as $ES_{cr} = \sum_{f \in cr} P(f|cr)S(f)$.

The sensitivity profile of a user is a tuple $PP =< S, \tau >$ where S is a user specific sensitivity function and τ is the sensitivity threshold for maximum value for average/expected sensitivity of any cloaking region *cr*. For a given *PP*, the safe cloaking region and the insensitive cloaking map are defined as follows.

Definition 4 (τ-insensitive cloaking region/map)**.** A cloaking region *cr* is average τ-insensitive if $AS_{cr} \leq \tau$. Similarly, a cloaking region *cr* is expected τ-insensitive if $ES_{cr} \leq \tau$. A cloaking map *CM* is average (resp. expected) τ-insensitive if every cloaking region $cr \in CM$ is average (resp. expected) insensitive.

Problem 3 (average (τ)-safe location cloaking privacy)**.** Given a set of geo-referenced feature set F and user-specific privacy profile $PP =< S, \tau >$, find a cloaking map CM where for each $cr \in CM$ it holds that $AS_{cr} \leq \tau$.

Problem 4 (expected (τ)-safe location cloaking privacy)**.** Given a set of geo-referenced feature set F, prior probability distribution P and user-specific privacy profile $PP =< S, \tau >$, find a cloaking map CM where for each $cr \in CM$ it holds that $ES_{cr} \leq \tau$.

Problem 3 defines a weak protection scheme as it does not use the prior probability distribution P. On the other hand, Problem 4 defines a strong protection scheme as it utilizes P when it is available. Note that Problem 3 is relevant in the absence of P.

Theorem 2 (monotonicity of average/expected (τ)-safe location cloaking)**.** *1. Let $cr1 \in CM$ and $cr2 \in CM$ be average (τ)-safe cloaking regions, then $cr1 \bigcup cr2$ is average (τ)-safe cloaking region too.*

2. Let $cr1 \in CM$ and $cr2 \in CM$ be expected (τ)-safe cloaking regions, then $cr1 \bigcup cr2$ is expected (τ)-safe cloaking region too.

Proof.

1. *$AS_{cr1} \leq \tau$ and $AS_{cr2} \leq \tau$ together implies, by a simple arithmetic, that $AS_{cr1 \bigcup cr2} = \frac{TS_{cr1 \bigcup cr2}}{|cr1|+|cr2|} \leq \tau$.*

2. *Suppose, after the union, all the probabilities in $cr1$ and $cr2$ are scaled by $0 < \alpha < 1$ and $0 < 1-\alpha < 1$, respectively. Then, $ES_{cr1 \bigcup cr2} = \alpha \sum_{f \in cr1} P(f|cr1)S(f) + (1-\alpha) \sum_{f \in cr2} P(f|cr2)S(f) \leq \alpha\tau + (1-\alpha)\tau \leq \tau$.*

3.3 Protecting whereabouts and whatabouts

We would like to note that whereabouts protection (studied in Problem 1 and Problem 2) does not ensure whatabouts protection (studied in Problem 3 and Problem 4), and vice versa. Indeed, they solve different location privacy issues as their problem formulations are quite different and are not convertible. Thus section provides new problem definitions aimed at solving whereabouts and whatabouts protections simultaneously. Monotonicity property is shown to be maintained so that all of the six problem formulations are solvable in a unified framework.

Definition 5 ((k, τ)-safe cloaking region/map)**.** A cloaking region cr is (k, τ)-safe if it is (i) k-anonymous according to Definition 1 and (ii) average τ-insensitive according to Definition 4. A cloaking map CM is (k, τ)-safe if every cloaking region $cr \in CM$ is (k, τ)-safe.

Problem 5 ((k, τ)-safe location cloaking privacy)**.** Given a set of geo-referenced feature set F, anonymity k and user-specific privacy profile $PP =< S, \tau >$, find a cloaking map CM which is (k, τ)-safe.

Strong notion (i.e., employing probability distribution P) of Definition 5 is provided in Definition 6.

Definition 6 ((l, τ)-safe cloaking region/map)**.** A cloaking region cr is (l, τ)-safe if it is (i) l-diverse according to Definition 2 and (ii) expected τ-insensitive according to Definition 4. A cloaking map CM is (l, τ)-safe if every cloaking region $cr \in CM$ is (l, τ)-safe.

Problem 6 ((l, τ)-safe location cloaking privacy)**.** Given a set of geo-referenced feature set F, prior probability distribution P, diversity l and user-specific privacy profile $PP =< S, \tau >$, find a cloaking map CM which is (l, τ)-safe.

Theorem 3 (monotonicity of (k,τ)-safe and (l,τ)-safe location cloaking)**.** *1. Let $cr1 \in CM$ and $cr2 \in CM$ be (k,τ)-safe cloaking regions, then $cr1 \bigcup cr2$ is (k,τ)-safe cloaking region too.*

2. Let $cr1 \in CM$ and $cr2 \in CM$ be (l,τ)-safe cloaking regions, then $cr1 \bigcup cr2$ is (l,τ)-safe cloaking region too.

Proof.

1. *This follows from Theorems 1.1 and 2.1.*

2. *This follows from Theorems 1.2 and 2.2.*

3.4 Optimal Cloaking

Problems 1 through 6 did not define the optimization objective but rather what is a safe cloaking map. However, trivial but useless solutions may exist meeting the objective unless no optimization objective is provided. Fortunately, it is very intuitive to define the optimal cloaking: the smaller the coverage of cloaking regions the better it is. The following problem definition formalizes this notion.

Problem 7 (Optimal location cloaking privacy)**.** Given one of the cloaking privacy problem formulations (i.e., Problem 1 through 6) with respective parameters, find a cloaking map *CM* so that,

- *CM* meets the location cloaking privacy definition of the problem, and
- $Max\{Coverage(cr) : cr \in CM\}$ is minimized.

Since minimum *k*-clustering, the relaxed version of Problem 1 which is being the simplest of the six, is NP-Hard [1] we develop an efficient polynomial solution in the next section.

4 A Top-Down Cloaking Map Generation

The monotonicity properties as shown in Theorems 1, 2 and 3 enable us to use a top-down region division approach to solve all of the six problem formulations within the same framework.

Our cloaking map generation consists of two stages: (i) intra-feature cloaking and (ii) inter-feature cloaking. In the former, we simply calculate the centroid point of the each feature's spatial coverage while in the latter we divide *R* into a number of cloaking regions by recursive partitioning. Note that the former stage is skipped in case the feature locations are already given as points, i.e., longitude-latitude pairs, or check-in places.

4.1 Inter-feature cloaking

Our inter-feature cloaking method partitions the region *R*. Starting from the whole region, the method evaluates the safety property against the specified problem definition. If it is found safe, then it is divided into two and each part is checked for safety again. We keep recursive partitioning as far as the safety property is satisfied, and stop when the safety property is violated. At each step we consider both equal area vertical and horizontal partitionings and pick the best promising one according to an evaluation function.

Algorithm 1 sketches the progress of the method. The inputs are the privacy profile (problem definition and its parameters) *PP* and the cloaking region *R*. *PP* is either Problem 1 through Problem

[1] `https://www.nada.kth.se/~viggo/wwwcompendium/node129.html#5750`

6 and its related parameters. For instance, in case it is Problem 2 then its parameters include the geo-referenced feature set F, prior probability distribution P and diversity preference l. R and the other regions (obtained through partitioning) are rectangles represented with the upper-left and lower-right corner points. The function $isSafe(r, PP)$ checks the safety property of the region r w.r.t. the privacy profile PP. In case region r is empty (i.e., with no feature within it) the function $isSafe(r, PP)$ always returns true as any region with no feature is not private.

A region r is called *degenerate* if it contains zero or one feature. The function $isDegenerate(r)$ returns true if it is *degenerate*. Clearly there is no point to further partition a degenerate region. Moreover, for most practical purposes, there is no point to shrink a cloaking region if it is already very small. Just consider a cloaking region with dimensions of $100m \times 100m$, then for most people there is no point for further partitioning it into two half size cloaking regions. To address this fact, the function $isSmallEnough(r, rst)$ returns true if the spatial area of r is not greater than region size threshold rst, a user parameter. Hence this way, for small enough cloaking regions, we do not consider further partitioning.

The algorithm considers both of the equal area horizontal and vertical partitionings. Let $rh1, rh2$ (resp. $rv1, rv2$) be the result of the candidate horizontal (resp. vertical) partitioning of the region r. The function $PreferredPartition(rh1, rh2, rv1, rv2, PP)$ returns the result of either horizontal partitioning or vertical partitioning. Algorithm 2 implements this function. In case only one of the partitioning is safe, then it returns the respective partitioning result. In case both of them are safe, it returns the more compact one. The compactness is measured as the diagonal length (i.e., maximum walk distance) of the resulting region.

The resulting cloaking map CM from Algorithm 1 contains two kinds of regions: (i) degenerate region set CM_d, and (ii) non-degenerate (true cloaking) region set CM_{nd}. The latter regions (added at lines 9 and 27) are indeed the true cloaking regions in the sense that any LBS request within this region is cloaked with the region identifier. On the other hand, the former regions (added at lines 17 and 22) indeed are not cloaking regions as any LBS request from the respective region need not to be cloaked, i.e., the precise point can be used at the LBS request.

4.2 Complexity and improvement

Let n be the number of features in R and m be the size of CM. The total number of regions generated and safety check performed is $O(m)$. This is because region cutting generates a binary tree where the leaf node set is CM. So, the number of the total nodes in the tree is not more than $2m$. For each tree node we consider vertical and horizontal cuts and take one of them. As a result, the total regions generated and the safety check performed are not more than $4m$. The complexity of each invocation of $PreferredPartition$ is $O(1)$, and for $O(m)$ invocation it has a total complexity of $O(m)$. In the straightforward implementation of Algorithm 1, each invocation of $isSafe$ and $isDegenerate$ have the worst-case complexity of $O(n)$ as they need all the features within a region to be identified. In total, the algorithm runs in $O(nm)$ time.

The total complexity can be improved by using a spatial index on the features. For instance, using kd-trees, the index can be constructed in $O(nlogn)$ and each range query can be answered in $O(\sqrt{n})$ time [13]. With spatial indexing, the total complexity of the algorithm then becomes $O(nlogn + m\sqrt{n})$. In our implementation, however, we assign the features to one of the partitions at each step and scan only features within the partition for next partitioning. This is a kind of binary search and hence achieves $O(n + mlogn)$ complexity.

```
Require: Privacy profile $PP$, operating region $R$, region size threshold $rst$
Ensure: Cloaking map $CM$
$CM_d \leftarrow \emptyset$
$CM_{nd} \leftarrow \emptyset$
$OpenQueue \leftarrow \emptyset$
$OpenQueue.Enqueue(R)$
while $!OpenQueue.isEmpty()$ do
    $r \leftarrow OpenQueue.Dequeue()$
    if $isSafe(r, PP)$ then
        if $isSmallEnough(r, rst)$ then
            $CM_{nd} \leftarrow CM_{nd} \bigcup \{r\}$
            continue
        end if
        $(rh1, rh2) \leftarrow HorizontalPartition(r)$
        $(rv1, rv2) \leftarrow VerticalPartition(r)$
        $(r1, r2) \leftarrow PreferredPartition(rh1, rh2, rv1, rv2, PP)$
        if $isSafe(r1, PP) \wedge isSafe(r2, PP)$ then
            if $isDegenerate(r1)$ then
                $CM_d \leftarrow CM_d \bigcup \{r1\}$
            else
                $OpenQueue.Enqueue(r1)$
            end if
            if $isDegenerate(r2)$ then
                $CM_d \leftarrow CM_d \bigcup \{r2\}$
            else
                $OpenQueue.Enqueue(r2)$
            end if
        else
            $CM_{nd} \leftarrow CM_{nd} \bigcup \{r\}$
        end if
    end if
end while
$CM \leftarrow CM_d \bigcup CM_{nd}$
return $CM$
```

Algorithm 1: Top-down Inter-Feature Cloaking

```
Require: Horizontal cut $(rh1, rh2)$ and vertical cut $(rv1, rv2)$ partitionings, Privacy profile $PP$
Ensure: The pair $(rh1, rh2)$ or $(rv1, rv2)$
if $!(isSafe(rv1, PP) \wedge isSafe(rv2, PP))$ then
    return $(rh1, rh2)$
else if $!(isSafe(rh1, PP) \wedge isSafe(rh2, PP))$ then
    return $(rv1, rv2)$
else
    if $Diagonal(rh1) < Diagonal(rv1)$ then
        return $(rh1, rh2)$
    else
        return $(rv1, rv2)$
    end if
end if
```

Algorithm 2: Function *PreferredPartition*

5 Experimental Evaluation

All of the experiments are done on a dual core laptop computer (2.33 GHz with total 8 GB of RAM) running Windows 10. The algorithms are implemented in Java.

5.1 Datasets

We experimented with two Foursquare check-in datasets: NYC and TKY [25]. The former contains 227.428 check-ins in New York City and the latter contains 573.703 check-ins in Tokyo. Each check-in record contains anonymized user id, venue id, venue category with category name, venue location (latitude and longitude) and check-in time. The following table gives some statistics from the datasets.

Dataset	# of users	# of venues (features)	# of venue categories (feature types)	# of check-ins
NYC	824	38.336	417	227.428
TKY	1.939	61.858	417	573.703

We used the number of check-ins in a venue as the indicator of respective popularity and obtained the probability distribution P accordingly. Assigning sensitivity function S to features is a subjective matter. In our scheme, we manually assigned the sensitivity values for each category according to our sense of location privacy. The sensitivity of the individual features are assigned the sensitivity value of the respective category.

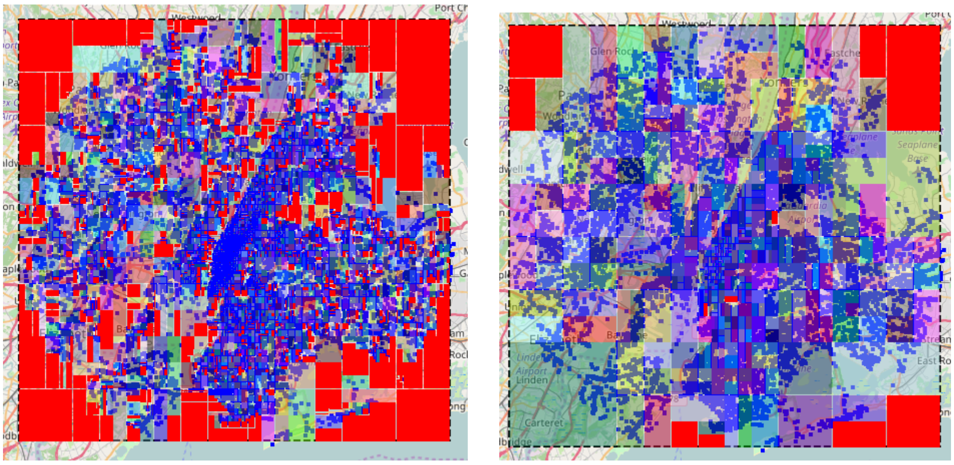
5.2 Performance metrics

We measure the algorithm's runtime as an efficiency metric and develop several effectiveness metrics to measure the quality of the resulting cloaking map. We distinguish between the degenerate regions CM_d and non-degenerate (true cloaking) regions CM_{nd}. Then, the cloaking map CM is union of CM_d and CM_{nd}, i.e., $CM = CM_d \bigcup CM_{nd}$. The effectiveness metrics are as follows:

- Cloaking ratio (CR): $CR(CM) = \frac{\sum_{r \in CM_{nd}} Coverage(r)}{\sum_{r \in CM} Coverage(r)}$, where $Coverage(r)$ is the area of r.
- Cloaking region count: It is simply the $|CM_{nd}|$.
- Mean feature count (MeanFC): $MeanFC(CM) = Avg\{Nbrfeatures(r) : r \in CM_{nd}\}$, where $Nbrfeatures(r)$ gives the number of features within r.
- Mean spatial coverage (MeanSC): $MeanSC(CM) = Avg\{Coverage(r) : r \in CM_{nd}\}$, where $Coverage(r)$ is the area of r.
- Mean spatial diameter (MeanSD): $MeanSD(CM) = Avg\{Diameter(r) : r \in CM_{nd}\}$, where $Diameter(r)$ is the length of r's diagonal.

5.3 Results

To see how a sample cloaking map visually looks like we give the resulting cloaking maps for Problem 1 at two extremes of k for $rst = 10000m^2$ in Figure 1. The red color shows the degenerated regions and the other colors show the true cloaking regions.



a) $k = 5$ b) $k = 50$

Figure 1: Cloaking maps due to Problem 1 with $rst = 10000m^2$ on NYC

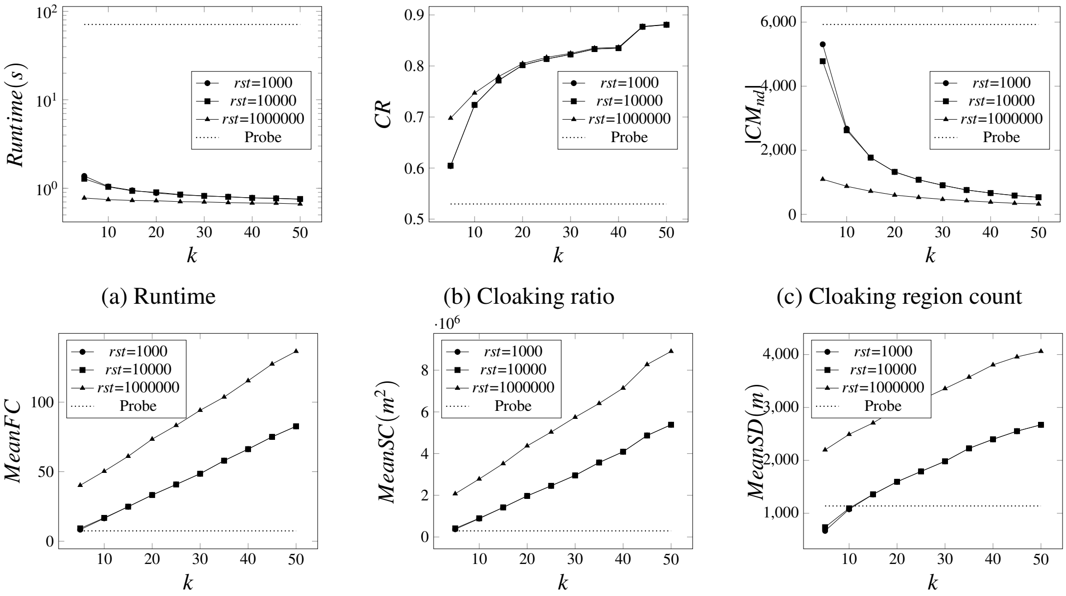
PROBE is the closest proposal to our approach. But PROBE's location privacy definition is quite different from all of our six problem definitions. Hence, the experimental comparison is not fair due to the different problem formulations. Nevertheless, we provided PROBE results in some figures for reference only, i.e., not for comparison. We used $Sens_{Reg}$ variant of PROBE as it is the only variant which outputs rectangular cloaking regions. Since PROBE is a bottom-up algorithm it needs a user-specified gridding. For NYC, we picked 512x512 gridding which ended up roughly 80mx120m size for each grid cell. For TKY we tried with several gridding ranging from 256x256 to 2048x2048, but for each case we ended up with a single connected cloaking region covering the extension of the whole dataset. This is because, the TKY dataset is too dense and our sensitivity levels are relatively high. Therefore, we are unable to show the reference PROBE results for TKY dataset.

PROBE operates by enlarging cloaking regions by vertically or horizontally in one of four dimensions of North, South, West and East. The stopping criteria is arrived when the cloaking region is large enough to provide enough sensitivity according the location privacy preference. In all of our experiments we used the default values (0.5) for its sensitivity arrays. As a result, PROBE performances are straight lines as it does not accept k and l as the parameter.

5.3.1 Whereabouts protection (Problem 1 and Problem 2):

The overall effectiveness and efficiency results of the privacy definition given in Problem 1 on NYC (resp. TKY) are shown in Figure 2 (resp. in Figure 3). The results are provided for three different rst thresholds of 1000, 10000 and 10000000 m^2s. The cloaking ratio increases with increasing k at all rst values. Indeed this is evident on the maps as shown in Figure 1. The number of cloaking regions and runtime tend to decrease with increasing k at all rst values. This is mostly because the algorithm terminates earlier with larger k. All the results agree with the expectation that the larger the k the more coarse the cloaking regions and hence bigger values for mean metrics as shown in the sub-figures (d), (e) and (f) of Figures 2 and 3 .

The overall effectiveness and efficiency results of the privacy definition given in Problem 2 on NYC (resp. TKY) are shown in Figure 4 (resp. in Figure 5). The results are provided for three different rst thresholds of 1000, 10000 and 10000000 m^2s. The cloaking ratio increases with increasing l at all rst values. The number of cloaking regions and runtime tend to decrease with increasing l at



(a) Runtime

(b) Cloaking ratio

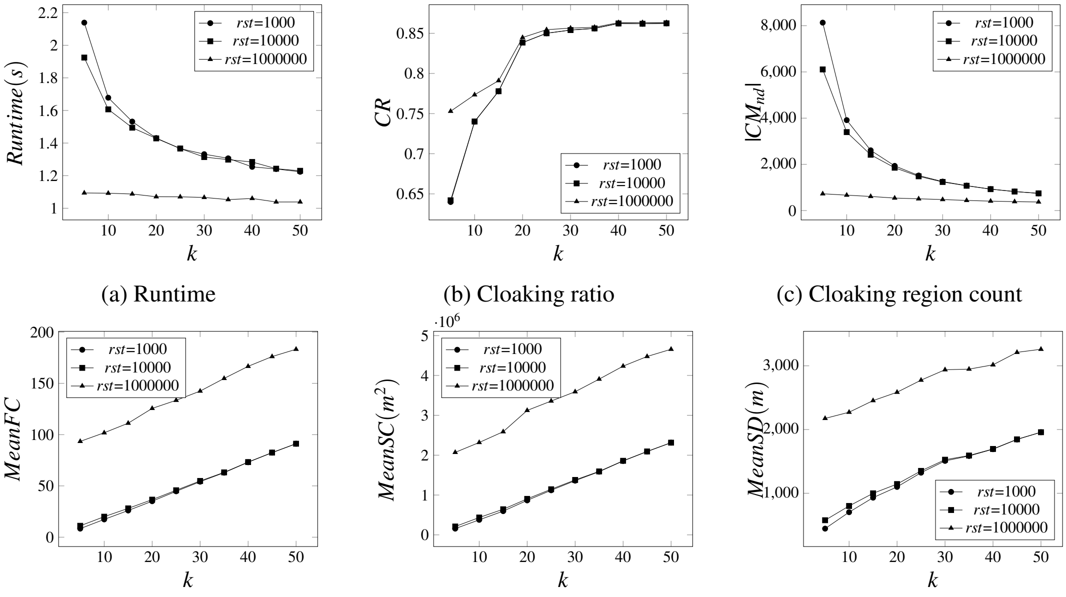
(c) Cloaking region count

(d) Mean feature count

(e) Mean spatial coverage

(f) Mean spatial diameter

Figure 2: Performance metrics of Problem 1 on NYC



(a) Runtime

(b) Cloaking ratio

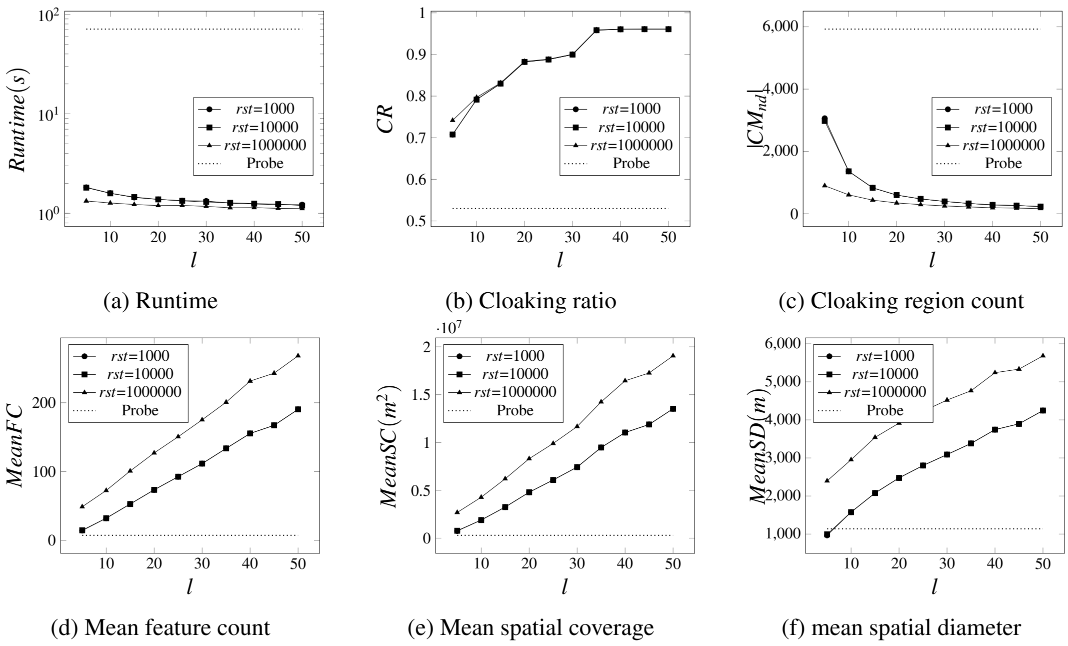
(c) Cloaking region count

(d) Mean feature count

(e) Mean spatial coverage

(f) Mean spatial diameter

Figure 3: Performance metrics of Problem 1 on TKY



(a) Runtime

(b) Cloaking ratio

(c) Cloaking region count

(d) Mean feature count

(e) Mean spatial coverage

(f) mean spatial diameter

Figure 4: Performance metrics of Problem 2 on NYC

all *rst* values. This is mostly because the algorithm terminates earlier with larger l. All the results agree with the expectation that the larger the l the more coarse the cloaking regions and hence bigger values for mean metrics as shown in the sub-figures (d), (e) and (f) of Figures 4 and 5.

Comparing the performance metrics of the resulting cloaking maps shows that the size of the cloaking regions are larger, as expected, with the strong version (Problem 2) in comparison to the results of the weak version (Problem 1) for both of the datasets. Similarly for both of the problems and for both of the datasets, bigger *rst*s result in coarser cloaking regions.

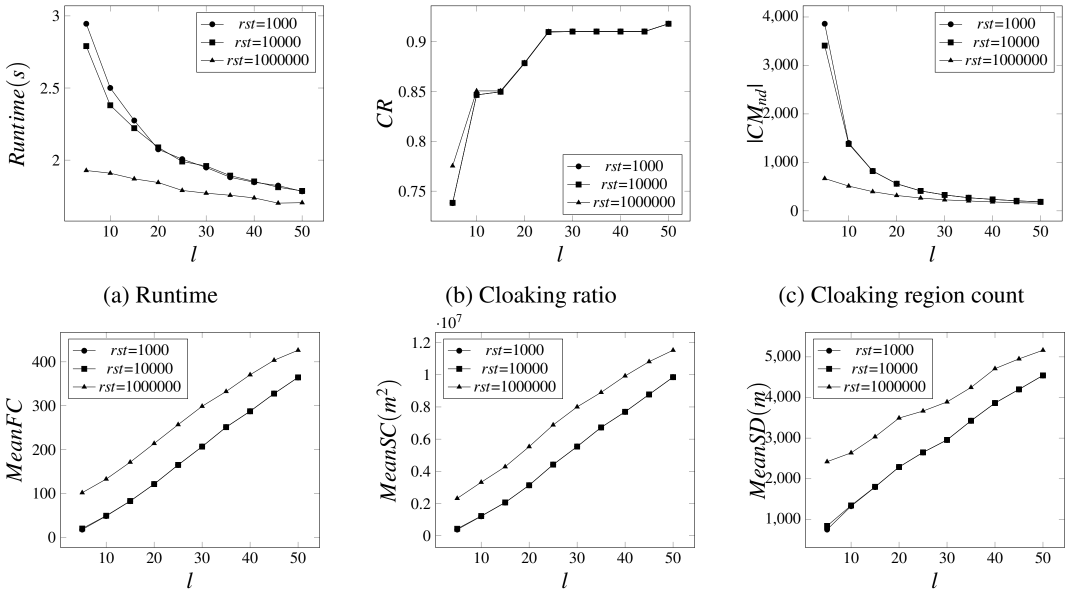
5.3.2 Whatabouts protection (Problem 3 and Problem 4):

The overall effectiveness and efficiency results of the privacy definition given in Problem 3 on NYC (resp. TKY) are shown in Figure 6 (resp. in Figure 7). The results are provided for three different *rst* thresholds of 1000, 10000 and 10000000 m^2s. The cloaking ratio decreases with increasing τ at all *rst* values. The number of cloaking regions and runtime tend to increase with increasing τ at all *rst* values. This is mostly because the algorithm terminates earlier with larger τ. All the results agree with the expectation that larger τ values result in fine grained cloaking regions, and hence gives decreasing value trends for mean metrics as shown in the sub-figures (d), (e) and (f) of Figures 6 and 7. We observe similar trends from Figures 8 and 9 for Problem 4 on both of the datasets.

Comparing the performance metrics of the resulting cloaking maps shows that the size of the cloaking regions are larger, as expected, with the strong version (Problem 4) in comparison to the results of the weak version (Problem 3) for NYC.

5.3.3 Whereabouts+Whatabouts protection (Problem 5 and Problem 6):

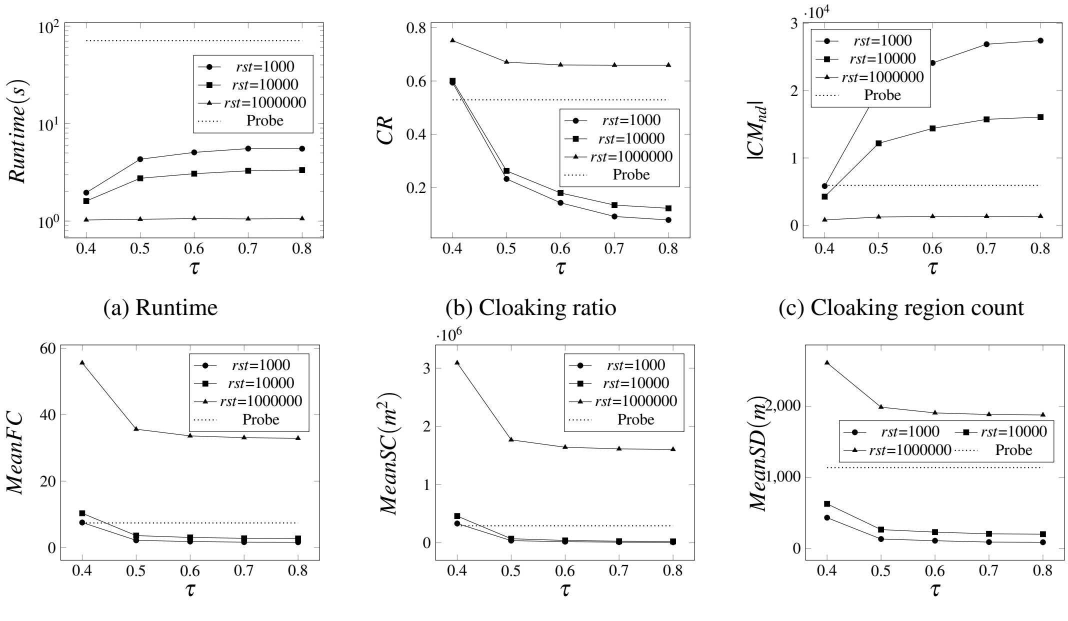
The overall effectiveness and efficiency results of the privacy definition given in Problem 5 on NYC (resp. TKY) are shown in Figure 10 (resp. in Figure 11). The results are provided for the *rst* threshold



(a) Runtime (b) Cloaking ratio (c) Cloaking region count

(d) Mean feature count (e) Mean spatial coverage (f) Mean spatial diameter

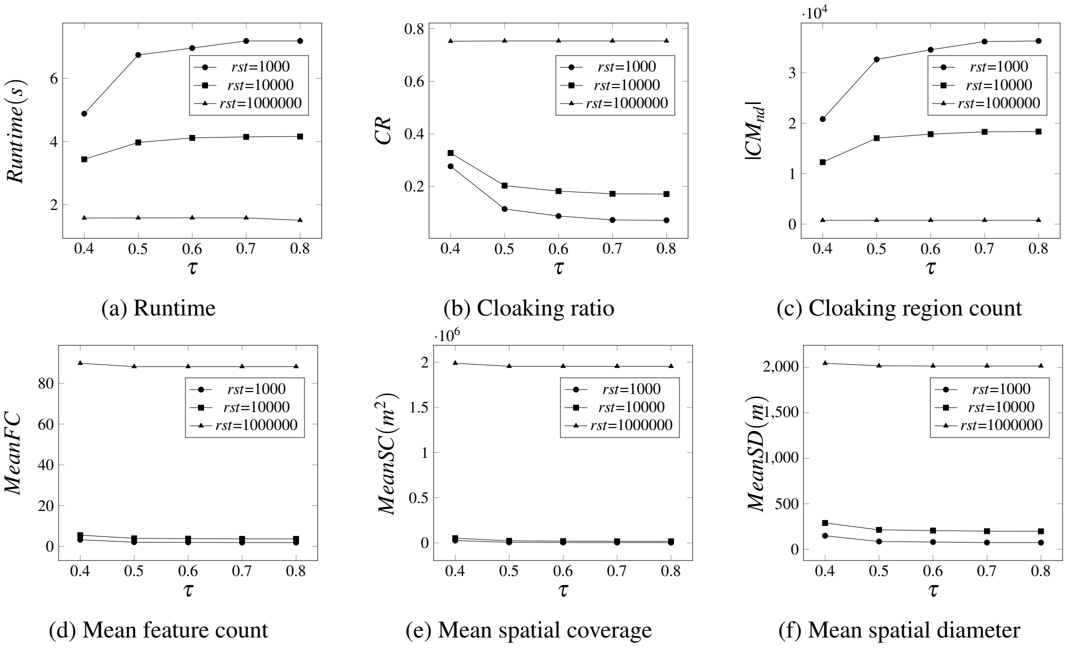
Figure 5: Performance metrics of Problem 2 on `TKY`



(a) Runtime (b) Cloaking ratio (c) Cloaking region count

(d) Mean feature count (e) Mean spatial coverage (f) Mean spatial diameter

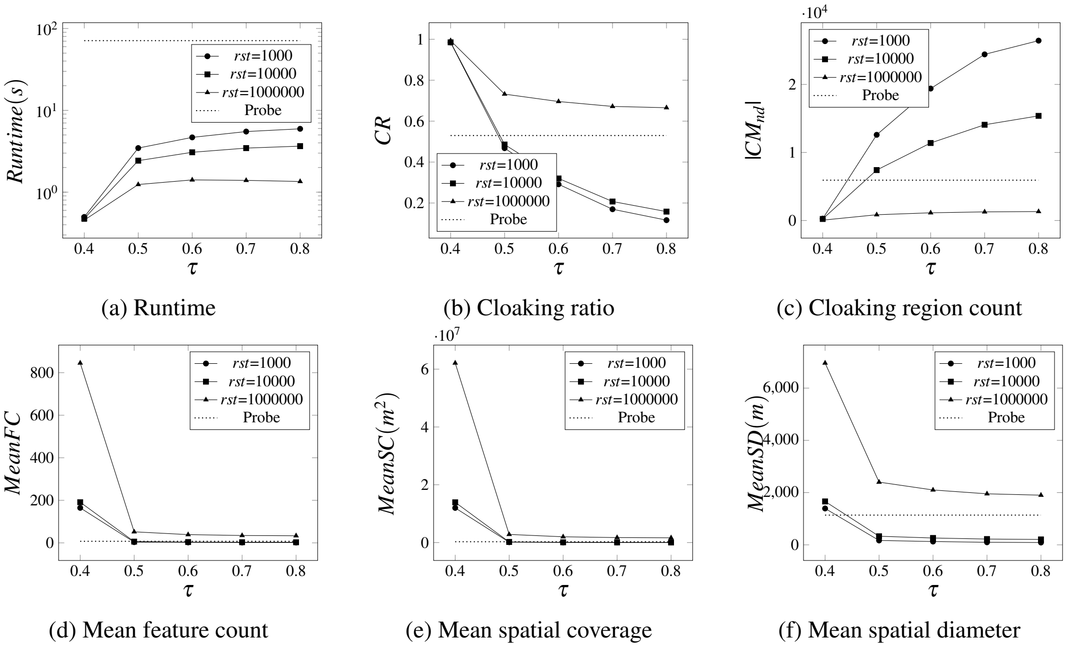
Figure 6: Performance metrics of Problem 3 on `NYC`



(a) Runtime (b) Cloaking ratio (c) Cloaking region count

(d) Mean feature count (e) Mean spatial coverage (f) Mean spatial diameter

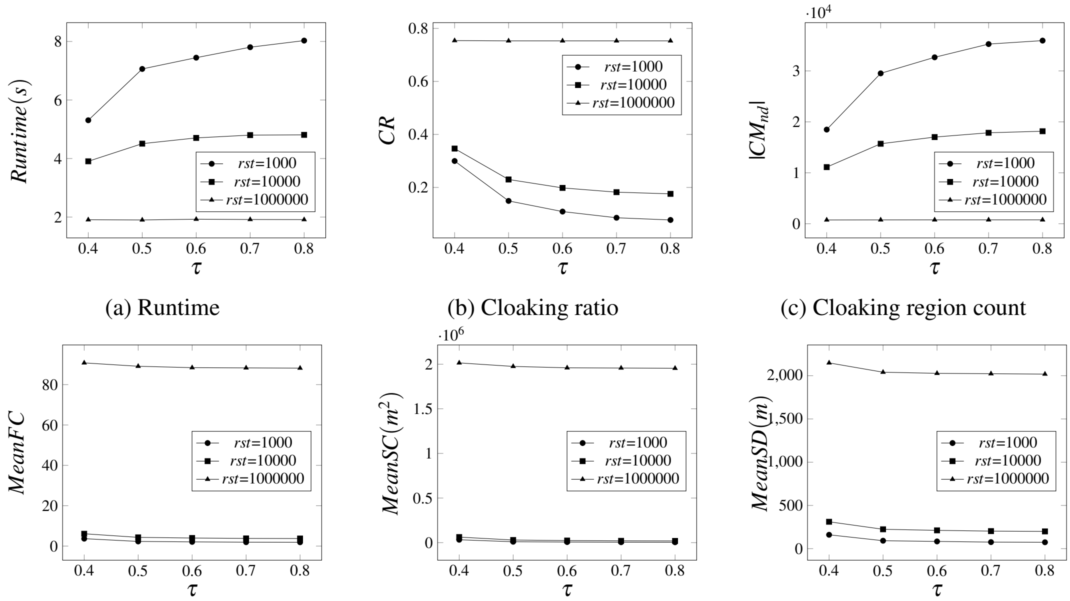
Figure 7: Performance metrics of Problem 3 on TKY



(a) Runtime (b) Cloaking ratio (c) Cloaking region count

(d) Mean feature count (e) Mean spatial coverage (f) Mean spatial diameter

Figure 8: Performance metrics of Problem 4 on NYC



(a) Runtime

(b) Cloaking ratio

(c) Cloaking region count

(d) Mean feature count

(e) Mean spatial coverage

(f) Mean spatial diameter

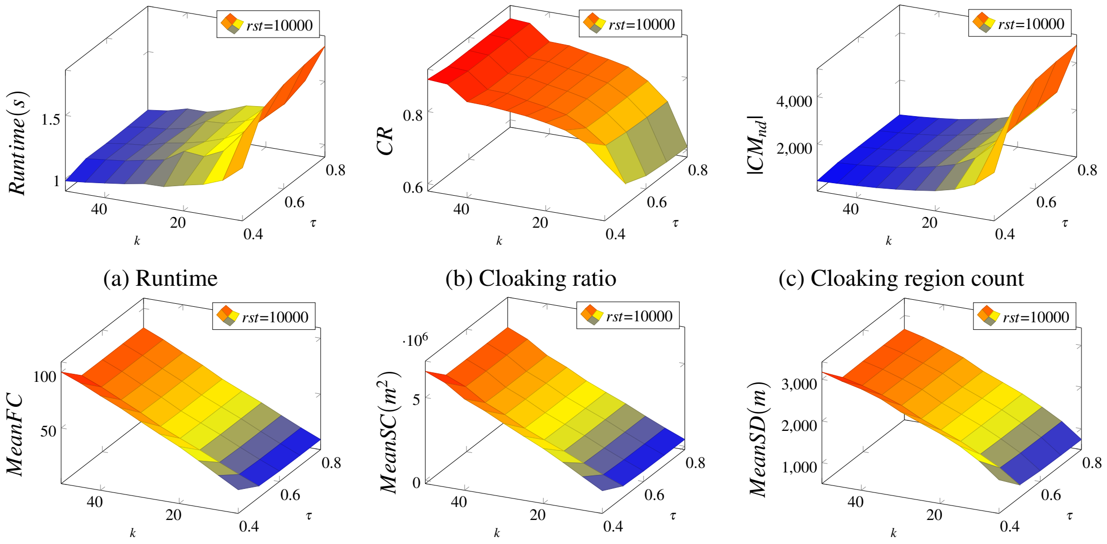
Figure 9: Performance metrics of Problem 4 on `TKY`

of 10000 m^2s. The runtime and the number of cloaking regions tend to increase with decreasing k and with increasing τ. On the other hand, with increasing k and decreasing τ we observe decreasing trends for other metrics, as shown in the sub-figures (b), (d), (e) and (f) of Figures 10 and 11. We would like to note that the effect of k is much more dominant in comparison to τ. We observe similar trends for Problem 6 on both of the datasets, as shown in Figures 12 and 13.

The experimental results, in essence, show that our framework produces effective cloaking maps even under the strict location privacy profiles (characterized by higher values of k, higher values of l and lower values of τ) for all of the six problem formulations on the two real world datasets. This can be simply verified by looking at the number of the cloaking regions in the plots. Indeed even with the strict cases (e.g., $k = 40$), the number of cloaking regions does not go to 1 (at which it degenerates to the whole city). In other words, our approach produces cloaking maps that is useful in practice. We consider that our framework is easily deployable in practice as it provides the user with six choices of location privacy preservation methods based on the additional available information and the individual privacy requirements. For instance, being the simplest of all, Problem 1 is very practical to use as it only asks the anonymity level k as the only parameter to be decided. The experimental results also confirm the utility/privacy tradeoff as with looser privacy requirements (e.g. $k = 10$) the cloaking maps contain large number of cloaking regions each with smaller spatial coverage.

6 Conclusion

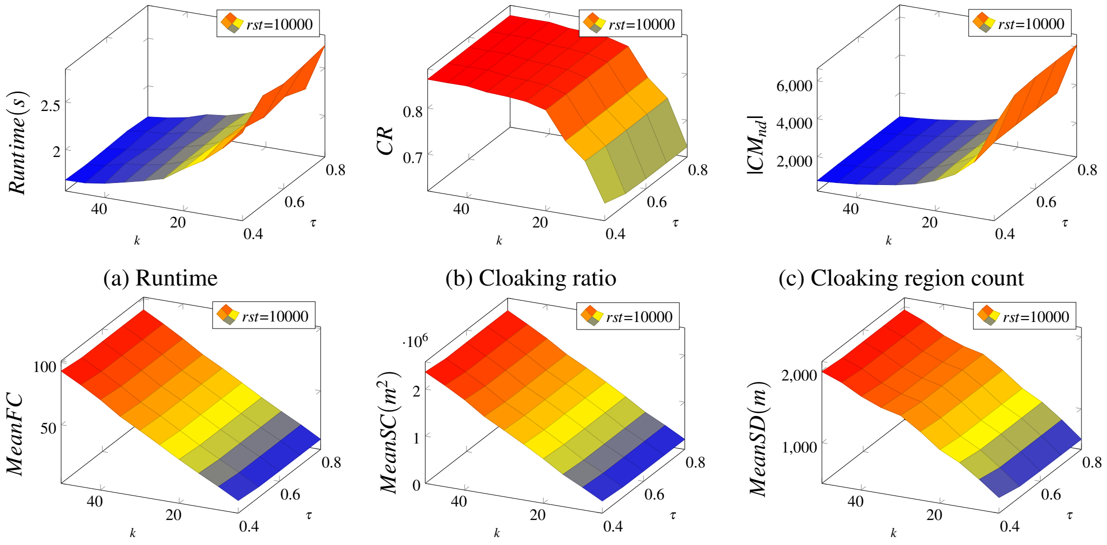
As the diversity and trustfulness of LBSs get widespread, each individual needs different location protection mechanisms and location privacy specification. For this reason, we proposed that users should be provided with a palette of location privacy specification alternatives addressing various



(a) Runtime (b) Cloaking ratio (c) Cloaking region count

(d) Mean feature count (e) Mean spatial coverage (f) Mean spatial diameter

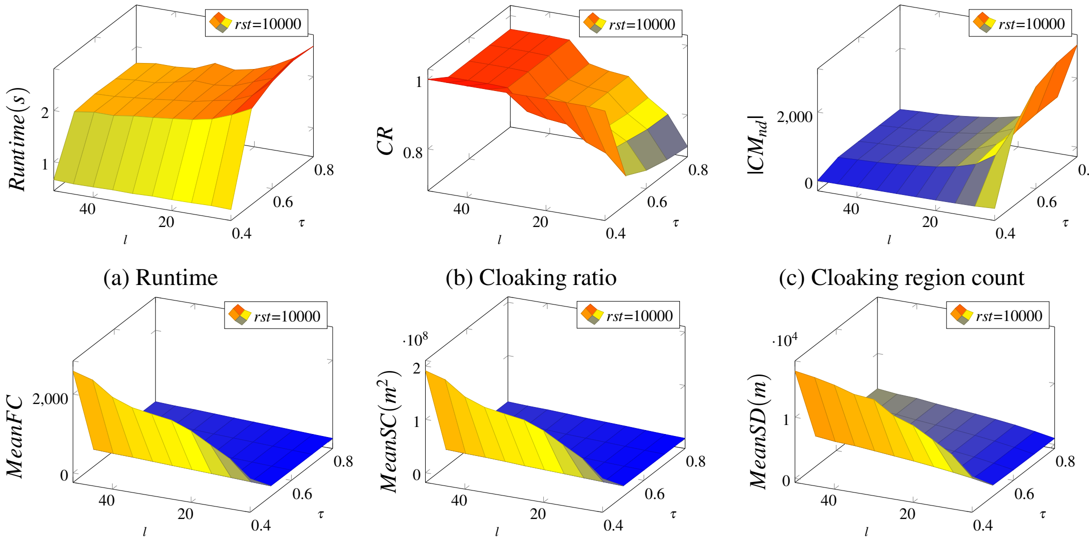
Figure 10: Performance metrics of Problem 5 on NYC



(a) Runtime (b) Cloaking ratio (c) Cloaking region count

(d) Mean feature count (e) Mean spatial coverage (f) Mean spatial diameter

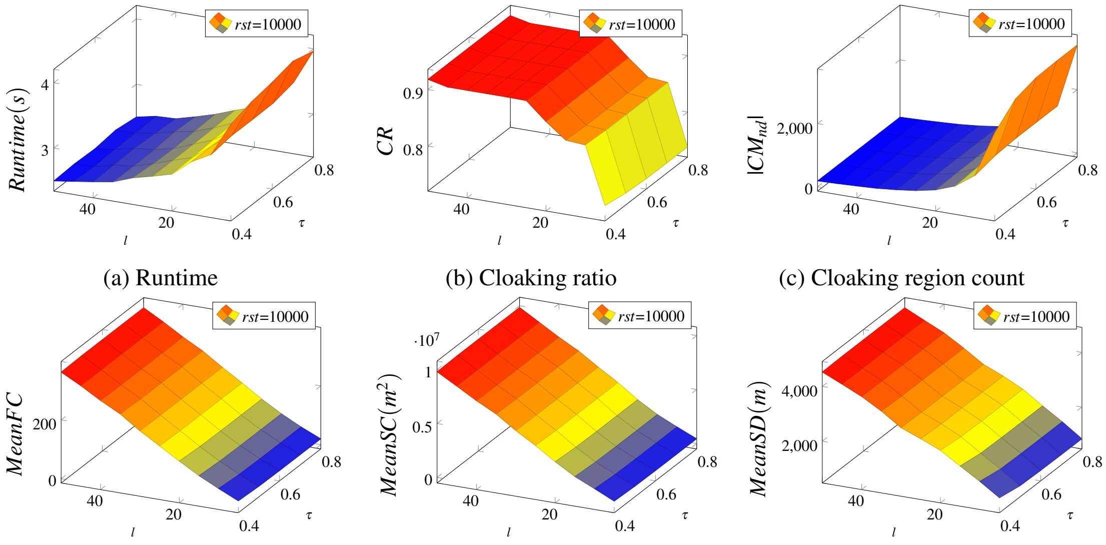
Figure 11: Performance metrics of Problem 5 on TKY



(a) Runtime (b) Cloaking ratio (c) Cloaking region count

(d) Mean feature count (e) Mean spatial coverage (f) Mean spatial diameter

Figure 12: Performance metrics of Problem 6 on `NYC`



(a) Runtime (b) Cloaking ratio (c) Cloaking region count

(d) Mean feature count (e) Mean spatial coverage (f) Mean spatial diameter

Figure 13: Performance metrics of Problem 6 on `TKY`

needs. As a result we introduced weak and strong notions of whereabouts, whatabouts and whereabouts+whatabouts protection for LBSs, resulting six different parametric location privacy specification alternatives. Fortunately, all of the six respective problem formulations can be solved within the same framework by exploiting the respective monotonicity properties. The algorithm, implementing the framework, progresses by a top-down method of space partitioning to obtain a cloaking map. The first stage of the algorithm, Intra-feature cloaking, is especially useful when spatial coverage rather than the representative points are available for the features.

We introduced several metrics to evaluate the utility of the resulting cloaking maps. The algorithm has been evaluated with the six problem formulations on two real world location check-in datasets. Using the datasets, we are able to obtain customized location privacy profiles with all of the six problems we proposed. This indeed shows the relevance and applicability of the problem formulations in the real world. The experimental results show that the algorithm is both effective and efficient with all of the problem formulations.

Acknowledgement

This work has been supported by TUBITAK under the grant number 114E132.

References

[1] O. Abul, F. Bonchi, and M. Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *Proc. of 24th International Conference on Data Engineering (ICDE)*, 2008.

[2] C. C. Aggarwal. On *k*-anonymity and the curse of dimensionality. In *Proc. of the 31th International Conference on Very Large Databases (VLDB 2005)*, pages 901–909, 2005.

[3] Mikhail J Atallah and Keith B Frikken. Privacy Preserving Location Dependent Query Processing. *Proc. of the The IEEE/ACS International Conference on Pervasive Services*, pages 9–17, 2004.

[4] M. Atzori, F. Bonchi, F. Giannotti, and D. Pedreschi. Anonymity preserving pattern discovery. *VLDB Journal*, 17(4):703–727, 2008.

[5] R. Cheng, Y. Zhang, E. Bertino, and S. Prabhakar. Preserving User Location Privacy In Mobile Data Management Infrastructures. *Proc. of the 6th international conference on Privacy Enhancing Technologies*, 4258:393–412, 2006.

[6] C. Chow, M. F. Mokbel, and W. G. Aref. Casper*: Query Processing for Location Services without Compromising Privacy. *ACM Transactions on Database Systems*, (34)4, 2009.

[7] M. L. Damiani, E. Bertino, and C. Silvestri. The PROBE Framework for the Personalized Cloaking of Private Locations. *Transactions on Data Privacy*, (3)2:123–148, 2010.

[8] C. Dwork. Differential privacy. In *Proc. of 33rd International Colloquium on Automata, Languages and Programming (ICALP'06)*, pages 1–12, 2006.

[9] B. Gedik and L. Liu. Location Privacy In Mobile Systems: A Personalized Anonymization Model. In *Proc. of 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, pages 620–629, 2005.

[10] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K-L. Tan. Private queries in location based services: anonymizers are not necessary. In *SIGMOD '08*, pages 121–132, New York, NY, USA, 2008. ACM.

[11] M. Gruteser and D. Grunwald. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *Proc. of the 1st International Conference on Mobile systems, Applications and Services*. ACM Press, 2003.

[12] H. Kido, Y. Yutaka, and T. Satoh. Protection of location privacy using dummies for location-based services. In *Proc. of 21st International Conference on Data Engineering Workshops (ICDEW '05)*, 2005.

[13] D. T. Lee and C. K. Wong. Worst-case analysis for region and partial region searches in multidimensional binary search trees and balanced quad trees. *Acta Informatica*, 9(1):23–29, 1977.

[14] N. Li, T. Li, and S. Venkatasubramanian. T-Closeness: Privacy Beyond K-Anonymity And L-Diversity. In *Proc. of 23rd International Conference on Data Engineering (ICDE)*, pages 106–115, 2007.

[15] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. *l*-diversity: privacy beyond *k*-anonymity. In *Proc. of the 22nd Int. Conf. on Data Engineering (ICDE'06)*, 2006.

[16] Anna Monreale, Roberto Pellungrini, et al. A survey on privacy in human mobility. *Transactions on Data Privacy*, 16(1):51–82, 2023.

[17] Nazmun Naher and Tanzima Hashem. Think ahead: Enabling continuous sharing of location data in real-time with privacy guarantee. *The Computer Journal*, 2018.

[18] M. E. Nergiz, M. Atzori, Y. Saygin, and B. Guc. Towards Trajectory Anonymization A Generalization Based Approach. *Transactions on Data Privacy*, 2(106):47–75, 2009.

[19] Dilay Parmar and Udai Pratap Rao. Privacy-preserving enhanced dummy-generation technique for location-based services. *Concurrency and Computation: Practice and Experience*, 35(2):e7501, 2023.

[20] D. N. Politis. *On the Entropy of a Mixture Distribution*. Technical Report #91-67. Purdue University, 1991.

[21] Zohaib Riaz, Frank Dürr, and Kurt Rothermel. Understanding vulnerabilities of location privacy mechanisms against mobility prediction attacks. In *Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services.*, pages 7–11, 2017.

[22] P. Samarati. Protecting respondents identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

[23] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.P. Hubaux, and Y. Le Boudec. Protecting location privacy: optimal strategy against localization attacks. In *Proc. of 19th ACM Conference on Computer and Communications Security (CCS'12)*, 2012.

[24] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[25] Dingqi Yang, Daqing Zhang, Vincent. W. Zheng, and Zhiyong Yu. Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(1):129–142, 2015.

[26] E. Yigitoglu, M.L. Damiani, O. Abul, and C. Silvestri. Privacy-preserving sharing of sensitive semantic locations under road-network constraints. In *Proc. of IEEE Mobile Data Management (MDM 2012)*, 2012.

[27] Man Lung Yiu, Christian S Jensen, Xuegang Huang, and Hua Lu. SpaceTwist Managing The Trade Offs Among Location Privacy Query Performance And Query Accuracy in Mobile Services. *2008 IEEE 24th International Conference on Data Engineering*, 00:366–375, 2008.

[28] Zhirun Zheng, Zhetao Li, Hongbo Jiang, Leo Yu Zhang, and Dengbiao Tu. Semantic-aware privacy-preserving online location trajectory data sharing. *IEEE Transactions on Information Forensics and Security*, 17:2256–2271, 2022.