Measuring Query Privacy in Location-Based Services

Xihui Chen*
Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg
xihui.chen@uni.lu

Jun Pang
Computer Science and Communications, University of Luxembourg
jun.pang@uni.lu

ABSTRACT

The popularity of location-based services leads to serious concerns on user privacy. A common mechanism to protect users' location and query privacy is spatial generalisation. As more user information becomes available with the fast growth of Internet applications, e.g., social networks, attackers have the ability to construct users' personal profiles. This gives rise to new challenges and reconsideration of the existing privacy metrics, such as k-anonymity. In this paper, we propose new metrics to measure users' query privacy taking into account user profiles. Furthermore, we design spatial generalisation algorithms to compute regions satisfying users' privacy requirements expressed in these metrics. By experimental results, our metrics and algorithms are shown to be effective and efficient for practical usage.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; K.4.1 [**Computers and Society**]: Public Policy Issues—*Privacy*

General Terms

Security, measurement

Keywords

Location based services, query privacy, anonymity, measurement

1. INTRODUCTION

The popularity of mobile devices with localisation chips and ubiquitous access to Internet give rise to a large number of location-based services (LBS). Consider a user who wants to know where the nearest gas station is. He sends a query to a location-based service provider (LBSP) using his smart-phone with his location attached. The LBSP then processes the query and responds with results. Location-based queries lead to privacy concerns especially in cases when LBSPs are not trusted. Attackers can cooperate with LBSPs and have access to users' location-related queries. The amount and risk of information leakage from LBS queries have been discussed, for example, in [7, 13]. The analysis mainly focused on information leakage from locations. However, query content itself is also a source of users' privacy leakage. For instance, a query about casinos implies the issuer's gambling habit which the issuer wants to keep secret. Thus besides location privacy, the anonymity of issuers with respect to queries is also important in privacy preservation. Intuitively, *query privacy* is the ability to prevent other parties to learn the issuers of queries. One way to protect query privacy is to anonymise queries by removing users' identities. However, this does not suffice when considering locations which can help reveal users' identities, since attackers can acquire users' locations through a number of ways, e.g., triangulating mobile phones' signals and localising users' access points to Internet. Sometimes, public information such as home addresses and yellow pages can also help obtain users' positions. Therefore, locations within queries are critical in protecting users' query privacy as well. Replacing locations with a generalised area is an alternative to break the linkability between users and their locations, which is called *spatial cloaking* [16, 20].

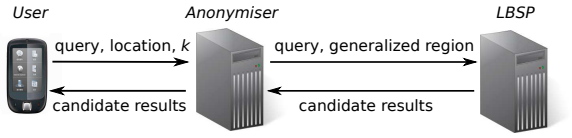

Figure 1: A centralised framework of LBSs

In the last few years, k-anonymity [24] has been widely used and investigated in the literature on releasing microdata, e.g., medical records. A user is k-anonymous if he is indistinguishable from at least other $k-1$ users. In the context of query privacy in LBS, k-anonymity can be interpreted as: given a query, any attacker based on the query location cannot identify the issuer with probability larger than $\frac{1}{k}$ [15]. Most of existing works adopt the centralised framework (depicted in Fig. 1), where a trusted agent *anonymiser* is introduced. Users first send their queries to the anonymiser who *anonymises the queries* and *generalises the locations* before sending them to the LBSP. The responses from the LBSP are first sent to the anonymiser and then forwarded to the corresponding users. In the centralised framework, normally it is assumed that the communication channels between users and the anonymiser are secure while the ones between the anonymiser and the LBSP are public.

A common assumption for k-anonymity is that all users have the same probability to issue queries. In other words, a uniform probability distribution is assumed over users with respect to sending any

*This work was supported by the FNR Luxembourg under project SECLOC 794361.

CODASPY'12, February 7–9, 2012, San Antonio, Texas, USA.

query, which is often not realistic especially when attackers gain more information about the users. Given a specific query, certain users tend to be more likely to issue it when compared to others. For instance, users who love movies are more possible to search for nearing cinemas. For any user in a generalised area satisfying k-anonymity, the probability to be the issuer is no longer $\frac{1}{k}$ in such situations. The case can be worse especially for those users who are more likely than others. Suppose a k-anonymised region of a query from a young person for searching clubs at midnight. If there are only two young people in the generalised region, then they are more likely to be taken as the candidates for the issuer from attackers' view than other users in this region. Therefore, k-anonymity is not a sufficient metric to describe users' privacy requirements when taking into account user profiles, which was addressed first by Shin et al. [26]. Nowadays, the popularity of social networks and more exposure of people's information on Internet provide attackers sources to gather enough background knowledge to obtain user profiles. Besides *passive attacks* in which attackers simply observe the connection between users, attackers can also perform *active attacks*, e.g., by creating new accounts so as to identify users even in an anonymised social network [17]. Wu et al. give a literature study on the existing attacks to obtain users' profiles [33]. Therefore, it is a new challenge to measure and protect users' query privacy in LBSs with the assumption that attackers have the knowledge of user profiles.

Our contributions. In this paper, we extend k-anonymity and propose new metrics to correctly measure users' query privacy in the context of LBSs, which enable users to specify their query privacy requirements in different ways. Furthermore, we design new generation algorithms to compute anonymising spatial regions according to users' privacy requirements. Through experiments, we show that our algorithms are efficient enough to meet users' demands on real-time responses and generate regions satisfying privacy requirements. We also show the different strengths of our metrics which help users choose the correct requirements to achieve a balance between privacy and the quality of service delivered by the LBSP.

Structure of the paper. Sect. 2 gives a brief investigation of related work on measuring anonymity, query privacy and area generalisation algorithms. In Sect. 3, we present our formal framework, the threat model, and the derivation of user profiles. We formally define a number of query privacy metrics in Sect. 4 and develop generalisation algorithms in Sect. 5. In Sect. 6, through experiments we discuss features of the metrics and evaluate the performance of the generalisation algorithms. The paper is concluded in Sect. 7.

2. RELATED WORK

We give a brief literature study on measuring anonymity and on query privacy metrics with focus on k-anonymity. Then we summarise existing region generalisation algorithms.

2.1 Anonymity metrics

In the literature, various ways to measure anonymity have been proposed. Chaum [6] uses the size of an anonymity set to indicate the degree of anonymity provided by a network based on Dining Cryptographers. An anonymity set is defined as the set of users who could have sent a particular message as observed by attackers. Berthold et al. [3] define the degree of anonymity as $\log N$, where N is the number of users. Reiter and Rubin [22] define the degree of anonymity as the probability that an attacker can assign to a user of being the original sender of a message. They introduce metrics like beyond suspicion, probable innocence and possible innocence. Serjantov and Danezis [25] define an anonymity metric based on entropy and a similar metric is given by Díaz et al. [11] which is normalised by the number of users. Zhu and Bettati [35] propose a definition of anonymity based on mutual information. The notion relative entropy is used by Deng et al. [10] to measure anonymity. Different information-theoretic approaches based on Kullback-Leider distance and min-entropy are proposed [5, 9, 31] to define information leakage or the capacity of noisy channels.

2.2 Query privacy metrics

The concept of k-anonymity was originally proposed by Samarati and Sweeney in the field of database privacy [24]. The main idea of k-anonymity is to guarantee that a database entry's identifier is indistinguishable from other $k-1$ entries. However, this method does not work in all cases. For instance, the fact that an HIV carrier is hidden in k carriers does not help protecting his infection of the virus. Further research has been done to fix this problem [18]. In the context of privacy in LBSs, k-anonymity is first introduced by Gruteser and Grunwald [15]. It aims to protect two types of privacy – *location privacy* and *query privacy*. The former means that given a published query, attackers cannot learn the issuer' exact position while the latter enforces the unlinkability between the issuer and the query. Because of its simplicity, k-anonymity has been studied and refined in many ways. For instance, Tan et al. define *information leakage* to measure the amount of revealed location information in spatial cloaking, which quantifies the balance between privacy and performance [32]. Xue et al. [34] introduce the concept of *location diversity* to ensure generalised regions to contain at least ℓ semantic locations (e.g., schools, hospitals).

Deeper understanding of k-anonymity reveals its drawbacks in preserving users' location privacy. Shokri et al. analyse the effectiveness of k-anonymity in protecting location privacy in different scenarios in terms of adversaries' background information [30], i.e., *real-time location information, statistical information* and *no information*. Based on the analysis, they conclude that cloaking (e.g., k-anonymity) is effective for protecting query privacy but not location privacy. They also show its flaws which the adversary can exploit to infer users' current locations. In this paper, we focus on protecting query privacy using cloaking with the assumption that the adversary learns users' real-time locations.

Recently, Shokri et al. design a tool *Location-Privacy Meter* that measures location privacy of mobile users in different attack scenarios [28, 29]. Their work assumes that attackers can utilise user profiles (e.g., mobility patterns) extracted from uses' sample traces to infer the ownership of collected traces. It is in spirit close to our work. They use the incorrectness of attackers' conclusions on users' positions drawn from observations as the privacy metric. In this paper, we focus on users' query privacy with regards to an individual query rather than query histories. Moreover, we make use of users' static and public personal information, such as professions and jobs as user profiles. Considering information such as mobility patterns and query histories is part of our future work.

The work by Shin et al. [26] is most closely related. They describe user profiles using a set of attributes whose domains are discretised into disjoint values. User profiles are represented by *profile vectors* with a bit for each value. Shin et al. propose three new metrics based on k-anonymity by restricting different levels of similarity between profiles of users in generalised regions. This is analogous to our notion of *k-approximate beyond suspicion* which will be discussed in Sect. 4. Compared to Shin et al.'s work [26], we define a more comprehensive set of metrics that can measure query privacy from different perspectives and develop corresponding generalisation algorithms.

2.3 Area generalisation algorithms

The first generalisation algorithm called IntervalCloaking is designed by Gruteser and Grunwald [15]. Their idea is to partition a region into quadrants with equal area. If the quadrant where the issuer is located contains less than k users, then the original region is returned. Otherwise, the quadrant with the issuer is taken as input for the next iteration. The algorithm CliqueCloak [14] is proposed by Gedik and Liu in which regions are generalised based on the users who have issued queries rather than all potential issuers. The major improvement is that this algorithm enables users to specify their personal privacy requirements by choosing different values for k. Mokbel et al. [21, 8] design the algorithm Casper which employs a quadtree to store the two-dimensional space. The root node represents the whole area and each of other nodes represents a quadrant region of its parent node. The generalisation algorithm starts from the leaf node which contains the issuer and iteratively traverses backwards to the root until a region with more than k users is found. Another algorithm nnASR [16] simply finds the nearest k users to the issuer and returns the region containing these users as the anonymising spatial region.

The above algorithms suffer from a particular attack called "outlier problem" [2], where attackers have the generalisation algorithms and users' spatial distribution as part of their knowledge. Intuitively, this happens when some users in a generalised region do not have the same region returned by the algorithm as the issuer. Thus, these users can be removed from the anonymity set, resulting in a set with less than k users. Hence, an algorithm against this attack needs to ensure that for each user in the anonymity set it always returns the same region. Kalnis et al. design the first algorithm called hilbASR that does not have the outlier problem [16]. The algorithm exploits the Hilbert space filling curve to store users in a total order based on their locations. The curve is then partitioned into blocks with k users. The block with the issuer is returned as the generalised region. Mascetti et al. propose two algorithms, dichotomicPoints and grid, which are also secure against the outlier problem [20]. The former iteratively partitions the region into two blocks until less than $2k$ users are located in the region while the latter draws a grid over the two-dimensional space so that each cell contains k users and returns the cell with the issuer. Because of the simplicity of implementation and the relatively smaller area of the generalised regions, we adopt and extend these two algorithms in our algorithm design.

The area of generalised regions is usually used to measure the quality of service responded by LBSPs, as smaller regions lead to more accurate query results and less communication overhead.

3. PRELIMINARIES

In this section, we present a formal framework, define the attacker model subsequently, and discuss how to derive a priori probabilities for users to issue a query based on their profiles.

3.1 A formal framework

Let $\mathcal{U}$ denote a set of users, $\mathcal{L}$ the set of locations (positions), and $\mathcal{T}$ the set of time instances that can be recorded. The granularity of time instances is determined by LBSs. Given a time t, we have a function to map a user to his location at t: $whereis : \mathcal{U} \times \mathcal{T} \to \mathcal{L}$. The user *spatial distribution* at time t can be defined as the set $\{(u, whereis(u,t)) \mid u \in \mathcal{U}\}$, denoted by $dis(t)$. Suppose the set of queries supported by LBSs is represented by $\mathcal{Q}$, e.g., the nearest gas station. Let $Q \subseteq \mathcal{U} \times \mathcal{L} \times \mathcal{T} \times \mathcal{Q}$ be the set of queries from users $\mathcal{U}$ at a specific time. An element in Q is a quadruple $\langle u, whereis(u,t), t, q\rangle$, where $u \in \mathcal{U}$ and $q \in \mathcal{Q}$.

Table 1: Notations

$\mathcal{U}$	set of users
$\mathcal{T}$	set of time instances
$\mathcal{L}$	set of locations
$\mathcal{R}$	set of possible generalised regions
$q \in \mathcal{Q}$	a query supported by the LBS
$\langle u, \ell, t, q\rangle \in Q$	a query issued by u at position ℓ at time t
$\langle r, t, q\rangle \in Q'$	a generalised query sent by the anonymiser
$dis(t)$	spatial distribution of users in $\mathcal{U}$ at time t
$\mathcal{M}(q)$	probability distribution of user to issue q
$u\ell(r,t)$	set of users located in region r at time t
$req(\langle u, \ell, t, q\rangle)$	user u's privacy requirement on $\langle u, \ell, t, q\rangle$
$p(u \mid q)$	probability of u to issue q among users in $\mathcal{U}$
$p(u \mid \langle r, t, q\rangle)$	probability of u to issue $\langle r, t, q\rangle$
$whereis(u,t)$	position of user u at time t
$f(\langle u, \ell, t, q\rangle)$	an algorithm computing generalised queries

Given a query $\langle u, whereis(u,t), t, q\rangle \in Q$, the anonymising server (*anonymiser*) would remove the user's identity and replace his location with a larger area to protect his query privacy. We only consider *spatial generalisation* in this paper as in LBSs users require instant responses. Let $2^{\mathcal{L}}$ be the power set of $\mathcal{L}$ and then we use $\mathcal{R} \subset 2^{\mathcal{L}}$ to denote the set of all possible generalised regions. The corresponding output of the anonymising server can be represented as $\langle r, t, q\rangle$, where $r \in \mathcal{R}$ and $whereis(u,t) \in r$. Suppose the set of generalised queries $Q' \subset \mathcal{R} \times \mathcal{T} \times \mathcal{Q}$. The generalisation algorithm of the aonymiser can be represented as a function $f : Q \to Q'$. For instance, we have $f(\langle u, whereis(u,t), t, q\rangle) = \langle r, t, q\rangle$.

The generalisation algorithm used by the anonymiser to compute generalised queries makes use of current user spatial distribution and might also take users' privacy requirements as part of its input. In our framework, a privacy requirement is represented by a pair – a chosen privacy metric by the issuer and the corresponding specified value (see more discussion in Sect. 4 and 5). We use $req(\langle u, whereis(u,t), t, q\rangle)$ to denote u's requirement on query $\langle u, whereis(u,t), t, q\rangle$.

We use $p(u_j \mid q_i)$ to denote the conditional probability of user u_j to be the issuer when query q_i is observed, and $\sum_{u_j \in \mathcal{U}} p(u_j \mid q_i) = 1$. Variations of users' profiles along with time and positions are out of the scope of this paper, and considered as part of our future work. For the sake of simplicity, in the following discussion we use a probability matrix $\mathcal{M}$, where element $m_{ij} = p(u_j \mid q_i)$. We use $\mathcal{M}(q_i)$ to denote the i-th row of $\mathcal{M}$, the probability distribution over users to issue the query q_i.

Let $u\ell : \mathcal{R} \times \mathcal{T} \to 2^{\mathcal{U}}$ be the function mapping a region to the set of users located in it. In other words, $u\ell(r,t) = \{u \in \mathcal{U} \mid whereis(u,t) \in r\}$. Given a generalised query $\langle r, t, q\rangle$, user u's probability to be the issuer among the users in region r can be computed as follows:

$$p(u \mid \langle r, t, q\rangle) = \frac{p(u \mid q)}{\sum_{u' \in u\ell(r,t)} p(u' \mid q)}$$

We summarise the list of important notations in Tab. 1.

3.2 The attacker model

Through generalising locations, users' query privacy is protected by preventing attackers from re-identifying issuers with high probabilities. Most approaches in the literature (e.g., see [20]) assume that attackers have a global view of users' real-time spatial distribution (*Assumption 1*). This assumption is conservative but possi-

ble in real scenarios. There are many ways to gather users' real-time locations. For instance, most people send queries from some fixed positions, e.g., office and home. Referring to address books or other public database, the potential issuers can be identified. We also adopt this assumption in this paper. It is also natural to assume that the attacker controls the communication channel between the anonymiser and the LBS server (see the second part of Fig. 1) (*Assumption 2*). This allows the attacker to acquire any generalised queries forwarded by the anonymiser. Meantime, we assume the anonymiser is trustworthy and users have a secure connection with the anonymiser through SSL or other techniques (see the first part of Fig. 1). The generalisation algorithm used by the anonymiser is assumed to be public (*Assumption 3*). This leads to an additional requirement. For each user in an anonymity set, a plausible algorithm must compute the same area as the one computed for the issuer.

Different from attackers in the literature (e.g., [20, 32, 30]), the attacker in our model has access to an a priori distribution over users with regards to issuing queries (i.e., the probability matrix $\mathcal{M}$) (*Assumption 4*). Thus, instead of assuming a uniform distribution among users for issuing a particular query, the attacker has a precise probabilistic distribution by exploring user profiles obtained, e.g., by available public information [17, 26].

Users may have different privacy requirements for queries dependent on time, positions and sensitivity of queries, which is usually a subjective decision. So we assume that attackers have no knowledge about this requirement decision process (*Assumption 5*). However, attackers can learn users' privacy requirements after observing the generalised queries by the anonymiser (*Assumption 6*). This is realistic as from the features of the generalised queries, attackers can infer the corresponding privacy requirements.

Last but not least, we assume that the attacker cannot link any two queries from the same user (*Assumption 7*). All queries are independent from the attacker's perspectives. This assumption is strong but still realistic as users tend to issue occasional queries and an issuer's identity is always removed by the anonymiser before forwarding the query to the LBSP.

3.3 Deriving probabilities from user profiles

User profiles can be associated with a set of attributes which can be divided into several categories, e.g., contact attributes (zip codes, addresses), descriptive attributes (age, nationalities, jobs) and preference attributes (hobbies, moving patterns) [26]. The values of these attributes can be discretised into a categorical form. For instance, the value of a home address can be represented by the corresponding zone which it lies in. In this way, each attribute has a finite number of candidate values.

Let $\phi_u = \langle a_1, \ldots, a_m \rangle$ be the profile of user u, where m is the number of attributes. Note that a_i is represented by a string of bits, each of which denotes a possible value of the corresponding attribute. We use $|\, a_i \,|$ to denote the length of a_i and $\hat{\phi}_u$ to represent the concatenation of the strings of all attributes. Moreover, let $\hat{\phi}_u[j]$ be the j-th bit of $\hat{\phi}_u$. As the values in the domain of any attribute are disjoint, there is at most one bit to be 1 for any a_i (perhaps all zeros because of lack of information). Consider a user profile consisting of two attributes – salary and gender. As the domain of gender consists of two values – *male* and *female* we use two bits to represent them, 01 and 10, respectively. We divide the numerical values of salary into three intervals – '≤ 1000', '$1000-5000$' and '≥ 5000'. Then user profile $\phi_u = \langle 001, 01 \rangle$ means user u is male and has a salary more than 5000, and $\hat{\phi}_u = 00101$.

Each query $q \in Q$ must have a subset of correlated attributes that can be used to deduce the issuer. Furthermore, each value of a relevant attribute has a different weight measuring the probability that the user issues the given query when having the attribute value. For instance, for the query asking for expensive hotels, the associated attributes should include salary, jobs and age while gender is irrelevant. Among them, a salary is much more relevant than age and moreover, a salary of more than 5000 euros is much more important than one of less than 1000 euros. Therefore, we introduce a relevance vector to express the relation between attributes' values and queries. Let $W(q) = \langle w_1, \ldots, w_n \rangle$ be the relevance vector of query q where $n = \sum_{i \leq m} |\, a_i \,|$.

For any $u \in \mathcal{U}$ and $q \in Q$, let $\mathcal{V}(u, q) = \sum_{i \leq n} w_i \cdot \hat{\phi}_u[i]$ be the relevance of user u's profile to query q. Subsequently, we have:

$$p(u \,|\, q) = \frac{\mathcal{V}(u, q)}{\sum_{u' \in \mathcal{U}} \mathcal{V}(u', q)}$$

4. QUERY PRIVACY METRICS

We propose a number of new metrics (except for k-anonymity) to measure query privacy taking into account user profiles and formally define them using the framework discussed in Sect. 3.

k-anonymity. In k-anonymity, a natural number k is taken as the metric of users' query privacy, which is the size of the anonymity set of the issuer. This means, for a given query, there are at least k users in the generalised region including the issuer. Moreover, in order to prevent attacks based on public generalisation algorithms [20], any user in the anonymity set must have the same generalised region for the same query. Similar to the definitions in the literature [27, 20], k-anonymity can be formally defined as follows:

DEFINITION 1. *Let $\langle u, whereis(u,t), t, q \rangle \in Q$ be a query and $\langle r, t, q \rangle \in Q'$ the corresponding generalised query. The issuer u is* k-anonymous *if*

$$|\, \{u' \in \mathcal{U} \mid \begin{array}{l} whereis(u',t) \in r \wedge \\ f(\langle u', whereis(u',t), t, q \rangle) = \langle r, t, q \rangle\} \mid \end{array} \geq k$$

Note that in Def. 1, as all users in the anonymity set take r as the generalised region for the query q at time t, they are all k-anonymous. The following proposed new anonymity metrics enjoy the same property.

k-approximate beyond suspicion. As discussed in Sect. 1, when user profiles are considered as part of the attacker's knowledge, the size of an anonymity set k cannot be a fair metric for query privacy. Especially for users with high a priori probabilities, they can easily be chosen as candidates of issuers. Inspired by anonymity degrees defined by Reiter and Rubin [22], we come up with the following new privacy metrics.

Beyond suspicion means from the attacker's viewpoint the issuer cannot be more likely than other potential users in the anonymity set to send the query. In other words, users in the anonymity set have the same probability to perform an action. In the context of LBSs, we need to find a set of users in which users are the same likely to send a given query. This set is taken as the anonymity set whose size determines the degree of users' privacy as in k-anonymity. Let $AS : Q' \rightarrow 2^{\mathcal{U}}$ denote the anonymity set of a generalised query. An issuer of query $\langle u, whereis(u,t), t, q \rangle$ is beyond suspicious with respect to the corresponding generalised query $\langle r, t, q \rangle$ if and only if $\forall u' \in AS(\langle r, t, q \rangle)$,

$$p(u \,|\, \langle r, t, q \rangle) = p(u' \,|\, \langle r, t, q \rangle)$$

In practice, the number of users with the same probability to send a query is usually small, which leads to a large generalised area with a fixed k. So we relax the requirement to compute an anonymity set consisting of users with *similar probabilities* instead of the exact same probability. Let $\|p_1, p_2\|$ denote the difference between

two probabilities and ϵ be the pre-defined parameter describing the largest difference allowed between similar probabilities.

DEFINITION 2. *Let $\langle u, whereis(u,t), t, q\rangle \in Q$ be a query and $\langle r, t, q\rangle \in Q'$ the corresponding generalised query. The issuer u is* k-approximate beyond suspicious *if*

$$|\{u' \in AS(\langle r,t,q\rangle) \mid \|p(u \mid \langle r,t,q\rangle), p(u' \mid \langle r,t,q\rangle)\| < \epsilon \wedge f(\langle u', whereis(u',t), t, q\rangle) = \langle r,t,q\rangle\}| \geq k.$$

Different from k-anonymity, the set of users that are k-approximate beyond suspicious is computed based on the spatial distribution of users with similar probabilities rather than the original distribution involving all users. The users in an anonymity set have similar probabilities and the size of the anonymity set is larger than k. Therefore, k-approximate beyond suspicion can be seen as a generalised version of k-anonymity. If for a specific query $q \in \mathcal{Q}$, any two users have the same probability to issue it (i.e., $\mathcal{M}(q)$ is a uniform distribution), then k-approximate beyond suspicion is equivalent to k-anonymity.

THEOREM 1. *For a given query $q \in \mathcal{Q}$, if for any two users $u_1, u_2 \in \mathcal{U}$ we have $p(u_1 \mid q) = p(u_2 \mid q)$, then k-anonymity is k-approximate beyond suspicion with respect to q.*

For short, we use k-ABS to denote k-approximate beyond suspicion in the following discussion.

User specified innocence. Two weaker anonymity metrics, *probable innocence* and *possible innocence*, are proposed by Reiter and Rubin as well [22]. An issuer is probably innocent if from the attacker's view the issuer appears no more likely to be the originator of the query. In other words, the probability of each user in the anonymity set to be issuer should be less than 50%. Meantime, possible innocence requires the attacker not be able to identify the issuer with a non-trivial probability. We extend these two notions into a metric with user-specified probabilities (instead of restricting to 50% or non-trivial probability which is not clearly defined). We call the new anonymity metric *user specified innocence* where $\alpha \in [0,1]$ is the specified probability given by the issuer.

DEFINITION 3. *Let $\alpha \in [0,1]$, $\langle u, whereis(u,t), t, q\rangle \in Q$ be a query and $\langle r, t, q\rangle \in Q'$ the corresponding generalised query. The issuer u is* α-user specified innocent *if for all $u' \in u\ell(r,t)$,*

$$p(u' \mid \langle r,t,q\rangle) \leq \alpha \wedge f(\langle u', whereis(u',t), t, q\rangle) = \langle r,t,q\rangle.$$

Recall that $u\ell(r,t)$ denotes the set of users in region r at time t. It is clear that the anonymity set consists of all users in the generalised area. We abbreviate α-user specified innocence as α-USI.

Intuitively, for a query, an issuer is α-user specified innocent, if the anonymiser generates the same region for any user in the region with the same specified value α. In other words, in the generalised region, the most probable user has a probability smaller than α from the attacker's point of view. With this property, α-USI can also be captured by *min-entropy*, which is an instance of Rènyi entropy [23] and is used to measure the uncertainty of the *one-try* adversary who has exactly one chance to guess the originator in our scenario. Obviously, the best strategy for the adversary is to choose the one with the highest probability. Formally, the *min-entropy* of a variable X is defined as $H_\infty(X) = -\log \max_{x \in \mathcal{X}} p(x)$ where $\mathcal{X}$ is the domain of X. Let U be the variable that stands for the issuer and its domain is $\mathcal{U}$. Then for query $\langle r,t,q\rangle$, the min-entropy of the attacker can be described as $H_\infty(U \mid \langle r,t,q\rangle) = -\log \max_{u \in u\ell(r,t)} p(u \mid \langle r,t,q\rangle)$. It is maximised when the users in region r at time t follow a uniform distribution with regards to issuing query q. It is easy to verify that if a generalised query $\langle r,t,q\rangle$ guarantees the issuer α-user specified innocent, then it also ensures that the corresponding min-entropy is bigger than $-\log\alpha$.

An entropy based metric. Serjantov and Danezis [25] define an anonymity metric based on entropy and Díaz et al. [11] provide a similar metric that is normalised by the number of users in the anonymity set. The concept *entropy* of a random variable X is defined as $H(X) = -\sum_{x \in \mathcal{X}} p(x) \cdot \log p(x)$ where $\mathcal{X}$ is the domain (all possible values) of X. In our context, entropy can also be used to describe the attacker's uncertainty to identify the issuer of a generalised query. Let variable U denote the issuer of query $\langle r,t,q\rangle$. Then the uncertainty of the attacker can be expressed as

$$H(U \mid \langle r,t,q\rangle) = -\sum_{u' \in u\ell(r,t)} p(u' \mid \langle r,t,q\rangle) \cdot \log p(u' \mid \langle r,t,q\rangle).$$

Users can express their query privacy by specifying an entropy value. For a given generalised query $\langle r,t,q\rangle$ and a given value β, we say the issuer is entropy based anonymous with respect to the value β if all users in region r can have r as the generalised region when issuing the same query and the entropy $H(U \mid \langle r,t,q\rangle)$ is not smaller than β.

DEFINITION 4. *Let $\beta > 0$, $\langle u, whereis(u,t), t, q\rangle \in Q$ be a query and $\langle r, t, q\rangle \in Q'$ the corresponding generalised query. The issuer u is* β-entropy based anonymous *if for all $u' \in u\ell(r,t)$,*

$$H(U \mid \langle r,t,q\rangle) \geq \beta \wedge f(\langle u', whereis(u',t), t, q\rangle) = \langle r,t,q\rangle.$$

For short, we call β-entropy based anonymity β-EBA.

A mutual information based metric. The notion *mutual information* in information theory quantifies the mutual dependence of two random variables. It is usually denoted as $I(X;Y)$ and computed as the difference $H(X) - H(X \mid Y)$ where $H(X \mid Y)$ is the conditional entropy of X when knowing Y. In the context of query privacy, we can use mutual information to evaluate the uncertainty reduced after revealing the generalised query. Before the generalised query is known to the attcker, he only knows that the query q can be issued by a user U in $\mathcal{U}$ with the probability $p(U \mid q)$. So the uncertainty of the attacker can be described as entropy $H(U \mid q)$. After the attacker learns the generalised query, the uncertainty on the issuer can be described as the conditional entropy $H(U \mid \langle r,t,q\rangle)$. Therefore, for a given query q the amount of information gained by the attacker after observing the corresponding generalised query can be computed as

$$\begin{aligned} I(U \mid q; \langle r,t,q\rangle) = {} & H(U \mid q) - H(U \mid \langle r,t,q\rangle) \\ = {} & -\textstyle\sum_{u' \in \mathcal{U}} p(u' \mid q) \cdot \log p(u' \mid q) \\ & +\textstyle\sum_{u' \in u\ell(r,t)} p(u' \mid \langle r,t,q\rangle) \cdot \\ & \qquad \log p(u' \mid \langle r,t,q\rangle). \end{aligned}$$

Similar to β-EBA, the issuer of query $\langle r,t,q\rangle$ is γ-mutual information based anonymous if $I(U \mid q; \langle r,t,q\rangle)$ is less than γ and all users in region r have it as the generalised region when issuing q.

DEFINITION 5. *Let $\gamma > 0$, $\langle u, whereis(u,t), t, q\rangle \in Q$ be a query and $\langle r, t, q\rangle \in Q'$ the corresponding generalised query. The issuer u is* γ-mutual information based anonymous *if for all $u' \in u\ell(r,t)$,*

$$I(U \mid q; \langle r,t,q\rangle) \leq \gamma \wedge f(\langle u', whereis(u',t), t, q\rangle) = \langle r,t,q\rangle$$

For short, we call γ-mutual information based anonymity γ-MIA.

5. GENERALISATION ALGORITHMS

In this section, we develop generalisation (or spatial cloaking) algorithms to compute regions satisfying users' privacy requirements in terms of the metrics presented in Sect. 4. As to find a region satisfying k-ABS is similar to compute a region satisfying k-anonymity on a given spatial distribution, we design an algorithm

for k-ABS by combining the algorithm grid [20] with the clustering algorithm K-Means [19]. For the other metrics, we design a uniform algorithm based on dichotomicPoints [20] with a newly developed function updateAS to update the intermediate regions.

5.1 An algorithm for k-ABS

To find an area that satisfies k-ABS, we have two main steps. The first is to obtain the spatial distribution of users who have similar probabilities to the issuer. The second step is to run a k-anonymity generalisation algorithm to find a region with at least k users based on the distribution computed at the first step.

The task of the first step can be transformed to the clustering problem. Given $q \in \mathcal{Q}$, we need to cluster the elements in $\mathcal{M}(q)$ such that the users with similar probabilities are grouped together. K-Means is the simplest learning algorithm to solve the clustering problem [19]. The number of clusters is fixed a priori. The main idea is to define K centroids, one for each cluster. In our algorithm, the K centroids are chosen uniformly in $[0, 1]$. Then the points (the elements in $\mathcal{M}(q)$ in our case) are associated to the nearest centroid, resulting in K clusters. The centers of these K clusters are updated as the new centroids. Afterwards, all points need to be binded to the current centroids. This process continues until the centroids remain unchanged between two consecutive iterations. In our case, K is chosen and fixed by the anonymiser. In fact, it defines 'similarity' in the definition of k-ABS in Sect. 4, i.e., ϵ. The larger K is, the smaller ϵ becomes.

For the second step, we use algorithm grid by Mascetti et al. [20] as it generates more regular regions with smaller area compared to others. A two-dimensional space is partitioned into a grid with $\lfloor \frac{N}{k} \rfloor$ cells each of which contains at least k users, where N denotes the number of users in $\mathcal{U}$. A user's position is represented by two dimensions x and y. The algorithm grid consists of two steps. First, users are ordered based on dimension x, and then on y. The ordered users are divided into $\lfloor \sqrt{\frac{N}{k}} \rfloor$ blocks of consecutive users. The block with the issuer enters the second step. The users in this block are then ordered first based on dimension y and then x. These users are also partitioned into $\lfloor \sqrt{\frac{N}{k}} \rfloor$ blocks. Then the block with the issuer is returned as the anonymity set. Details of the grid algorithm can be found in [20].

Alg. 1 describes our algorithm for k-ABS. In general, it takes the user requirement k and the number of clusters K defined by the anonymiser as inputs and gives the generalised region as output. Function K-Means returns the cluster of users with similar probabilities to that of u with respect to query q. Then the function grid outputs a subset of sim_users with at least k users who are located in the rectangular region. The generalised region is computed by function region.

Algorithm 1 A generalisation algorithm for k-ABS.

1: FUNCTION: kABS
2: INPUT: $\langle u, whereis(u,t), t, q\rangle, dis(t), \mathcal{M}(q), K, k$
3: OUTPUT: A region r that satisfies k-ABS
4:
5: sim_users :=K-Means$(u, q, K, \mathcal{M}(q))$;
6: $AS :=$ grid$(sim_users, dis(t), k)$;
7: $r :=$ region(AS)

Note that the clustering algorithm does not have to run each time when there is a query coming to the anoymiser. As long as the spatial distribution remains static or does not have big changes, for the queries received during this period, the anonymiser just executes the clustering algorithm once and returns the cluster containing the issuer as output of function K-Means directly.

In Alg. 1, K-Means can terminate in time $\mathcal{O}(N^{K+1} \log N)$ where N is the number of users [1]. The complexity of grid algorithm is $\mathcal{O}(\sqrt{kN} \log \sqrt{kN})$ [20]. Therefore, in general, the complexity of Alg. 1 is $\mathcal{O}(N^{K+1} \log N + \sqrt{kN} \log \sqrt{kN})$. The correctness of Alg. 2 is stated as Thm. 2.

THEOREM 2. *For any $\langle u, \ell, t, q\rangle \in Q$, Alg. 1 computes a generalised region in which the issuer u is k-approximate beyond suspicious.*

5.2 An algorithm for α-USI, β-EBA, γ-MIA

For privacy metrics α-USI, β-EBA, and γ-MIA, we design a uniform algorithm where users can specify which metric to use. Recall that in grid, the number of cells is pre-determined by k and the number of users. Thus it is not suitable to perform area generalisation for metrics without a predefined number k. Instead we use the algorithm dichotomicPoints to achieve our design goal.

The execution of dichotomicPoints involves multiple iterations in each of which users are split into two subsets. Similar to grid, positions are represented in two dimensions x and y, and users are also ordered based on their positions. However, different from grid the orders between axises are determined by the shape of intermediate regions rather than fixed beforehand. Specifically, if a region has a longer projection on dimension x, then x is used as the first order to sort the users. Otherwise, y is used as the first order. Users are then ordered based on the values of their positions on the first order axis and then the second order. Subsequently, users are partitioned into two blocks with the same or similar number of users along the first order axis. The block containing the issuer is taken into the next iteration. This process is repeated until any of the two blocks contains less than $2k$ users. This termination criterion is to ensure security against the outlier problem for k-anonymity (see Sect. 2).

However, in our uniform algorithm, instead of checking the number of users, we takes the satisfaction of users' privacy requirement as the termination criterion, e.g., if all users in the two blocks have a probability smaller than α. When issuing a query $q \in \mathcal{Q}$, the issuer u's privacy requirement $req(\langle u, whereis(u,t), t, q\rangle)$ consists of a chosen metric (i.e., USI, EBA, MIA) and its corresponding value (i.e., α, β, γ). For instance, if a user wants to hide in a set of users with a probability smaller than 20% for issuing a query, then his privacy requirement is specified as (USI, 20%).

In our uniform algorithm, after the determination of the first order axis, we call function updateAS. It takes a set of users and partitions them into two subsets along the first order axis, both of which should satisfy the issuer's privacy requirement and updateAS returns the one containing the issuer as the updated anonymity set. When it is not possible to partition users along the first order axis, i.e., one of the two blocks generatlised by any partition fails the issuer's requirement, the second order axis will be tried. If both tries have failed, updateAS simply returns the original set, which means no possible partition can be made with respect to the privacy requirement. In this situation, the whole algorithm terminates. Otherwise, the new set of users returned by update AS is taken into the next iteration.

Our uniform algorithm is described in Alg. 2. The boolean variable $cont$ is used to decide whether the algorithm should continue. It is set to $false$ when the set of users in AS does not satisfy the requirement (line 6) or when AS cannot be partitioned furthermore (line 26). In both cases, the algorithm terminates. The anonymity set AS is represented as a two-dimensional array. After ordering users in AS, $AS[i]$ consists of all users whose positions have the

Algorithm 2 The uniform generalisation algorithm for α-USI, β-EBA, and γ-MIA.

```
1: FUNCTION: uniformDP
2: INPUT: qu=⟨u, whereis(u,t), t, q⟩, req(qu), dis(t), M(q)
3: OUTPUT: Region r that satisfies req(qu)
4:
5: AS := U;
6: cont := check(AS, req(u));
7: while cont do
8:    min_x := min_{u'∈AS} whereis(u').x;
9:    min_y := min_{u'∈AS} whereis(u').y;
10:   max_x := max_{u'∈AS} whereis(u').x;
11:   max_y := max_{u'∈AS} whereis(u').y;
12:   if (max_x − min_x) ≥ (max_y − min_x) then
13:      first := x;
14:      second := y;
15:   else
16:      first := y;
17:      second := x;
18:   end if
19:   AS' = updateAS(AS, req(qu), first, dis(t), M(q));
20:   if AS' = AS then
21:      AS' = updateAS(AS, req(qu), second, dis(t), M(q));
22:   end if
23:   if AS' ≠ AS then
24:      cont := true;
25:   else
26:      cont := false;
27:   end if
28: end while
29: return region(AS);
```

same value on the first order axis. We use $\mathsf{len}(order)$ to denote the size of AS in the dimension denoted by $order$. For instance, in Fig. 2(a), axis x is the first order axis and $AS[3]$ has three users with the same x values. Moreover, $\mathsf{len}(first)$ is 6.

The function updateAS shown in Alg. 3 is critical for our algorithm uniformDP. It takes as input a set of users and outputs a subset that satisfies the issuer's privacy requirement $req(qu)$. It first orders the users and then divides them into two subsets with the same number of users along the first order axis (indicated by the variable $order$). This operation is implemented by the function $\mathsf{mid}(AS, order)$ which returns the middle user's index in the first dimension of AS. If both of the two subsets satisfy $req(qu)$, then the one containing the issuer is returned (implemented by function $\mathsf{part}(i, u)$). Otherwise, an iterative process is started. In jth iteration, the users are partitioned into two sets one of which contains the users in $AS[1], \ldots, AS[j]$ (denoted by $\mathsf{left}(j)$) and the other contains the rest (denoted by $\mathsf{right}(j)$). These two sets are checked against the privacy requirement $req(qu)$. If both $\mathsf{left}(j)$ and $\mathsf{right}(j)$ satisfy $req(qu)$, the one with issuer u is returned by $\mathsf{part}(j, u)$. If there are no partitions feasible after $\mathsf{len}(order)$ iterations, the original set of users is returned.

An example execution of Alg. 2 is shown in Fig. 2. The issuer is represented as a black dot. In Fig. 2(a) the users are first partitioned into two parts from the middle. Assume both parts satisfy $req(qu)$, so the set b_1 is returned as the anonymity set AS for the next iteration. As b_1's projection on axis y is longer, the first order is set to axis y (Fig. 2(b)). If after dividing the users from the middle, the set b_2 does not satisfy $req(qu)$. Thus, the users are partitioned from $AS[1]$ to $AS[4]$ (Fig. 2(c)). Suppose no partitions are feasible. The

Algorithm 3 The function updateAS.

```
1: FUNCTION: updateAS
2: INPUT: AS, req(qu), order, dis(t), M(q)
3: OUTPUT: AS' ⊆ AS that contains u and satisfies req(qu)
4:
5: AS := reorder(AS, order);
6: i := mid(AS, order);
7: if check(left(i), req(qu)) ∧ check(right(i), req(qu)) then
8:    AS := part(i, u);
9: else
10:   found := false;
11:   j := 0;
12:   while j ≤ len(order) ∧ ¬found do
13:      if check(left(j), req(qu)) ∧ check(right(j), req(qu))
         then
14:         found := true;
15:         AS := part(j, u);
16:      else
17:         j := j + 1;
18:      end if
19:   end while
20: end if
21: return AS;
```

first order axis is then switched to axis x. Function updateAS is called again to find a partition along axis x (Fig. 2(d)).

We can see Alg. 2 iterates for a number of times. In each iteration, some users are removed from the previous anonymity set. Operations such as partition and requirement check are time-linear in the size of the anonymity set. The number of iterations is logarithmic in the number of the users. So in the worst case, the time complexity of Alg. 2 is $\mathcal{O}(N \log N)$, where N denotes the number of all users in $\mathcal{U}$. The correctness of Alg. 2 is stated as Thm. 3.

THEOREM 3. *For any query $\langle u, \ell, t, q\rangle$, Alg. 2 computes a generalised region that satisfies the issuer u's privacy requirement $req(\langle u, whereis(u,t), t, q\rangle)$.*

Detailed proof of the theorem is given in the appendix.

6. EXPERIMENTAL RESULTS

We have performed an extensive experimental evaluation of the metrics presented in Sect. 4 using the algorithms in Sect. 5. The experiments are based on a dataset with 10,000 users' locations generated by the moving object generator developed by Brinkhoff [4]. Users' locations are scattered in the city of Oldenburg (Germany). As we focus on evaluating our generalisation algorithms, we randomly assign a priori probabilities to users although it is possible to generate user profiles as in [26] and calculate the probabilities using our methodology described in Sect. 3.

We implemented the algorithms using Java and experiments are run on a Linux laptop with 2.67Ghz Intel Core(TM) and 4GB memory. The results discussed in this section are obtained by taking the average of 100 simulations of the corresponding algorithms.

Through experiments, for all the proposed metrics we present the impact of the user specified parameters to the average area of generalised regions, in order to help users determine the right trade-off between privacy protection and the quality of services. Moreover, we illustrate the features of different metrics. In particular, we show that k-ABS gives a better protection than k-anonymity to users, who are potentially more likely to issue a query that others. The other metrics α-USI, β-EBA, γ-MIA are insensitive to users' a priori probabilities. Last, we show our algorithms are efficient

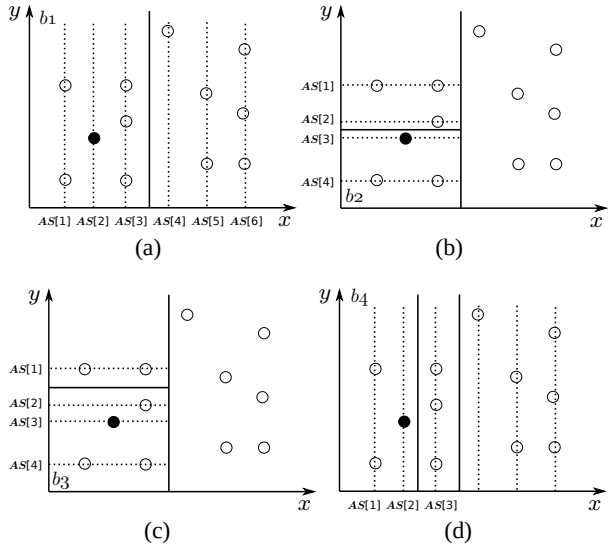

Figure 2: An example execution of our algorithm uniformDP

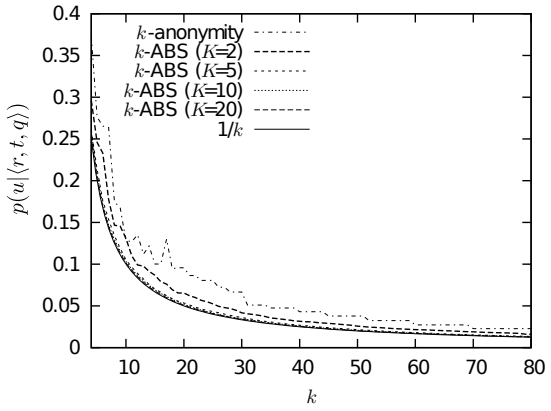

Figure 3: Posterior probabilities (k-ABS)

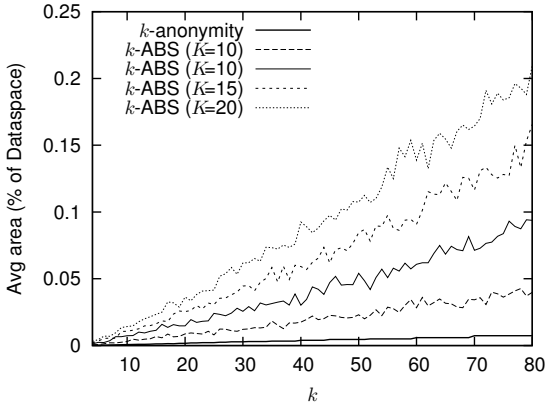

Figure 4: Area of generalised regions (k-ABS)

enough for practical applications which require real-time response by evaluating the average processing time.

6.1 k-ABS

We first address the comparison between k-anonymity and k-ABS and discuss the impact of the parameter K used in the clustering algorithm K-Means (see Alg. 1). In Fig. 3 we show how a user's a posteriori probability ($p(u \mid \langle r, t, q\rangle)$) changes with with respect to k and K. We have selected a user with a relatively high a priori probability so as to compare the performance of both metrics in protecting users who are more likely to issue the query.

First, the user's a posteriori probability decreases as k increases. This is because larger k means more users are in the generalised region. Second, for a given k, the issuer's a posteriori probability is normally larger than $\frac{1}{k}$ when k-anonymity is used, but closer to $\frac{1}{k}$ when k-ABS is adopted. This is because that in an anonymity set of k-anonymity, uses have larger differences among their a priori probabilities than the users in an anonymity of k-ABS. Third, in k-ABS, for a given k, the issuer's a posteriori probability is much closer to $\frac{1}{k}$ when more clusters are divided (i.e., bigger K). This can be explained by the observation that more clusters make the users in a cluster containing the issuer become more probable to be the same likely to issue the query.

Fig. 4 shows the average area of generalised regions by Alg. 1. In general, the area becomes larger when k increases. We can also observe that compared to k-anonymity, k-ABS has larger regions for a given value of k. Moreover, when k is fixed the area gets larger when K increases. These observations are all due to the fact that more clusters result in fewer users in each cluster, which in turn leads to larger regions to cover k users.

According to the above observations, the anonymiser can determine an appropriate value of K based on users' a priori distribution for a query (i.e. $\mathcal{M}(q)$) in order to balance users' query privacy and quality of service (smaller area, better quality).

6.2 α-USI

An issuer satisfies α-USI if from the attacker's view each user in the generalised region has a probability smaller than the specified value α to be the issuer.

In Fig. 5, we show that the average a posteriori probabilities of issuers with different a priori probabilities (indicated by lines marked 'high', 'medium' and 'low') and different specified values of α. We use a reference line to indicate the difference between users' requirement (α) and the result of Alg. 2. First, we find that users' a posteriori probabilities are always smaller than α, which shows the correctness of our algorithm. Second, for users with relatively high a priori probabilities, their a posteriori probabilities are closer to their requirements in terms of α. Meanwhile, for the users with low a priori probabilities, the value of α does not have a big influence on users' a posteriori probabilities. This can be explained by the definition of α-USI. A generalised region has to ensure that all users within it have a posteriori probabilities smaller than α (this is required to fix the outlier problem).

Fig. 6 shows changes of generalised regions' area along with α and the impact of users' a priori probabilities. The generalised regions become smaller as α increases. As we can see in Alg. 2, issuers' positions and α determine the generalised regions. Users' a priori probabilities have little impact on the generalisation process. This is also confirmed by experiments. In Fig. 6 users with different levels of a priori probabilities have regions with similar sizes.

Usually, for a given query users have an approximate estimation of their a priori probabilities compared to others, e.g., high or low. The above analysis enables users to estimate their a posteriori probabilities with regards to different values of α. This in turn helps them to choose an appropirate value for α to balance their query privacy and quality of service.

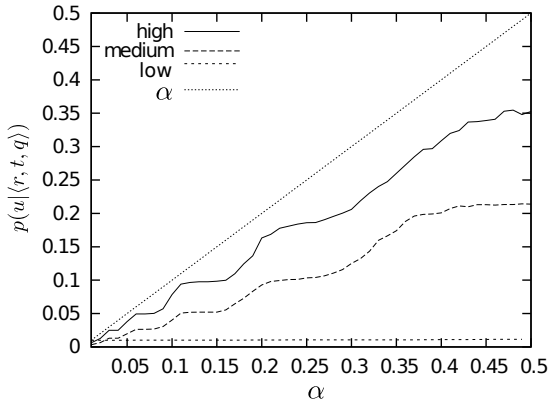

Figure 5: Posterior probabilities (α-USI)

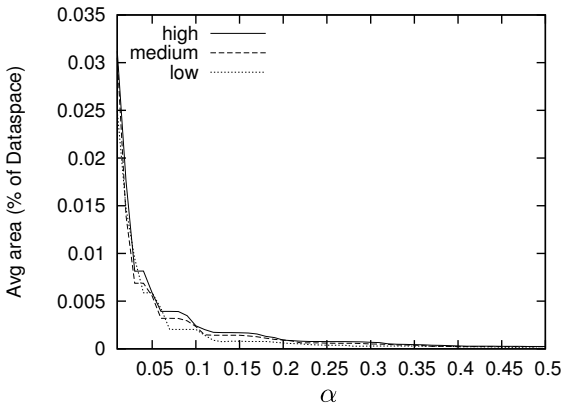

Figure 6: Area of generalised regions (α-USI)

6.3 β-EBA and γ-MIA

A generalised region that satisfies β-EBA ensures that the entropy over users in the region is larger than β, while γ-MIA ensures that the amount of uncertainty reduced is less than γ.

In Fig. 7 and Fig. 9, we show that the entropies and mutual information corresponding to the generalised regions by our algorithm satisfy the definitions of β-EBA and γ-MIA. We can observe that users' a priori probabilities do not have impact on the outputs – the two lines for users with high and low a priori probabilities almost coincide. Similar to α-USI, this is because a generalised region is determined by the values β or γ and issuers' positions rather than their a priori probabilities. The values of entropy (resp. mutual information) change sharply when β (resp. γ) is getting close to integers, this is due to the nature of entropy. Similarly, we show how the average area of generalised regions changes along with β and γ in Fig. 8 and Fig. 10, respectively – the area usually gets doubled when β and γ are increased by one.

6.4 Performance analysis

We illustrate the performance of Alg. 1 through Fig. 12. Although the clustering algorithm needs to run only once for a spatial distribution for a given K, we execute it for each query instead in order to test the performance in the worst case when there happens to be only one query issued. As algorithm K-Means has a complexity depending on K, for a given k the computation time increases when K becomes larger. When $K = 5$, the average computation time is about 140ms while it is around 250ms when $K = 20$.

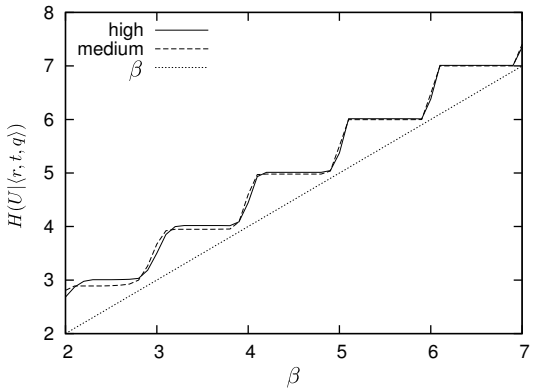

Figure 7: Entropies (β-EBA)

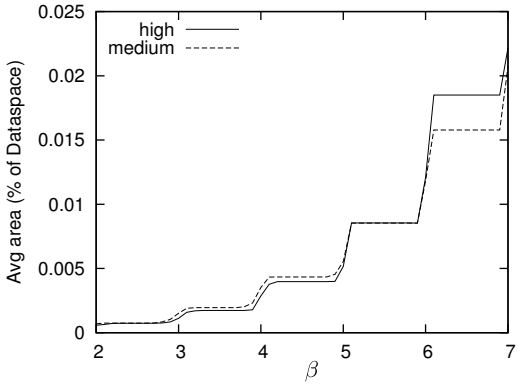

Figure 8: Area of generalised regions (β-EBA)

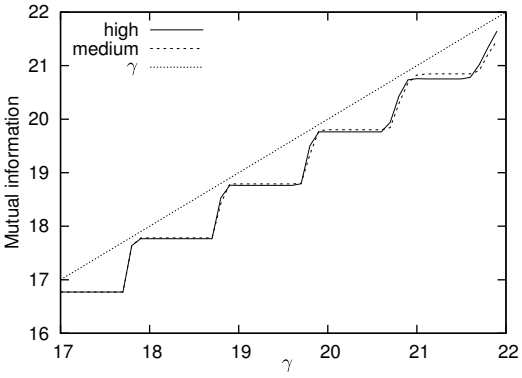

Figure 9: Mutual information (γ-MIA)

Fig. 11 shows the average computation time of Alg. 2 (for α-USI and β-EBA) and grid (for k-anonymity). In this figure, we use a normalised value $norm$ to compare the performance for different metrics: $norm = k$ for k-anonymity, while $norm = 1/\alpha$ for α-USI and $norm = 2^{\beta}$ for β-EBA, respectively. The computation time of β-EBA (11 – 12ms) is a bit larger than α-USI (about 10ms) because computing entropy is a bit more complex. Furthermore, as $norm$ increases, more time is needed for β-EBA. This is also determined by Alg. 3, where larger k leads to more time to traverse the region in order to find a possible partition. The implementation for γ-MIA is based on the calculation of entropies, so in general the computation time of γ-MIA is almost same as β-EBA.

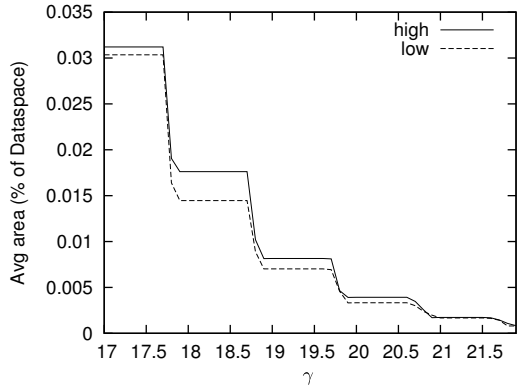

Figure 10: Area of generalised regions (γ-MIA)

We can observe that the computation time of algorithm grid is linear with k (see k-anonymity in Fig. 11), which confirms the results in [20]. However, due to the complexity of the clustering algorithm K-Means used in Alg. 1, the impact of k is not obvious in Fig. 12.

There exist a few ways to improve the efficiency of our implementations such as using a better data structure and reducing redundant computation. With powerful servers deployed in practice, our roposed generalisation algorithms are efficient enough to handle concurrent queries and give real-time responses.

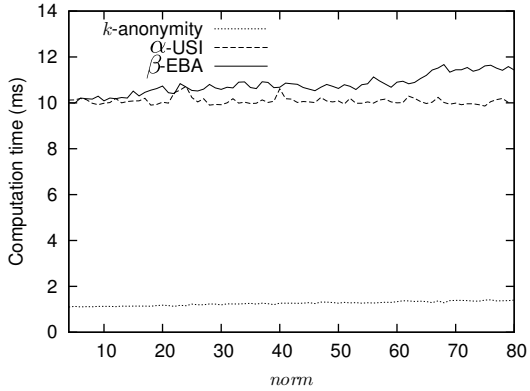

Figure 11: Computation time of the algorithms.

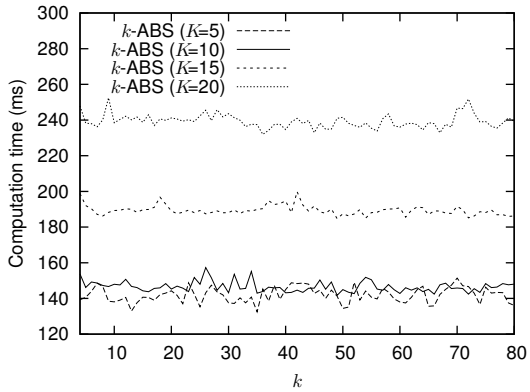

Figure 12: Computation time of the algorithms.

7. CONCLUSION

In this paper, we consider a powerful attacker who can obtain user profiles and has access to users' real-time positions in the context of LBSs. Assuming this stronger attacker model, we propose new metrics to correctly measure users' query privacy in LBSs, including k-ABS, α-USI, β-EBA and γ-MIA. For information theory based metrics, the determination of users' specified values is not intuitive. However, users can use other metrics as references. For instance, k-anonymity corresponds to $\log k$-EBA when the distribution for users to issue a query is (close to) uniform. Spacial generalisation algorithms are developed to compute regions satisfying user's privacy requirements specified in the proposed metrics. Extensive experiments show our metrics are effective in balancing privacy and quality of service in LBSs and the algorithms are efficient to meet the requirement of real-time responses.

Our metrics are not exhaustive, and there exist other ways to express query privacy. For instance, we can use min-entropy to express information leakage [31] in a way analogous to mutual information: $I_\infty(X;Y) = H_\infty(X) - H_\infty(X \mid Y)$. Intuitively, it measure the amount of min-entropy reduced after the attacker has observed a generalised query. It is very interesting to study differential privacy [12] to see how it can be adopted for LBS scenarios.

In future, we want to develop an application for an LBS, making use of the proposed metrics to protect users' query privacy. This can lead us to a better understanding of privacy challenges in more realistic situations. The implementation of our algorithms can also be improved as well, e.g., using a better clustering algorithm for kABS. Another interesting direction is to study a more stronger attacker model, where the attacker, for instance, can have access to mobility patterns of users.

8. REFERENCES

[1] D. Arthur, B. Manthey, and H. Röglin. k-Means has polynomial smoothed complexity. In *Proc. 50th Symposium on Foundations of Computer Science (FOCS)*, pp. 405–414. IEEE CS, 2009.

[2] A. R. Beresford. *Location privacy in ubiquitous computing*. PhD thesis, University of Cambridge, 2005.

[3] O. Berthold, A. Pfiztmann, and R. Standtke. The disadvantages of free mix routes and how to overcome them. In *Proc. Workshop on Design Issues in Anonymity and Unobservability*, *LNCS* 2009, pp. 30–45. Springer, 2000.

[4] T. Brinkhoff. A framework for generating network-based moving objects. *GeoInformatica*, 6(2):153–180, 2002.

[5] K. Chatzikokolakis, C. Palamidessi, and P. Panangaden. Anonymity protocols as noisy channels. *Information and Computation*, 206(2-4):378–401, 2008.

[6] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1(1):65–75, 1988.

[7] R. Cheng, Y. Zhang, E. Bertino, and S. Prabhakar. Preserving user location privacy in mobile data management infrastructures. In *Proc. 6th Workshop on Privacy Enhancing Technologies (PET)*, *LNCS* 4258, pp. 393–412, 2006.

[8] C.-Y. Chow, M. F. Mokbel, and W. G. Aref. Casper*: Query processing for location services without compromising privacy. *ACM Transactions on Database Systems*, 34(4):1–48, 2009.

[9] M. R. Clarkson, A. C. Myers, and F. B. Schneider. Quantifying information flow with beliefs. *Journal of Computer Security*, 17(5):655–701, 2009.

[10] Y. Deng, J. Pang, and P. Wu. Measuring anonymity with relative entropy. In *Proc. 4th Workshop on Formal Aspects in Security and Trust (FAST)*, *LNCS* 4691, pp. 65–79. Springer, 2007.

[11] C. Díaz, S. Seys, J. Claessens, and B. Preneel. Towards measuring anonymity. In *Proc. 2nd Workshop on Privacy Enhancing Technologies (PET)*, *LNCS* 2482, pp. 54–68. Springer, 2003.

[12] C. Dwork. Differential privacy. in *Proc. 33rd International Colloquium on Automata, Languages and Programmin (ICALP)*, *LNCS* 4052, pp. 1-12. Springer, 2006.

[13] J. Freudiger, R. Shokri, and J. P. Hubaux. Evaluating the privacy risk of location-based services. In *Proc. 15th Conference on Financial Cryptography and Data Security (FC)*, *LNCS*. Springer, 2011.

[14] B. Gedik and L. Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Transactions on Mobile Computing*, 7(1):1–18, 2008.

[15] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proc. 1st Conference on Mobile Systems, Applications, and Services (MobiSys)*. USENIX, 2003.

[16] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias. Preventing location-based identity inference in anonymous spatial queries. *IEEE Transactions on Knowledge and Data Engineering*, 19(12):1719–1733, 2007.

[17] J. M. Kleinberg. Challenges in mining social network data: processes, privacy, and paradoxes. In *Proc. 13th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 4–5. ACM, 2007.

[18] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Proc. 23rd Conference on Data Engineering (ICDE)*, pp. 106–115. IEEE CS, 2007.

[19] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pp. 281–297. University of California, 1967.

[20] S. Mascetti, C. Bettini, D. Freni, and X. S. Wang. Spatial generalization algorithms for LBS privacy preservation. *Journal of Location Based Services*, 1(3):179–207, 2007.

[21] M. F. Mokbel, C.-Y. Chow, and W. G. Aref. The new casper: A privacy-aware location-based database server. In *Proc. 23rd Conference on Data Engineering (ICDE)*, pp. 1499–1500. IEEE CS, 2007.

[22] M. K. Reiter and A. D. Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.

[23] A. Rényi. On measures of entropy and information. In *Proc. 4th Berkeley Symposium on Mathematics, Statistics, and Probability*, pp. 547–561. University of California, 1961.

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

[25] A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In *Proc. 2nd Workshop on Privacy Enhancing Technologies (PET)*, *LNCS* 2482, pp. 41–53. Springer, 2003.

[26] H. Shin, V. Atluri, and J. Vaidya. A profile anonymization model for privacy in a personalized location based service environment. In *Proc. 9th International Conference on Mobile Data Management (MDM)*, pages 73–80. IEEE CS, 2008.

[27] R. Shokri, J. Freudiger, M. Jadliwala, and J.-P. Hubaux. A distortion-based metric for location privacy. In *Proc. 2009 ACM Workshop on Privacy in the Electronic Society (WPES)*, pp. 21–30. ACM, 2009.

[28] R. Shokri, G. Theodorakopoulos, J.-Y. L. Boudec, and J.-P. Hubaux. Quantifying location privacy. In *Proc. 32nd IEEE Symposium on Security and Privacy (S&P)*. IEEE CS, 2011.

[29] R. Shokri, G. Theodorakopoulos, G. Danezis, and J.-P. Hubaux. Quantifying location privacy: The case of sporadic location exposure. In *Proc. 11th Privacy Enhancing Technologies Symposium (PETS)*, 2011.

[30] R. Shokri, C. Troncoso, C. Díaz, J. Freudiger, and J.-P. Hubaux. Unraveling an old cloak: k-anonymity for location privacy. In *Proc. 2010 ACM Workshop on Privacy in the Electronic Society (WPES)*, pp. 115–118. ACM, 2010.

[31] G. Smith. On the foundations of quantitative information flow. in *Proc. 12th International Conference on Foundations of Software Science and Computation Structures (FOSSACS)*, *LNCS* 5504, pp. 288–302. Springer, 2009.

[32] K. W. Tan, Y. Lin, and K. Mouratidis. Spatial cloaking revisited: Distinguishing information leakage from anonymity. In *Proc. 11th Symposium on Spatial and Temporal Databases (SSTD)*, *LNCS* 5644, pp. 117–134. Springer, 2009.

[33] X. Wu, X. Ying, K. Liu, and L. Chen. A survey of algorithms for privacy-preservation of graphs and social networks. In *Managing and Mining Graph Data*, pp. 421–442, 2009.

[34] M. Xue, P. Kalnis, and H. K. Pung. Location diversity: Enhanced privacy protection in location based services. In *Proc. 4th Symposium on Location and Context Awareness (LoCA)*, *LNCS* 5561, pp. 70–87. Springer, 2009.

[35] Y. Zhu and R. Bettati. Anonymity vs. information leakage in anonymity systems. In *Proc. 25th Conference on Distributed Computing Systems (ICDCS)*, pp. 514–524. IEEE CS, 2005.

APPENDIX

A. PROOF OF THM. 3

PROOF. By Def. 3, Def. 4 and Def. 5, Alg. 2 computes a region r for a query $\langle u, whereis(u,t), t, q\rangle$ that satisfies a constraint related to the issuer's a posteriori probability, entropy, or mutual information. Furthermore, for any $u' \in u\ell(r)$, the algorithm computes the same region. We take α-USI as an example to show the correctness of our algorithm and the proofs of the other two are analogous.

By Def. 3, we have to prove (1) the a posteriori probability of user u is smaller than α, i.e., $p(u \mid \langle r,t,q\rangle) \leq \alpha$; (2) for any $u' \in u\ell(r)$, $f(\langle u', whereis(u',t),t,q\rangle) = \langle r,t,q\rangle$.

(1) At the line 5 of Alg. 2, we set AS to the original user set $\mathcal{U}$ and the algorithm continues only if $\mathcal{U}$ satisfies the issuer's requirement $req(\langle u, whereisu, t, q\rangle)$. Otherwise, it is impossible to return a region satisfying the requirement. The set AS is only reassigned to another set when a partition is made (line 8 or line 15 in Alg. 3). The two sets by the partition satisfy the requirement and the one containing the issuer is assigned to AS. Thus, it is guaranteed that the final region r ensures $p(u\,|\,\langle r,t,q\rangle) \leq \alpha$.

(2) Let u' be any user in the generalised region r of Alg. 2. Let AS_j and AS'_j be the values of AS in the jth iteration of Alg. 2 of u and u', respectively. We show that $AS_j = AS'$ by induction on the number of iterations, i.e. j.

INDUCTION BASIS: Initially, we suppose $\mathcal{U}$ meets the requirement. Then we have $AS_1 = AS'_1$.

INDUCTION STEP: Assume at jth iteration $AS_j = AS'_j$. We have to show that the algorithm either terminates with AS_j and AS'_j, or enter the next iteration with $AS_{j+1} = AS'_{j+1}$. The equality that $AS_j = AS'_j$ is followed by that $\mathsf{mid}(AS_j, order) = \mathsf{mid}(AS'_j, order)$. There are three possible executions.

Case 1: if $left(i)$ and $right(i)$ of AS_j and AS'_j satisfy the requirements (line 7 of Alg. 3), the part containing the issuer is returned. Thus AS_{j+1} contains u as well as all other users in $u\ell(r)$, including u'. Thus, $AS_{j+1} = AS'_{j+1}$.

Case 2: if the check at line 7 of Alg. 3 fails, then the algorithm switches to find from the beginning the first feasible partition. Suppose the partition is made at the position x for AS_j. Then x is also the right position for AS'_j as $AS_j = AS'_j$. Because of the similar reason in the previous possible execution, the same subset is set to AS_{j+1} and AS'_{j+1}. Thus, $AS_{j+1} = AS'_{j+1}$.

Case 3: if there are no possible partitions, Alg. 3 returns AS_{j+1} and AS'_{j+1} in both cases. Then the first order is changed and Alg. 3 is called again. If one of the first two execution is taken, with the analysis above, we have $AS_{j+1} = AS'_{j+1}$. Otherwise, Alg. 2 terminates with $\mathsf{region}(AS_j)$ and $\mathsf{region}(AS'_j)$ which are equal. □