# Anonymity and Historical-Anonymity in Location-Based Services

Claudio Bettini[1], Sergio Mascetti[1], X. Sean Wang[2],
Dario Freni[1], and Sushil Jajodia[3]

[1] EveryWare Lab - DICo, Università degli Studi di Milano, Italy
{bettini,mascetti,freni}@dico.unimi.it
[2] Department of CS, University of Vermont, VT, USA
sean.wang@uvm.edu
[3] Center for Secure Information Systems, George Mason University, VA, USA
jajodia@gmu.edu

**Abstract.** The problem of protecting user's privacy in Location-Based Services (LBS) has been extensively studied recently and several defense techniques have been proposed. In this contribution, we first present a categorization of privacy attacks and related defenses. Then, we consider the class of defense techniques that aim at providing privacy through anonymity and in particular algorithms achieving "historical k-anonymity" in the case of the adversary obtaining a trace of requests recognized as being issued by the same (anonymous) user. Finally, we investigate the issues involved in the experimental evaluation of anonymity based defense techniques; we show that user movement simulations based on mostly random movements can lead to overestimate the privacy protection in some cases and to overprotective techniques in other cases. The above results are obtained by comparison to a more realistic simulation with an agent-based simulator, considering a specific deployment scenario.

## 1 Introduction

Location-based services (LBS) have recently attracted much interest from both industry and research. Currently, the most popular commercial service is probably car navigation, but many other services are being offered and more are being experimented, as less expensive location aware devices are reaching the market. Consciously or unconsciously, many users are ready to give up one more piece of their private information in order to access the new services. Many other users, however, are concerned with releasing their exact location as part of the service request or with releasing the information of having used a particular service [1]. To safeguard user privacy while rendering useful services is a critical issue on the growth path of the emerging LBS.

An obvious defense against privacy threats is to eliminate from the request any data that can directly reveal the issuer's identity, possibly using a pseudonym whenever this is required (e.g., for billing through a third party). Unfortunately,

C. Bettini et al. (Eds.): Privacy in Location-Based Applications, LNCS 5599, pp. 1–30, 2009.

simply dropping the issuer's personal identification data may not be sufficient to anonymize the request. For example, the location and time information in the request may be used, with the help of external knowledge, to restrict the possible issuer to a small group of users. This problem is well-known for the release of data in databases tables [2]. In that case, the problem is to protect the association between the identity of an individual and a tuple containing her sensitive data; the attributes whose values could possibly be used to restrict the candidate identities for a given tuple are called *quasi-identifiers* [3,4].

For some LBS, anonymity may be hard to achieve and alternative approaches have been proposed, including obfuscation of sensitive information and the use of private information retrieval (PIR) techniques. For example, sensitive service parameters (possibly including location) can be generalized, partly suppressed, transformed, or decomposed using multiple queries in order to obfuscate their real precise value, while preserving an acceptable quality of service.

While the main goal of this contribution is to illustrate anonymity-based privacy protection techniques, the first two sections are devoted to a categorization of LBS privacy attacks, and to the classification of the main proposed defense techniques, including private information obfuscation and PIR, according to the threats they have been designed for, and according to other general features. This contribution does not discuss techniques aimed to the *off-line* anonymization of sets of trajectories (as in [5]), but only on techniques that are incrementally applied to service requests at the time they are issued. In Section 4, we focus on anonymity-based approaches and we show how historical $k$-anonymity can be achieved when an adversary has the ability to recognize sequences of requests by the same issuer. In Section 5, we report an experimental evaluation of anonymization algorithms showing the impact of realistic user movement simulations in these evaluations. Section 6 identifies some interesting research directions, and Section 7 concludes the chapter.

## 2 A Classification of Attacks to LBS Privacy

There is a privacy threat whenever an adversary is able to associate the identity of a user to information that the user considers private. In the case of LBS, this *sensitive association* can be possibly derived from location-based requests issued to service providers. More precisely, the identity and the private information of a single user can be derived from requests issued by a group of users as well as from available background knowledge. Figure 1 shows a graphical representation of this general privacy threat in LBS.

A *privacy attack* is a specific method used by an adversary to obtain the sensitive association. Privacy attacks can be divided into categories mainly depending on several parameters that characterize the *adversary model*. An adversary model has three main components: a) the target private information, b) the ability to obtain the messages exchanged during service provisioning, and c) the *background knowledge* and the *inferencing abilities* available to the adversary.

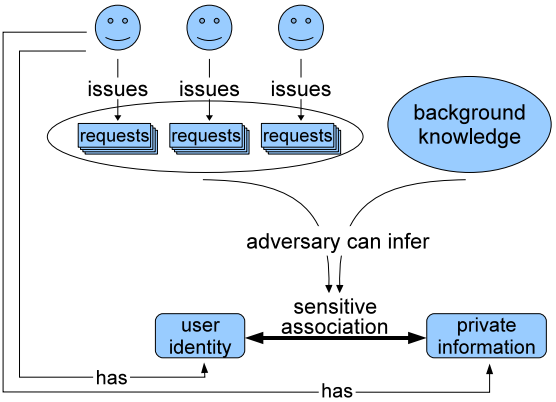



**Fig. 1.** General privacy threat in LBS

The target private information is the type of information that the adversary would like to associate with a specific individual, like e.g., her political orientation, or, more specifically, her location. Different classes of adversaries may also have different abilities to obtain the messages exchanged with the service provider, either by eavesdropping the communication channels or by accessing stored data at the endpoints of the communication. This determines, for example, the availability to the adversary of a single message or multiple messages, messages from a specific user or from multiple users, etc.. Finally, the adversary may have access to external knowledge, like e.g., phone directories, lists of members of certain groups, voters lists, and even presence information for certain locations, and may be able to perform inferences, like joining information from messages with external information as well as more involved reasoning. For example, even when a request does not explicitly contain the sensitive association (e.g., by using pseudo-identifiers to avoid identification of the issuer), the adversary may re-identify the issuer by joining location data in the request with presence data from external sources.

Regarding background knowledge, two extreme cases can be considered. When no background knowledge is available, a privacy threat exists if the sensitive association can be obtained only from the messages in the service protocol. When "complete" background knowledge is available, the sensitive association is included and the privacy violation occurs independently from the service request.

Hence, privacy attacks should not only be categorized in terms of the target private information, and of the availability to the adversary of service protocol messages (the first two of the main components mentioned above), but also in terms of the available background knowledge and inferencing abilities. In the following, we list some categories of privacy attacks specifically enabled by background knowledge.

- Attacks exploiting *quasi-identifiers* in requests;
- Snapshot versus historical attacks;

– Single- versus multiple-issuer attacks;
– Attacks exploiting knowledge of the defense;

Each category is discussed in the rest of this section.

### 2.1 Attacks Exploiting *Quasi-Identifiers*

Either part of the sensitive association can be discovered by joining information in a request with external information. When we discover the identity of the issuer (or even restrict the set of candidate issuers) we call the part of the request used in the join *quasi-identifier*. For example, when the location data in the request can be joined with publicly available presence data to identify an individual, we say that location data act as quasi-identifier. Similarly to privacy preserving database publication, the recognition of what can act as quasi-identifier in service request is essential to identify the possible attacks (as well as to design appropriate defenses).

### 2.2 Snapshot versus Historical Attacks

Most of the approaches presented so far in the literature [6,7,8,9] have proposed techniques to ensure a user's privacy in the case in which the adversary can acquire a single request issued by that user. More specifically, these approaches do not consider attacks based on the correlation of requests made at different time instants. An example are attacks exploiting the ability of the adversary to *link* a set of requests, i.e., to understand that the requests have been issued by the same (anonymous) user.

When historical correlation is ignored, we say that the corresponding threats are limited to the *snapshot case*. Intuitively, it is like the adversary can only obtain a snapshot of the messages being exchanged for the service at a given instant, while not having access to the complete history of messages.

In contrast with the snapshot case, in the *historical case* it is assumed that the adversary is able to *link* a set of requests. Researchers [10,11] have considered such a possibility. Several techniques exist to *link* different requests to the same user, with the most trivial ones being the observation of the same identity or pseudo-identifier in the requests, and others being based on spatiotemporal correlations. We call *request trace* a set of requests that the adversary can correctly associate to a single user. More dangerous threats can be identified in contexts characterized by the historical case as explained in [12].

### 2.3 Single versus Multiple-Issuer Attacks

When the adversary model limits the requests that can be obtained to those being issued by a single (anonymous) user, we say that all the attacks are *single-issuer attacks*. When the adversary model admits the possibility that multiple requests from multiple users are acquired, and the adversary is able to

understand if two requests are issued by different users, we have a new important category of attacks, called *multiple-issuer attacks*. Note that this is an orthogonal classification with respect to snapshot and historical. Example 1 shows that, in the multiple-issuer case, an adversary can infer the sensitive association for a user even if the identity of that user is not revealed to the adversary.

*Example 1.* Suppose Alice issues a request $r$ and that the adversary can only understand that the issuer is one of the users in a set $S$ of potential issuers. However, if all of the users in $S$ issue requests from which the adversary can infer the same private information inferred from $r$, then the adversary can associate that private information to Alice as well.

In the area of privacy in databases, this kind of attack is known as *homogeneity attack* [13]. In LBS, differently from the general case depicted in Figure 1), in the snapshot, multiple-issuer case, a single request for each user in a group is considered. More involved and dangerous threats can occur in the historical, multiple-issuer case.

### 2.4 Attacks Exploiting Knowledge of the Defense

In the security research area, it is frequently assumed that the adversary knows the algorithms used for protecting information, and indeed the algorithms are often released to the public. We have shown [14] that the first proposals for LBS privacy protection ignored this aspect leading to solutions subject to so called *inversion* attacks. As an example of these attacks, consider spatial cloaking as a defense technique, and suppose that a request with a certain cloaked region is observed by the adversary. Suppose also that he gets to know the identity of the four potential issuers of that request, since he knows who was in that region at the time of the request; Still he cannot identify who, among the four, is the actual issuer, since cloaking has been applied to ensure 4-anonymity. However, If he knows the cloaking algorithm, he can simulate its application to the specific location of each of the candidates, and exclude any candidate for which the resulting cloaked region is different from the one in the observed request. Some of the proposed algorithms are indeed subject to this attack. Kalnis et al. [8] show that each generalization function satisfying a property called *reciprocity* is not subject to the inversion attack. In our chapter, depending on the assumption in the adversary model about the knowledge of the defense algorithm we distinguish *def-aware attacks* from *def-unaware attacks*.

## 3 Defenses to LBS Privacy Threats

Defense techniques can be categorized referring to the attacks' classification reported above, depending on which specific attacks they have been designed for. However, there are other important criteria to distinguish defense approaches:

1. Defense technique: Identity anonymity versus private information obfuscation versus encryption
2. Defense architecture: Centralized versus decentralized
3. Defense validation: Theoretical versus experimental.

The different defense techniques can be classified as *anonymity-based* if they aim at protecting the association between an individual and her private information by avoiding the re-identification of the individual through a request (or a sequence of requests). This is achieved by transforming the parts of the *original request* acting as quasi-identifiers to obtain a *generalized request*. On the contrary, techniques based on *private information obfuscation* aim to protect the same association by transforming the private information contained in the original request, often assuming that the identity of the individual can be obtained. Finally, *encryption-based* techniques use private information retrieval (PIR) methods that can potentially protect both the identity of the issuer and the private information in the request.

Centralized defense architectures assume the existence of one or more trusted entities acting as a proxy for service requests and responses between the users and the service providers. The main role of the proxy is to transform requests and possibly responses according to different techniques in order to preserve the privacy of the issuers. Decentralized architectures, on the contrary do not assume intermediate entities between users and service providers. Among the benefits of centralized architectures are a) the ability of the proxy to use information about a group of users (e.g., their location) in order to more effectively preserve their privacy, and b) the availability of more computational and communication resources than the users' devices. The main drawbacks are considered the overheads in updating on the proxy the information about the users, and the need for the user to trust these entities.

A third criteria to distinguish the defenses that have been proposed is the validation method that has been used. In some cases, formal results, based on some assumptions, have been provided so that a certain privacy is guaranteed in all scenarios in which the assumptions hold. In other cases, only an experimental evaluation, usually based on synthetic data, is provided. It will be clear later in this contribution that this approach may be critical if the actual service deployment environment does not match the one used in the evaluation.

In this section we classify the main proposals appeared in the literature according to this categorization.

### 3.1 Anonymity Based Defenses

Most of the techniques proposed in the LBS literature to defend privacy through anonymity consider the location as a quasi-identifier. Indeed, it is implicitly or explicitly assumed that background knowledge can in some cases lead an adversary to infer the identity of the issuer given her location at a given time. Consequently, the target private information for the considered attacks is usually

the specific service being requested, or the location of the issuer whenever that location cannot be used as quasi-identifier.[1]

When the location acts as a quasi-identifier, the defense technique transforms the location information in the original request into a *generalized location*. In the following we call *anonymity set* of a generalized request, the set of users that, considering location information as quasi-identifier, are not distinguishable from the issuer.

**Centralized Defenses against Snapshot, Single-Issuer and Def-Unaware Attacks.** Anonymity based defenses with centralized architectures assume the existence of a trusted proxy that is aware of the movements of a large number of users. We call this proxy Location-aware Trusted Server (LTS).

The first generalization algorithm that appeared in the literature is named *IntervalCloaking* [7]. The paper proposes to generalize the requests along the spatial and/or temporal dimension. For what concerns the spatial dimension, the idea of the algorithm is to iteratively divide the total region monitored by the LTS. At each iteration the current area $q_{prev}$ is partitioned into quadrants of equal size. If less than $k$ users are located in the quadrant $q$ where the issuer of the request is located, then $q_{prev}$ is returned. Otherwise, iteration continues considering $q$ as the next area. For what concerns the temporal dimension, the idea is to first generalize the spatial location (with the above algorithm) at a resolution not finer than a given threshold. Then, the request is delayed until $k$ users pass through the generalized spatial location. This defense algorithm has only been validated through experimental results.

An idea similar to the spatial generalization of *IntervalCloaking* is used by Mokbel et al. [9] that propose *Casper*, a framework for privacy protection that includes a generalization algorithm. The main difference with respect to *IntervalCloaking* is that, in addition to the anonymity parameter $k$, the user can specify the minimum size of the area that is sent to the SP. While it is not explicit in the paper, the idea seems to be that, in addition to $k$-anonymity, the algorithm also provides a form of location obfuscation. Similarly to *IntervalCloaking*, *Casper* has been validated through experimental results.

**Centralized Defenses against Snapshot, Single-Issuer and Def-Aware Attacks.** Many papers extend *IntervalCloaking* to provide defenses techniques that guarantee anonymity when more conservative assumptions are made for the adversary model. Kalnis et al. [8], propose the *Hilbert Cloak* algorithm that provides anonymity also in the case in which the adversary knows the generalization function. The idea of *Hilbert Cloak* is to exploit the Hilbert space filling curve to define a total order among users' locations. Then, *Hilbert Cloak* partitions the users into blocks of $k$: the first block from the user in position 0 to the user in position $k-1$ and so on (note that the last block can contain up to $2 \cdot k - 1$ users). The algorithm then returns the *minimum bounding rectangle* (MBR) computed considering the position of the users that are in the same block

[1] Indeed, location cannot be the target private information when it can be found explicitly associated with identities in background knowledge.

as the issuer. The correctness of the *Hilbert Cloak* algorithm is formally provided and the performance of the algorithm has been also experimentally evaluated.

A different algorithm, called *CliqueCloak* is proposed by Gedik et al. [15]. The main difference with respect to the *IntervalCloaking* algorithm is that *CliqueCloak* computes the generalization among the users that actually issue a request and not among the users that are potential issuers. Indeed, *CliqueCloak* collects original requests without forwarding them to the SP until it is possible to find a spatiotemporal generalization that includes at least $k$ pending requests. Then, the requests are generalized and forwarded to the SP. The advantage of the proposed technique, whose correctness is formally proved, is that it allows the users to personalize the degree of anonymity as well as the maximum tolerable spatial and temporal generalizations. However, the algorithm has high computational costs and it can be efficiently executed only for small values of $k$.

In [14] Mascetti et al. present other three generalization algorithms that are proved to guarantee anonymity against snapshot, single-issuer and def-aware attacks. The aim is to provide anonymity while minimizing the size of the generalized location. The algorithm with the best performance with respect to this metric is called *Grid*. Intuitively, this algorithm partitions all users according to their position along one dimension. Then, it considers the users in the same block as the issuer and it partitions them according to their location along the other dimension. Finally, each block has at least cardinality $k$ and the algorithm computes the generalized location as the minimum bounding rectangle (MBR) that covers the location of the users in the same block as the issuer.

**Decentralized Defenses against Snapshot, Single-Issuer Attacks.** Some papers propose defense techniques that do not require a centralized architecture. Chow et al. [16] propose a decentralized solution called *CloakP2P* in which it is assumed that users can communicate with each other using an ad-hoc network. Basically, before sending the request, a user looks for the $k-1$ closest users in the neighborhood through the ad-hoc network. The location information of the request is then generalized to the region containing these users and the request is issued to the server through one of these users that is randomly selected. This algorithm guarantees privacy only against def-unaware attacks and it is evaluated through experimental results only.

*Privè* is a distributed protocol based on the *Hilbert Cloak* algorithm ([17]). In this case, the data structure that contains the positions of the users on the Hilbert curve is a $B^+$-tree that is distributed among the users in the system. The generalization is a distributed algorithm that traverses the tree starting from the root and finds the set of users containing the issuer. The algorithm is proven to be correct and guarantees privacy also against def-aware attacks. However, this solution suffers from some scalability issues. To address these issues, Ghinita et al. [18] propose the *MobiHide* algorithm which improves the scalability but that does not guarantee anonymity if the generalization algorithm is known to the adversary. The algorithm is formally validated.

A different decentralized solution is proposed by Hu et al. [19]. The main characteristic of the proposed technique is that it does not require the users to

disclose their locations during the anonymization process. Indeed, it is assumed that a user's devices is able to measure the closeness from its peers through its omnidirectional antenna (using WiFi signal, for example). When a request is generalized, the distance information is used to compute the anonymity set and the generalized location is obtained through a secure computation among the users in the anonymity set. The proposed approach is safe against def-aware attacks and its correctness is formally proved.

**Centralized Defenses against Historical, Single-Issuer Attacks.** Several papers further extend the ideas of *IntervalCloaking* to provide a defense in the historical case. The problem of anonymity in the historical, single-issuer case has been first investigated in [12]. In the paper it is shown that the defense technique for the snapshot case cannot be straightforwardly applied to provide protection against a historical attack. In addition, a centralized algorithm is proposed. The model proposed in the paper is used in this contribution and is presented in details in Section 4.

Following the main ideas presented in [12] other anonymization techniques for the historical case have been proposed in [20,21]. The work in [20] also aims at providing protection against a def-aware attack, however it is not clear if the proposed algorithm achieves this goal since it is only evaluated through experimental results. The work in [21] proposes two generalization algorithms, the first one, called *plainKAA*, exploits the same general idea presented in [12]. The second one is an optimization of the first, based on the idea that in the generalization of the requests the users that were not in the anonymity set of a previous request can contribute to anonymity protection. It is unclear if this optimization can preserve historical $k$-anonymity. Both algorithms are validated through experimental results only.

Mascetti et al. propose a formal model for the historical case [22] and experimentally show that, under certain conservative assumptions, it is not possible to guarantee anonymity without generalizing the user locations to large areas. Under these assumptions, considered in most of the related work on the snaphot case, the adversary knows the association between each user identity and the location of that user. The *ProvidentHider* algorithm is proposed to guarantee anonymity in the historical case under the relaxed assumptions that the adversary knows this association only when users are located in certain areas (e.g., workplaces). The correctness of the algorithm is formally proved and its applicability is experimentally evaluated.

**Centralized Defenses against Multiple-Issuer Attacks.** Preliminary results on the privacy leaks determined by multiple-issuer attacks are reported in [23]. Defenses for this kind of attacks are based on accurately generalizing location (as a quasi-identifier) in order to obtain QI-groups of requests with a certain degree of *diversity* in private values. A defense against multiple-issuer attacks both in the snapshot and in a limited version of the historical case is proposed by Riboni et Al. [24] using a combination of identity anonymity and private information obfuscation techniques. Further research is needed along this

line. For example, to understand under which conditions close values in private information can really be considered different (e.g., location areas).

### 3.2 Defenses Based on Private Information Obfuscation

As mentioned at the beginning of this section, these defenses aim at obfuscating private information released by users' requests as opposed to generalizing quasi-identifiers. To the best of our knowledge, all of the techniques in this category consider *location* as the private information to be protected, and implicitly or explicitly assume that user identity is known to the adversary or could be discovered. In the following of this chapter, we use *location obfuscation* to denote the general category of defenses aimed at obfuscating the exact location as private information of the (possibly identified) issuer.

Differently from the anonymity based defenses considering location as quasi-identifier, in this case it is less important to know the location of other users in order to provide privacy protection. For this reason, most of the location obfuscation techniques do not require a common location-aware trusted entity and, according to our categorization, they have a decentralized architecture. Sometimes these defenses are also claimed to provide a form of $k$-anonymity, leading to confusion with anonymity based defenses. The underlying idea is that due to the obfuscation, the location of the issuer (who is possibly not anonymous at all) cannot be distinguished among $k$ possible locations. In order to avoid confusion this property should be called *location anonymity*.

The idea of protecting location privacy by obfuscating location information was first proposed by Gruteser et al. [25]. The technique is aimed at avoiding the association of a user with a *sensitive area* she is crossing or approaching. The proposed defense is based on appropriately suspending user requests, ensuring that the location of the user may be confused among at least other $k$ areas. The proposed technique require a centralized entity, but it should not be difficult to modify the proposed algorithm so that it could be run directly on the users' mobile device. This defense algorithm is only validated via experiments. It is also not clear which privacy guarantees are provided if the adversary knows the algorithm.

Duckham et al. propose a protocol that allows a user to obtain the result of 1-NN (Nearest Neighbor) queries among a set of points of interest without disclosing her exact location [26]. The protocol is iterative. At the first iteration the user sends her obfuscated location to the SP that replies with the pair $\langle q, C \rangle$ where $q$ is the point of interest having the highest confidence $C$ of being the closest to the user. At each following iteration, the user can decide whether to provide additional location information in order to obtain a result with higher confidence. It is not specified how the generalization of the user's location is computed.

A different approach, proposed by Kido et al. [27], consists in sending, together with the real request, a set of fake requests. Since the adversary cannot distinguish the real request from the fake ones, it cannot discover the real location of the issuer, among the locations of the fake requests. This decentralized

solution is effective also in the case in which the adversary knows the defense function. However, this solution has the problem that, in order to effectively protect the location information, a high number of fake requests should be sent hence impacting on the communication costs. The technique is validated through experimental results only.

In [28], Ardagna et al. propose to use a combination of location obfuscation techniques and a metric to measure the obfuscation achieved. The difference with respect to other approaches is that the resulting obfuscation area may not contain the actual location of the issuer; moreover, the location measurement error introduced by sensing technologies is taken into account. It is not formally proved that the proposed defense protects against def-aware attacks. According to our categorization, the paper considers a centralized architecture, even if the proposed obfuscation techniques can be probably run on the client side.

Recently, Yiu et al. [29] proposed a different solution to obfuscate location information, specific for LBS requests that require $K$-NN queries. The idea of the algorithm, named SpaceTwist, is to issue each request as if it would originate from a location different from the real user location. The request may be repeated (from the same fake location) incrementally retrieving more nearest neighbor resources, until a satisfactory answer for the real location is obtained. This solution is particularly interesting since it does not require the existence of the centralized entity that provide privacy protection and involves no range NN queries on the server side. In the paper it is also formally shown how the adversary can compute the area where the user is possibly located under the assumptions that the adversary only knows the fake location, the number of requested resources, the replies from the server and the termination condition of the algorithm.

Referring to our categorization of attacks, the existing location obfuscation defenses focus on snapshot and single-issuer attacks. Example 2 shows that, in some cases, a historical attack can further restrict the possible locations of a user.

*Example 2.* A request issued by Alice is obfuscated in such a way that an adversary only knows that Alice is located in an area $A_1$ at time $t_1$. After a short time, Alice issues a second request that is obfuscated in such a way that the adversary knows that Alice is located somewhere in area $A_2$ at time $t_2$. Now, assume that there is a subregion $A'$ of $A_2$ such that, due to speed constraints, no matter where Alice were located in $A_1$ at time $t_1$, she has no way to get to $A'$ at time $t_2$. Now the adversary knows that at time $t_2$, Alice cannot be located in $A'$ and hence she must be in $A_2 \setminus A'$.

**Encryption Based Defenses.** We call encryption based, the defense proposals based on private information retrieval (PIR) techniques. The general objective of a PIR protocol is to allow a user to issue a query to a database without the database learning the query. In [30] this techniques is used to protect users' privacy in the LBS that computes 1-NN queries. The proposed solution is proved to solve the privacy problem under the most conservative assumptions about the adversary model as it does not reveal any information about the requests

to the adversary. Nevertheless, some concerns arises about the applicability of the proposed technique. First, the proposed solution applies to 1-NN queries only and it is not clear how it could be extended to other kinds of queries like $K$-NN queries or range queries. Second, this technique has high computational and communication overhead. Indeed, the experimental results shown in the paper give evidence that, also using a small database of objects to be retrieved, the computation time on the server side is in the order of seconds, while the communication cost is in the order of megabytes. In particular, the amount of data that needs to be exchanged between the server and the client is larger than the size of the database itself. It is not clear for which kind of services this overhead could be tolerable.

## 4 Historical k-Anonymity

Most of the defenses presented in Section 3 deal with snapshot attacks, while less attention has been given to historical attacks, namely those attacks that take advantage of the acquisition of a history of requests that can be recognized as issued by the same (anonymous) user. We believe that the conditions enabling this kind of attacks are very likely to occur in LBS. In this section, we present a general algorithm for providing historical anonymity as a defense against historical attacks. Consistently with the categorization of attacks and defenses presented in Sections 2 and 3 we formally characterize the attack we are dealing with, and the proposed defense. We then present the algorithm and provide its analysis.

In the following, the format of a LBS request is represented by the triple: $\langle IdData, STData, SSData \rangle$. **IdData** may be empty, contain the identity of the issuer, or a pseudo-identifier. **STData** contains spatiotemporal information about the location of the user performing the request, and the time the request was issued. This information may be a point in 3-dimensional space (with time being the third dimension) or an uncertainty region in the same space. $STData$ is partitioned into **SData** and **TData** that contain the spatial and temporal information about the user, respectively. **SSData** contains (possibly generalized) parameters characterizing the required service and service provider. An original request is denoted with $r$, while the same request transformed by a defense technique is denoted with $r'$.

### 4.1 Attack Category

Before we categorize attack and defense we are interested in, we use Example 3 to show that defense techniques for the snapshot cases cannot straightforwardly be used in the historical case. This example also provides our motivation for the attack and defense categories.

*Example 3.* Suppose Alice requires 3-anonymity and issues a request $r$. An algorithm safe against def-aware attacks is used to generalize $r$ into a request $r'$ whose spatiotemporal region includes only Alice, Bob, and Carl. Afterwards,

Alice issues a new request $r_1$ that is generalized into a request $r'_1$ whose spatiotemporal region includes only Alice, Ann, and John. Suppose the adversary is able to link requests $r'$ and $r'_1$, i.e., he is able to understand that the two requests have been issued by the same user. The adversary can observe that neither Bob nor Carl can be the issuer of $r'_1$, because they are not in the spatiotemporal region of $r'_1$; Consequently, they cannot be the issuers of $r'$ either. Analogously, considering the spatiotemporal region in $r'$, he can derive that Ann and John cannot be the issuers of the two request. Therefore, the adversary can identify Alice as the issuer of $r'$ and $r'_1$.

In this example, in addition to adversary's ability of using location as quasi-identifier, the ability to link requests is crucial for the attack to be successful. In a general scenario, in terms of the privacy attack dimensions identified in Section 2, we deal with attacks that:

1. Exploit location and time as *quasi-identifiers* in requests, that is, the adversary can identify users by their location information;
2. Use historical request traces, that is, the adversary can link requests that have been issued by the same user;
3. Do not correlate requests or sequences of requests issued by different users. This is equivalent to consider single-issuer attacks only.
4. Exploit knowledge of the defense, that is, we assume that the adversary knows the defense algorithm.

We will formalize items 1. and 2. below in order to analyze our defense rigorously, and the remaining items are exactly as discussed in the snapshot attack cases.

**Location as Quasi-Identifier.** Item 1. can be formalized as follows. For users' locations, we assume that the adversary has the knowledge expressed as the following $Ident$ function:

$$Ident_t : the\ Areas \longrightarrow the\ User\ sets,$$

that is, given an area $A$, $Ident_t(A)$ is the set of users whom, through certain means, the adversary has identified to be located in area $A$ at time $t$. In the following, when no confusion arises, we omit the time instant $t$. We further assume that this knowledge is *correct* in the sense that these identified users in reality are indeed in area $A$ at the time.

For a given user $i$, if there exists an area $A$ such that $i \in Ident(A)$, then we say $i$ is *identified* by the adversary. Furthermore, we say that $i$ is *identified in* $A$. Note that there may be users who are also in $A$ but the adversary does not identify them. This may happen either because the adversary is not aware of the presence of users in $A$, or because the adversary cannot identify these users even if he is aware of their presence. We do not distinguish these two cases as we shall see later that the distinction of the two cases does not make any perceptible difference in the ability of the adversary when the total population is large.

Clearly, in reality, there are lots of different sources of external information that can lead the adversary to estimate the location of users. Some may lead the adversary to know that a user is in a certain area, but not the exact location. For example, an adversary may know that Bob is in a pub (due to his use of a fidelity card at the pub), but may not know which room he is in. Some statistical analysis may be done to derive the *probability* that Bob is in a particular room, but this is beyond the scope of this chapter.

The most conservative assumption regarding this capability of the adversary is that $Ident(A)$ will give *exactly* all the users for each area $A$. It can be seen that if the privacy of the user is guaranteed in this most conservative assumption, then privacy is also guaranteed against any less precise $Ident$ function. However, this conservative assumption is unlikely true in reality, while some observed that this assumption degenerates the quality of service unnecessarily. It will be interesting to see how much privacy and quality of service change with more realistic $Ident$ functions.

Another function we assume to be known to the adversary is the following:

$$Num_t : the\ Areas \longrightarrow [0, \infty),$$

that is, given an area $A$, $Num_t(A)$ gives an estimate of the number of users in the area at time $t$. This is useful to the adversary to derive some statistical information when $Ident$ function does not recognize all the users in an area. This function can be obtained from statistical information publicly available or through some kind of counting mechanism such as tickets to a theater. Again, when no confusion arises, we do not indicate the time instant $t$.

**Request Traces Recognized by the Adversary.** In item (2) of the attack category, we assume that the adversary has the ability to link requests of the same user. This is formalized as the following function $L$:

$$L : the\ Requests \longrightarrow the\ Request\ sets,$$

that is, given a (generalized) request $r'$, $L(r')$ gives a set of requests such that the adversary has concluded, through certain means, are issued by the same user who issued the request $r'$. In other words, all the requests in $L(r')$ are *linked* to $r'$, although the adversary may still not know who the user is.

### 4.2 Defense Category

We now turn to discuss the category for our proposed defense strategy. The attacks being targeted by our defense are historical attacks more precisely described in Section 4.1. Moreover, based on the categorization of Section 3, our defense technique has the following characteristics:

1. *Defense technique:* we are using anonymity, or more specifically *historical k-anonymity*
2. *Defense architecture:* centralized; we are using LTS as our centralized defense server.

3. *Defense Validation:* we validate the effectiveness and efficiency via experiments.

As indicated in item (1) above, we use a notion of historical anonymity [12] to provide the basis for defense. To define the notion of historical anonymity, it is reasonable to assume that the LTS not only stores in its database the set of requests issued by each user, but also stores for each user the sequence of her location updates. This sequence is called *Personal History of Locations* (PHL). More formally, the PHL of user $u$ is a sequence of 3D points $(\langle x_1, y_1, t_1\rangle, \ldots, \langle x_m, y_m, t_m\rangle)$, where $\langle x_i, y_i\rangle$, for $i = 1, \ldots, m$, represents the position of $u$ (in two-dimensional space) at the time instant $t_i$.

A PHL $(\langle x_1, y_1, t_1\rangle, \ldots, \langle x_m, y_m, t_m\rangle)$ is defined to be *LT-consistent* with a set of requests $r_1, \ldots, r_n$ issued to a SP if for each request $r_i$ there exists an element $\langle x_j, y_j, t_j\rangle$ in the PHL such that the area of $r_i$ contains the location identified by the point $x_j, y_j$ and the time interval of $r_i$ contains the instant $t_j$.

Then, given the set $\bar{R}$ of all requests issued to a certain SP, a subset of requests $\bar{R}' = \{r_1, \ldots, r_m\}$ issued by the same user $u$ is said to satisfy *Historical k-Anonymity* if there exist $k-1$ PHLs $P_1, \ldots, P_{k-1}$ for $k-1$ users different from $u$, such that each $P_j$, $j = 1, \ldots, k-1$, is LT-consistent with $R'$.

The open problem in this case is how to generalize each request in order to obtain traces that are historical $k$-anonymous. One problem is that the LTS has to generalize each request when it is issued, without having the knowledge of the future users' locations nor the future requests that are to be issued. A separate problem is to avoid long traces; indeed, the longer is a trace, the more each request needs to be generalized in order to guarantee historical $k$-anonymity.

### 4.3 The *Greedy* Algorithm for Historical k-Anonymity

We now present a generalization algorithms for historical anonymity. In the next subsection we will analyze the anonymity achieved by a set of generalized requests. In the experimental section, we will present an evaluation of the effectiveness of the algorithm.

Our algorithm uses a snapshot anonymization algorithm, like *Grid*, as presented in Section 3. We modify this algorithm by adding the requirement that the perimeter of the MBR be always smaller than a user-given $maxP$ value. To achieve this, we basically recursively shrink the obtained MBR from the snapshot algorithm until its perimeter is smaller than $maxP$.

The idea of the *Greedy* algorithm was first proposed in [12] and a similar algorithm was also described in [21]. *Greedy* is aimed at preserving privacy under the attack given in Section 4.1. This algorithm computes the generalization of the first request $r$ in a trace using an algorithm for the snapshot case. (In our implementation, we use *Grid* as the snapshot algorithm to compute the generalization of the first request.) When this first request is generalized, the set $A$ of users located in the generalized location for the first request is stored. The generalized locations of each subsequent request $r'$ that is linked with $r$ is then taken as the MBR of the location of the users in $A$ at the time of $r'$. As in the

**Algorithm 1.** *Greedy*

**Input**: a request $r$, an anonymity set $A$, anonymity level $k$, and a maximum perimeter $maxP$.
**Output**: a generalized request $r'$ and an anonymity set $A'$.
**Method**:

```
find the MBR of all the current locations (at the time of request r) of users in A
(note that if A = ∅ then the MBR is empty).
if (the perimeter of the MBR is smaller than maxP) then
   if (|A| > 1) then
      replace the spatial information in r with the MBR, obtaining r'
      let A' = A
   else
      call Grid algorithm* with r, k, and maxP, obtaining r'
      let A' be the set of users currently in the spatial region of r'
   end if
else
   recursively shrink the MBR until its perimeter is smaller than maxP
   replace the spatial region in r with the resulting MBR, obtaining r'
   let A' be the set of users currently located in the resulting MBR
   if (|A'| ≤ 1) then
      call Grid algorithm with r, k, and maxP, obtaining r'
      let A' be the set of users currently in the spatial region of r'
   end if
end if
return r' and A'
```

* *Instead of* Grid*, other snapshot algorithms can be used here.*

modification of the *Grid* algorithm, when the MBR is smaller than $maxP$, we will recursively shrink it and exclude the users that fall out of the region. Algorithm 1 gives the pseudocode. This algorithm is called initially with the first request $r$ and empty set $A = \emptyset$, and subsequently, it is called with the successive request and the $A'$ returned from the previous execution.

### 4.4 Analysis of Anonymity

A successive use of Algorithm 1 returns a sequence of generalized requests for the user, and these generalized requests are forwarded to the SP. The question we have now is how much privacy protection such a sequence of generalized requests provides. That is, we want to find the following function:

$$Att : the\ Request\ set \times the\ Users \longrightarrow [0, 1],$$

Intuitively, given a (generalized) request $r'$ and a user $i$, $Att(r', i)$ gives the probability that the adversary can derive, under the assumption of the attack category of Section 4.1, that $i$ is the issuer of $r'$ among all the users.

In the following of this section we show how to specify the attack function. Once the attack function is specified, we can use the following formula to evaluate the privacy value of a request:

$$Privacy(r') = 1 - Att(r', issuer(r')) \quad (1)$$

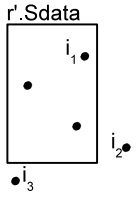
Intuitively, this value is the probability that the adversary will not associate the issuer of request $r'$ to $r'$.

In order to specify the $Att$ function, we introduce the function $Inside(i, r')$ that indicates the probability of user $i$ to be located in $r'.Sdata$ at the time of the request. Intuitively, $Inside(i, r') = 1$ if user $i$ is identified by the adversary as one of the users that are located in $r'.Sdata$ at time $r'.Tdata$, i.e., $i \in Ident_t(r'.Sdata)$ when $t = r'.Tdata$. On the contrary, $Inside(i, r') = 0$ if $i$ is recognized by the adversary as one of the users located outside $r'.Sdata$ at time $r'.Tdata$, i.e., there exists an area $A$ with $A \cap r'.Sdata = \emptyset$ such that $i \in Ident(A)$. Finally, if neither of the above cases hold, then the adversary does not know where $i$ is. There is still a probability that $i$ is in $r'.Sdata$. This is a much more involved case, and we first analyze the simple case, in which the adversary cannot link $r'$ to any other requests, i.e., there is no historical information about the issuer of $r'$. In this case, theoretically, this probability is the number of users in $r'.Sdata$ that are not recognized by the adversary (i.e., $Num(r'.Sdata) - |Ident(r'.Sdata)|$) divided by all the users who are not recognized by the adversary anywhere (i.e., $|I| - |Ident(\Omega)|$, where $I$ is the set of all users, and $\Omega$ is the entire area for the application). Formally,

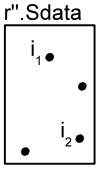
$$Inside(i, r') = \begin{cases} 1 & \text{if } i \in Ident(r'.Sdata) \\ 0 & \text{if } \exists A : A \cap r'.Sdata = \emptyset \text{ and } i \in Ident(A) \\ \frac{Num(r'.Sdata) - |Ident(r'.Sdata)|}{|I| - |Ident(\Omega)|} & \text{otherwise} \end{cases} \quad (2)$$

*Example 4.* Consider the situation shown in Figure 2(a) in which there is the request $r'$ such that, at time $r'.Tdata$, there are three users in $r'.Sdata$: one of them is identified as $i_1$, the other two are not identified. The adversary can also identify users $i_2$ and $i_3$ outside $r'.Sdata$ at time $r'.Tdata$. Assume that the set $I$ contains 100 users.

Clearly, $i_2$ and $i_3$ have zero probability of being the issuers, since they are identified outside $r'.Sdata$ and due to the assumption that the spatial region of any generalized request must contain the spatial region of the original request. That is, $Inside(i_2, r') = Inside(i_3, r') = 0$. On the contrary, the adversary is




(a) First request, $r'$.




(b) Second request, $r''$.

**Fig. 2.** Example of attack

sure about the fact that $i_1$ is located in $r'.Sdata$, i.e., $Inside(i_1, r') = 1$. By Formula 2, for each user $i$ in $I \setminus \{i_1, i_2, i_3\}$, $Inside(i, r') = 2/97$.

However, when the adversary is assumed to link $r'$ to other requests, then we need to be more careful. We define $Inside(i, L(r'))$ to be the probability that $i$ is located in $r.STdata$ for each request $r$ in $L(r')$. To calculate $Inside(i, L(r'))$, we need to know the probability of a user $i$ in area $B$ at time $t$ if we know that the same user was in a series areas $A_1, \ldots, A_p$ at time $t_1, \ldots, t_p$, respectively, i.e., we need estimate the conditional probability:

$$P(Inside_t(i, B)|Inside_{t_1}(i, A_1), \ldots, Inside_{t_p}(i, A_p)).$$

This conditional probability depends on many factors, including the distance between these areas and the assumed moving speed of the user. We may use historical data to study this conditional probability. Absent of the knowledge of user's moving speed or historical data, in this contribution, we use a simplifying independence assumption that the probability of a user in $A$ is independent of where the user has been in the past. Hence, we assume

$$Inside(i, L(r')) = \Pi_{r \in L(r')} Inside(i, r),$$

where $Inside(i, r)$ is as given in Formula 2.

*Example 5.* Continue from Example 4 and assume a second request $r''$ (see Figure 2(b)) is issued after $r'$ and that $r''$ is linked with $r'$, so $L(r'') = \{r', r''\}$. We call $L(r'')$ a trace and denote it $\tau$. At time $r''.Tdata$, there are 4 users inside $r''.Sdata$, two of which are identified as $i_1$ and $i_2$. No user is identified outside $r''.Sdata$. From the above discussion, it follows that $Inside(i_2, \tau) = Inside(i_3, \tau) = 0$ since $i_2$ and $i_3$ are identified outside the first generalized request $r'$. All the other users have a non-zero probability of being inside the generalized location of each request in the trace. In particular, $Inside(i_1, \tau) = 1$ since $i_1$ is recognized in both requests. Consider a user $i \in I \setminus \{i_1, i_2, i_3\}$. Since $Inside(i, r') = 2/97$ and $Inside(i, r'') = 2/98$, we have $Inside(i, \tau) = 0.00042$, a very small number.

Now we can obtain the attack formula:

$$Att(r', i) = \frac{Inside(i, L(r'))}{\sum_{i' \in I} Inside(i', L(r'))} \tag{3}$$

*Example 6.* Continue from Example 5. We now know $Att(r'', i_1) = 1/(1 + 97 * 0.00042) \approx 96\%$, $Att(r'', i_2) = Att(r'', i_3) = 0$, and $Att(r'', i) = 0.00042/(1 + 97 * 0.00042) \approx 0$ for each user $i$ in $I \setminus \{i_1, i_2, i_3\}$.

From this example, we can observe that the independence assumption causes an overestimate of the probability of $i_1$ to be the issuer, but an underestimate of the probability of users other than $i_1$, $i_2$, and $i_3$. If we knew that a user in $r'.Sdata$ is very likely to be in $r''.Sdata$ (at the respective times), then the estimate of the attack values in Example 6 needs to be revised.

## 5 Impact of Realistic Simulations on the Evaluation of Anonymity-Based Defense Techniques

As we motivated in the previous sections, the correctness of an anonymity-preserving technique can be formally proved based on the specific assumptions made on the adversary model. However, in practice, different adversaries may have different background knowledge and inferencing abilities. Hence, one approach consists in stating conservative assumptions under which anonymity can be guaranteed against a broad range of potential adversaries. The drawback of this approach is clear from the conservative assumptions about location knowledge considered so far by anonymity based solutions: in order to protect from the occasional knowledge by the adversary about people present at a given location (unknown to the defender), it is (often implicitly) assumed the same knowledge for all locations. Such assumptions are not realistic and lead to overprotect the users' anonymity, hence negatively impacting on the quality of service. A different approach, taken by several researcher is experimental evaluation. Since large set of real, accurate data are very hard to obtain, in most cases experiments are based on synthetic data generated through simulators. In this section we focus on validating anonymity-based defense techniques, and we show that in order to obtain significant results, simulations must be very carefully designed. In addition to evaluating the *Greedy* algorithm as a representative of historical anonymity based defenses, we are interested in the following more general questions: a) *how much does the adversary model affect the privacy obtained by the defense according to the evaluation?*, and b) *how much does the specific service deployment model affect the results of the evaluation?*

### 5.1 The *MilanoByNight* Simulation

In order to carefully design the simulation, we concentrate on a specific class of LBS called *friend-finder*. A friend-finder reveals to a participating user the presence of other close-by participants belonging to a particular group (friends is only one example), possibly showing their position on a map. In particular, we consider the following service: a user issues a request specifying a threshold distance $\delta_A$ and the group of target participants (e.g., the users sharing a certain interest). The SP replies with the set of participants belonging to that group whose location is not farther than $\delta_A$ from the issuer.

A first privacy threat for a user of the friend-finder service is the association of that user's identity with the service parameters and, in particular, with the group of target participants, since this can reveal the user's interests or other private information. Even if the user's identity is not explicit in a request, an adversary can obtain this association, by using the location information of a request as a quasi-identifier.

A second privacy threat is the association of the user's identity with the locations visited by that user[2]. We recall that this association takes place

[2] A obfuscation-based defense against this threat, specifically designed for the friend-finder service, has recently been proposed [31].

independently from the service requests if the adversary's background location knowledge is "complete" (see Section 2). However, consider the case in which the background knowledge is "partial" i.e., it contains the association between user identity and location information only for some users in some locations at some time instants. Example 7 shows how, in this case, an adversary can exploit a set of friend-finder requests to derive location information that are not included in the background knowledge.

*Example 7.* User $A$ issues a friend-finder request $r_1$. An adversary obtains $r_1$ and discovers that $A$ is the issuer by joining the location information in the request with his background knowledge (i.e., the location information of $r_1$ is used as quasi-identifier). Then, $A$ moves to a different location and issues a request $r_2$. The adversary obtains $r_2$, but in this case his background knowledge does not contain sufficient information to identify the issuer of the request. However, if the adversary can understand that $r_1$ and $r_2$ are linked (i.e., issued from the same issuer), then he derives that $A$ is also the issuer of $r_2$ and hence obtains new location information about $A$.

We suppose that the friend-finder service is primarily used by people during entertainment hours, especially at night. Therefore, the ideal dataset for our experiments should represent movements of people on a typical Friday or Saturday night in a big city, when users tend to move to entertainment places. To our knowledge, currently there are no datasets like this publicly available, specially considering that we want to have large scale, individual, and precise location data (i.e., with the same approximation of current consumer GPS technology).

**Relevant Simulation Parameters.** For our experiments we want to artificially generate movements for $100,000$ users on the road network of Milan[3]. The total area of the map is 324 km$^2$, and the resulting average density is 308 users/km$^2$. The simulation includes a total of $30,000$ home buildings and $1,000$ entertainment places; the first value is strictly related to the considered number of users, while the second is based on real data from public sources which also provide the geographical distribution of the places. Our simulation starts at 7 pm and ends at 1 am. During these hours, each user moves from house to an entertainment place, spends some time in that place, and possibly moves to another entertainment place or goes back home.

All probabilities related to users' choices are modeled with probability distributions. In order to have a realistic model of these distributions, we prepared a survey to collect real users data. We are still collecting data, but the current parameters are based on interviews of more than 300 people in our target category.

**Weaknesses of Mostly Random Movement Simulations.** Many papers in the field of privacy preservation in LBS use artificial data generated by moving object simulators to evaluate their techniques. However, most of the simulators

[3] $100,000$ is an estimation of the number of people participating in the service we consider.

are usually not able to reproduce a realistic behavior of users. For example, objects generated by the Brinkhoff generator [32] cannot be aggregated in certain places (e.g., entertainment places). Indeed, once an object is instantiated, the generator chooses a random destination point on the map; after reaching the destination, the object disappears from the dataset. For the same reason, it is not possible to reproduce simple movement patterns (e.g.: a user going out from her home to another place and then coming back home), nor to simulate that a user remains for a certain time in a place.

Despite these strong limitations, we made our best effort to use the Brinkhoff simulator to generate a set of user movements with characteristics as close as possible to those described above. For example, in order to simulate entertainment places, some random points on the map, among those points on the trajectories of users, were picked. The simulation has the main purpose of understanding if testing privacy preservation over random movement simulations gives significantly different results with respect to more realistic simulations.

**Generation of User Movements with a Context Simulator.** In order to obtain a dataset consistent with the parameters specified above, we need a more sophisticated simulator. For our experiments, we have chosen to customize the Siafu context simulator [33]. With a context simulator it is possible to design models for agents, places and context. Therefore, it is possible to define particular places of aggregation and make users dynamically choose which place to reach and how long to stay in that place.

The most relevant parameters characterizing the agents' behavior are derived from our survey. For example, one parameter that characterizes the behavior of the agents is the average time spent in an entertainment place; This value was collected in our survey and resulted to have the following values: 9.17% of the users stays less than 1 hour, 34.20% stays between 1 and 2 hours, 32.92% stays between 2 and 3 hours, 16.04% stays between 3 and 4 hours, and 7.68% stays more than 4 hours. Details on the simulation can be found in [34].

### 5.2 Experimental Settings

In our experiments we used two datasets of users movements. The dataset *AB* (Agent-Based) was generated with the customized Siafu simulator, while the dataset *MRM* (Mostly Random Movement) was created with the Brinkhoff simulator. In both cases, we simulate LBS requests for the friend-finder service by choosing random users in the simulation, we compute for each request the generalization according to a given algorithm, and finally we evaluate the anonymity of the resulting request as well as the Quality of Service (QoS).

Different metrics can be defined to measure QoS for different kind of services. For instance, for the friend-finder service we are considering, it would be possible to measure how many times the generalization leads the SP to return an incorrect result i.e., the issuer is not notified of a close-by friend or, vice versa, the issuer is notified for a friend that is not close-by. While this metric is useful for this specific application, we want to measure the QoS independently from the specific

kind of service. For this reason, in this chapter we evaluate how QoS degrades in terms of the perimeter of the generalized location.

In addition to the dataset of user movements, we identified other two parameters characterizing the deployment model that significantly affect the experimental results: the *number of users* in the system, which remains almost constant at each time instant and the user-required degree of indistinguishability $k$. These two parameters, together with the most important others, are reported in Table 1, with the values in bold denoting default values.

We also identified three relevant parameters that characterize the adversary model. The parameter $P_{id-in}$ indicates the probability that the adversary can identify a user when she is located in a entertainment place while $P_{id-out}$ is the probability that the adversary identifies a user in any other location (e.g., while moving from home to a entertainment place). While we also perform experiments where the two probabilities are the same, our scenario suggests as much more realistic a higher value for $P_{id-in}$ (it is considered ten times higher than $P_{id-out}$). This is due to the fact that restaurants, pubs, movie theaters, and similar places are likely to have different ways to identify people (fidelity or membership cards, WiFi hotspots, cameras, credit card payments, etc.) and in several cases more than one place is owned by the same company that may have an interest in collecting data about its customers. Finally, $P_{link}$ indicates the probability that two consecutive requests can be identified as issued by the same user.[4] While we perform our tests considering a full range of values, the specific default value reported in the table is due to a recent study on the ability of linking positions based on spatiotemporal correlation [35].

**Table 1.** Parameter values

| **Parameter** | **Values** |
|---|---|
| dataset | ***AB***, *MRM* |
| number of users | 10k, 20k, 30k, 40k, 50k, 60k, 70k, 80k, 90k, **100k** |
| $k$ | **10**, 20, 30, 40, 50, 60 |
| $P_{id-in}$ | 0.1, **0.2**, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0 |
| $P_{id-out}$ | 0.01, **0.02**, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1 |
| $P_{link}$ | 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, **0.87**, 0.9, 1.0 |

The experimental results we show in this section are obtained by running the simulation for 100 issuers and then computing the average values.

In our experiments we evaluated two generalization algorithms. One algorithm is *Greedy* which is described in Section 4 and is a representative of the historical generalization algorithm proposed so far [12,20,21]. The other algorithm is *Grid* which is briefly described in Section 3.1 is a representative of the snapshot generalization algorithms. In [14] *Grid* is shown to have better performance (in terms of the quality of service) when compared to other snapshot generalization

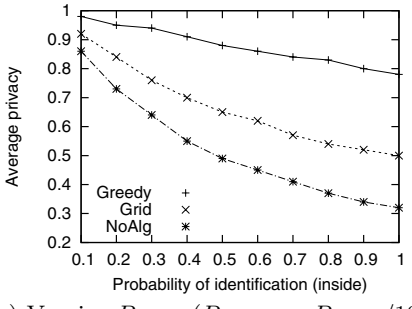
[4] The limitation to consecutive requests is because in our specific scenario we assume linking is performed mainly through spatiotemporal correlation.

algorithms like, for example, *Hilbert Cloak*. We also evaluated the privacy threat when no privacy preserving algorithm is applied. The label *NoAlg* is used in the figures to identify results in this particular case.

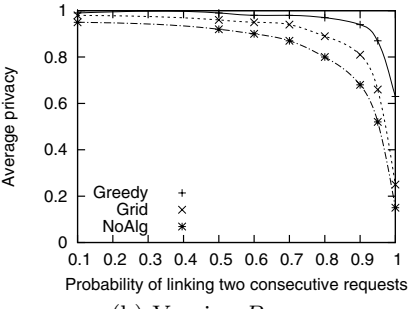
### 5.3 Impact of the Adversary Model on the Evaluation of the Generalization Algorithms

We now present a set of experiments aimed at evaluating the impact of the adversary model on the anonymity provided by the generalization algorithms.

Two main parameters characterizing the adversary model are $P_{id-in}$ and $P_{link}$. In Figure 3(a) we show the average anonymity for different values of $P_{id-in}$ when, in each test, $P_{id-out}$ is set to $P_{id-in}/10$. As expected, considering a trace of requests, the higher is the probability of identifying users in one or more of the regions from which the requests in the trace were performed, the smaller is the level of anonymity.




(a) Varying $P_{id-in}$ ($P_{id-out} = P_{id-in}/10$).




(b) Varying $P_{link}$.

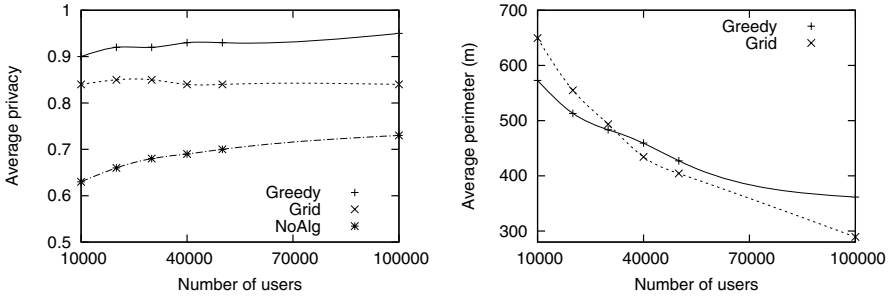
**Fig. 3.** Average anonymity

Figure 3(b) shows the impact of $P_{link}$ on the average privacy. As expected, high values of $P_{link}$ lead to small values of privacy. Our results show that the relation between the $P_{link}$ and privacy is not linear. Indeed, privacy depends almost linearly on the average length of the traces identified by the adversary. In turn, the average length of the traces grows almost exponentially with the value of $P_{link}$.

To summarize the first set of experiments, our findings show that the parameters that characterize the adversary model significantly affect the evaluation of the generalization algorithms. This implies that when a generalization algorithm is evaluated it is necessary to estimate realistic values for these parameters. Indeed, an error in the estimation may lead to misleading results.

### 5.4 Impact of the Deployment Model on the Evaluation of the Generalization Algorithms

We now show a set of experimental results designed to evaluate the impact of the deployment model on the evaluation of the generalization algorithms.




(a) Average privacy. (b) Average perimeter.

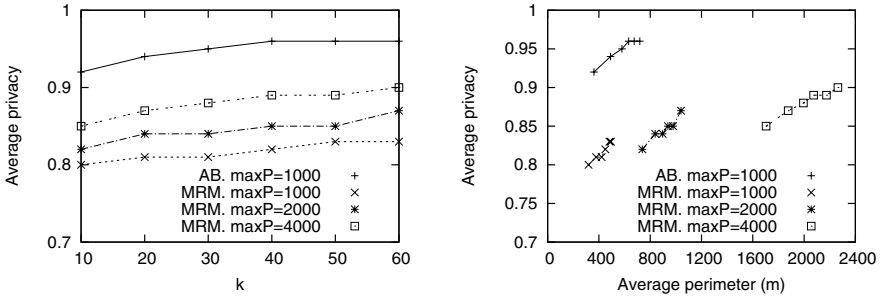
**Fig. 4.** Performance evaluation for different values of the total population

Figure 4(a) shows that the average privacy obtained with *Greedy* and *Grid* is not significantly affected by the size of the total population. Indeed, both algorithms, independently from the total number of users, try to have generalized locations that cover the location of $k$ users, so the privacy of the requests is not affected. However, when the density is high, the two algorithms can generalize to a small area, while when the density is low, a larger area is necessary to cover the location of $k$ users (see Figure 4(b)). On the contrary, the privacy obtained when no generalization is performed is significantly affected by the total population. Indeed, a higher density increases the probability of different users to be in the same location and hence it increases privacy also if the requests are not generalized.

The set of tests reported in in Figure 5 compares the privacy achieved by the *Greedy* algorithm on the two datasets for different values of $k$ and for different values of QoS. The experiments on $MRM$ were repeated trying also larger values for the QoS threshold ($maxP = 2000$ and $maxP = 4000$), so three different versions of $MRM$ appear in the figures. In order to focus on these parameters only, in these tests the probability of identification was set to the same value for any place ($P_{id-in} = P_{id-out} = 0.1$), and for the $MRM$ dataset the issuer of the requests was randomly chosen only among those that stay in the simulation for 3 hours, ignoring the ones staying for much shorter time that inevitably are part of this dataset. This setting allowed us to compare the results on the two datasets using the same average length of traces identified by the adversary.

Figure 5(a) shows that the average privacy of the algorithm evaluated on the $AB$ dataset is much higher than on the $MRM$ dataset. This is mainly motivated by the fact that in $AB$ users tend to concentrate in a few locations (the entertainment places) and this enhances privacy. This is also confirmed by a similar test performed without using any generalization of locations; we obtained values constantly higher for the $AB$ dataset (the average privacy is 0.67 in AB and 0.55 in $MRM$).

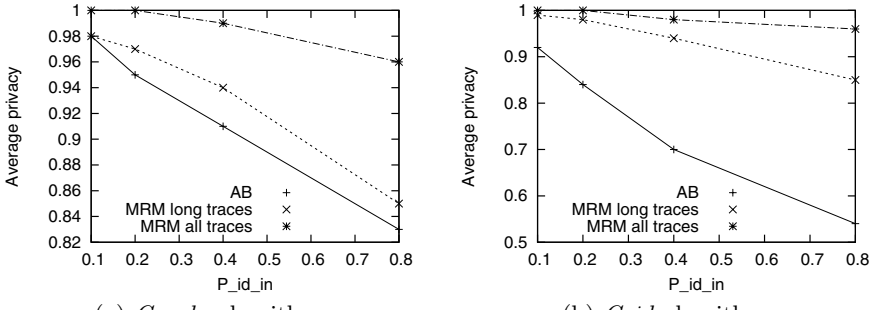
In Figure 5(b) we show the QoS achieved by the algorithm in the two datasets with respect to the average privacy achieved. This result confirms that the level of privacy evaluated on the $AB$ dataset using small values of $k$ and $maxP$ for




(a) Average privacy as a function of the level of indistinguishability $k$.

(b) Average privacy as a function of the average perimeter.

**Fig. 5.** Evaluation of the *Greedy* algorithm using $AB$ and $MRM$ data sets. $P_{id-in} = P_{id-out} = 0.1$.




(a) *Greedy* algorithm.

(b) *Grid* algorithm.

**Fig. 6.** Average privacy using $AB$ and $MRM$ data sets. $P_{id-out} = P_{id-in}/10$.

the algorithm cannot be observed on the $MRM$ dataset even with much higher values for these parameters.

From the experiments shown in Figure 5 we can conclude that if the $MRM$ dataset is used as a benchmark to estimate the values of $k$ and $maxP$ that are necessary to provide a desired average level of privacy, then the results will suggest the use of values that are over-protective. As a consequence, it is possible that the service will exhibit a much lower QoS than the one that could be achieved with the same algorithm.

The above results may still support the safety of using $MRM$, since according to what we have seen above a technique achieving a certain level of privacy may only do better in a real scenario. However, our second set of experiments shows that this is not the case.

In Figure 6 we show the results we obtained by varying the probability of identification. For this test, we considered two sets of issuers in the $MRM$ data set. One set is composed by users that stay in the simulation for 3 hours, (*MRM long traces*, in Figure 6), while the other contains issuers randomly chosen in the

entire set of users (*MRM all traces*, in Figure 6), hence including users staying in the simulation for a much shorter time.

In Figure 6(a) and 6(b) we can observe that the execution on the *MRM* dataset leads to evaluate a privacy level that is higher than the one obtained on the *AB* dataset. In particular, the evaluation of the *Grid* algorithm using the *MRM* dataset (Figure 6(b)), would suggest that the algorithm is able to provide a high privacy protection. However, when evaluating the same algorithm using the more realistic dataset *AB*, this conclusion seems to be incorrect. In this case, the evaluation on the *MRM* dataset may lead to underestimate the privacy risk, and hence to deploy services based on generalization algorithms that may not provide the minimum required level of privacy.

## 6 Open Problems

As seen from the previous sections, progress has been made in protecting users' privacy in using location based services. However, much research is still needed. In this section we discuss some open problems that are immediately related to the anonymity-based techniques we discussed in this contribution.

*Recognizing the dynamic role of quasi-identifiers and of private information.* All the techniques proposed so far in the literature assume that the informations in the request acting as quasi-identifier or as private information do not change among different requests. However, it should be observed that this may not always be the case. Indeed, in a realistic scenario, only some locations can act as quasi-identifiers, and, similarly, only some service requests contain private information (location and/or service parameters). The proper recognition of the role of information in the requests is crucial in designing an effective defense technique. Indeed, over conservative assumptions lead to quality of service degradation, and ignoring the role of data as quasi-identifier or private information in a request leads to privacy violation.

*Pattern-based quasi-identifiers.* In the historical case the adversary can observe some movement patterns. In this case, even if a single request contains no information acting as quasi-identifier, the sequence of movements can lead to the identification of the issuer. Consider the following example: a user issues several linkable requests from her home (location A) and workplace (location B). Assume that, since the requests are generalized to areas containing public places, the adversary cannot restrict the set of possible issuers by considering each single request. However, if the adversary is able to extract a movement pattern from the requests, he can infer that the issuer most probably lives in location A and works in location B, and this information is a quasi-identifier [12,36].

*Personalization of the degree of anonymity.* In our discussion we never considered issues related to the personalization of defense parameters, as for example, the degree of anonymity $k$ to be enforced. Some approaches (e.g. [9]) actually explicitly allow different users to specify different values of $k$. A natural question is if the other techniques can be applied and can be considered safe even in

this case. Once again, to answer this question it is essential to consider which knowledge an adversary may obtain. The degree of anonymity $k$ desired by each user at the time of a request is not assumed to be known by the adversary (even in the def-aware attacks) in the presented algorithms, hence the algorithms that are safe against the corresponding attacks remain safe even when personalized values for $k$ are considered.

However, it may be reasonable to consider attacks in which the adversary may obtain information about $k$. In the multiple-issuer case, the adversary may use, for example, data mining techniques to figure out the $k$ value. Example 8 shows that, in such a scenario, the presented algorithms need to be extended in order to provide an effective defense.

*Example 8.* Alice issues a request $r$ asking a degree of anonymity $k = 2$. Using a defense algorithm against def-aware attacks, $r$ is generalized to the request $r'$ that has a spatiotemporal region containing only Alice and Bob. Since the generalization algorithm is safe against def-aware attacks, if $r$ were issued by Bob with $k = 2$, then it would be generalized to $r'$. However, if the adversary knows that Bob always issues requests with $k \geq 3$, then he knows that if the issuer of $r$ were Bob, the request would have been generalized to a request $r''$ different from $r'$, because the spatiotemporal region of $r''$ should include at least 3 users. Hence the adversary would identify Alice as the issuer of $r'$.

*Deployment-aware data generator.* Earlier, we claimed that the experimental evaluation of LBS privacy preserving techniques should be based on user movement datasets obtained through simulations *tailored* to the specific deployment scenario of the target services. Our results support our thesis for the class of LBS known as friend-finder services, for defense techniques based on spatial cloaking, and for attack models that include the possibility for the adversary to occasionally recognize people in certain locations. These results can be extended to other types of LBS, other defense techniques, and various types of attacks. Thus, we believe a significant effort should be devoted to the development of new flexible and efficient context-aware user movement simulators, as well as to the collection of real data, possibly even in an aggregated form, to properly tune the simulations. In our opinion this is a necessary step to have significant common benchmarks to evaluate LBS privacy preserving techniques.

## 7 Conclusion

In this contribution, we introduced the privacy problem in LBS by categorizing both attacks and existing defense techniques. We then discussed the use of anonymity for protection, focusing on the notion of historical $k$-anonymity and on the techniques to ensure this form of anonymity. Finally, we provided a performance evaluation of these techniques depending on the adversary model and on the specific service deployment model. Based on our extensive work on the simulation environment, we believe that the design of realistic simulations for specific services, possibly driven by real data, is today one of the main challenges in this

field, since proposed defenses need serious evaluation, and theoretical validation is important but has several limits, mainly due to the conservative assumptions that seem very hard to avoid.

## Acknowledgments

This work was partially supported by National Science Foundation (NSF) under grant N. CNS-0716567, and by Italian MIUR under grants InterLink-II04C0EC1D and PRIN-2007F9437X.

## References

1. Barkhuus, L., Dey, A.: Location-based services for mobile telephony: a study of users privacy concerns. In: Proc. of the 9th International Conference on Human-Computer Interaction, pp. 709–712. IOS Press, Amsterdam (2003)
2. Ciriani, V., di Vimercati, S.D.C., Foresti, S., Samarati, P.: $k$-Anonymity. In: Secure Data Management in Decentralized Systems. Springer, Heidelberg (2007)
3. Bettini, C., Wang, X.S., Jajodia, S.: How anonymous is k-anonymous? look at your quasi-id. In: Jonker, W., Petković, M. (eds.) SDM 2008. LNCS, vol. 5159, pp. 1–15. Springer, Heidelberg (2008)
4. Dalenius, T.: Finding a needle in a haystack - or identifying anonymous census record. Journal of Official Statistics 2(3), 329–336 (1986)
5. Abul, O., Bonchi, F., Nanni, M.: Never walk alone: Uncertainty for anonymity in moving objects databases. In: Proc. of the 24th International Conference on Data Engineering, pp. 376–386. IEEE Computer Society, Los Alamitos (2008)
6. Bettini, C., Mascetti, S., Wang, X.S., Jajodia, S.: Anonymity in location-based services: towards a general framework. In: Proc. of the 8th International Conference on Mobile Data Management, pp. 69–76. IEEE Computer Society, Los Alamitos (2007)
7. Gruteser, M., Grunwald, D.: Anonymous usage of location-based services through spatial and temporal cloaking. In: Proc. of the 1st International Conference on Mobile Systems, Applications and Services, pp. 31–42. The USENIX Association (2003)
8. Kalnis, P., Ghinita, G., Mouratidis, K., Papadias, D.: Preventing location-based identity inference in anonymous spatial queries. IEEE Transactions on Knowledge and Data Engineering 19(12), 1719–1733 (2007)
9. Mokbel, M.F., Chow, C.Y., Aref, W.G.: The new casper: query processing for location services without compromising privacy. In: Proc. of the 32nd International Conference on Very Large Data Bases, VLDB Endowment, pp. 763–774 (2006)
10. Beresford, A.R., Stajano, F.: Mix zones: User privacy in location-aware services. In: Proc. of the 2nd Annual Conference on Pervasive Computing and Communications, pp. 127–131. IEEE Computer Society, Los Alamitos (2004)
11. Hoh, B., Gruteser, M.: Protecting location privacy through path confusion. In: Proc. of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks, pp. 194–205. IEEE Computer Society, Los Alamitos (2005)
12. Bettini, C., Wang, X.S., Jajodia, S.: Protecting privacy against location-based personal identification. In: Jonker, W., Petković, M. (eds.) SDM 2005. LNCS, vol. 3674, pp. 185–199. Springer, Heidelberg (2005)

13. Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M.: $l$-Diversity: Privacy Beyond $k$-Anonymity. In: Proceedings of the 22nd International Conference on Data Engineering, p. 24. IEEE Computer Society, Los Alamitos (2006)
14. Mascetti, S., Bettini, C., Freni, D., Wang, X.S.: Spatial generalization algorithms for LBS privacy preservation. Journal of Location Based Services 2(1), 179–207 (2008)
15. Gedik, B., Liu, L.: Protecting location privacy with personalized $k$-anonymity: Architecture and algorithms. IEEE Transactions on Mobile Computing 7(1), 1–18 (2008)
16. Chow, C.Y., Mokbel, M.F., Liu, X.: A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In: Proc. of the 14th International Symposium on Geographic Information Systems, pp. 171–178. ACM, New York (2006)
17. Ghinita, G., Kalnis, P., Skiadopoulos, S.: Prive: anonymous location-based queries in distributed mobile systems. In: Proc. of the 16th international conference on World Wide Web, pp. 371–380. ACM Press, New York (2007)
18. Ghinita, G., Kalnis, P., Skiadopoulos, S.: Mobihide: A mobile peer-to-peer system for anonymous location-based queries. In: Papadias, D., Zhang, D., Kollios, G. (eds.) SSTD 2007. LNCS, vol. 4605, pp. 221–238. Springer, Heidelberg (2007)
19. Hu, H., Xu, J.: Non-exposure location anonymity. In: Proc. of the 25th International Conference on Data Engineering, pp. 1120–1131. IEEE Computer Society, Los Alamitos (2009)
20. Chow, C.Y., Mokbel, M.: Enabling private continuous queries for revealed user locations. In: Papadias, D., Zhang, D., Kollios, G. (eds.) SSTD 2007. LNCS, vol. 4605, pp. 258–275. Springer, Heidelberg (2007)
21. Xu, T., Cai, Y.: Location anonymity in continuous location-based services. In: Proc. of ACM International Symposium on Advances in Geographic Information Systems, p. 39. ACM Press, New York (2007)
22. Mascetti, S., Bettini, C., Wang, X.S., Freni, D., Jajodia, S.: ProvidentHider: an algorithm to preserve historical $k$-anonymity in lbs. In: Proc. of the 10th International Conference on Mobile Data Management, pp. 172–181. IEEE Computer Society, Los Alamitos (2009)
23. Bettini, C., Jajodia, S., Pareschi, L.: Anonymity and diversity in LBS: a preliminary investigation. In: Proc. of the 5th International Conference on Pervasive Computing and Communications, pp. 577–580. IEEE Computer Society, Los Alamitos (2007)
24. Riboni, D., Pareschi, L., Bettini, C., Jajodia, S.: Preserving anonymity of recurrent location-based queries. In: Proc. of 16th International Symposium on Temporal Representation and Reasoning. IEEE Computer Society, Los Alamitos (2009)
25. Gruteser, M., Liu, X.: Protecting privacy in continuous location-tracking applications. IEEE Security & Privacy 2(2), 28–34 (2004)
26. Duckham, M., Kulik, L.: A formal model of obfuscation and negotiation for location privacy. In: Gellersen, H.-W., Want, R., Schmidt, A. (eds.) PERVASIVE 2005. LNCS, vol. 3468, pp. 152–170. Springer, Heidelberg (2005)
27. Kido, H., Yanagisawa, Y., Satoh, T.: Protection of location privacy using dummies for location-based services. In: Proc. of the 21st International Conference on Data Engineering Workshops, p. 1248. IEEE Computer Society, Los Alamitos (2005)
28. Ardagna, C.A., Cremonini, M., Damiani, E., di Vimercati, S.D.C., Samarati, P.: Location privacy protection through obfuscation-based techniques. In: Barker, S., Ahn, G.-J. (eds.) Data and Applications Security 2007. LNCS, vol. 4602, pp. 47–60. Springer, Heidelberg (2007)

29. Yiu, M.L., Jensen, C.S., Huang, X., Lu, H.: Spacetwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services. In: Proc. of the 24th International Conference on Data Engineering, pp. 366–375. IEEE Computer Society, Los Alamitos (2008)
30. Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C., Tan, K.L.: Private queries in location based services: Anonymizers are not necessary. In: Proc. of SIGMOD, pp. 121–132. ACM Press, New York (2008)
31. Mascetti, S., Bettini, C., Freni, D., Wang, X.S., Jajodia, S.: Privacy-aware proximity based services. In: Proc. of the 10th International Conference on Mobile Data Management, pp. 31–40. IEEE Computer Society, Los Alamitos (2009)
32. Brinkhoff, T.: A framework for generating network-based moving objects. GeoInformatica 6(2), 153–180 (2002)
33. Martin, M., Nurmi, P.: A generic large scale simulator for ubiquitous computing. In: Proc. of the 3rd Conference on Mobile and Ubiquitous Systems: Networks and Services. IEEE Computer Society, Los Alamitos (2006)
34. Mascetti, S., Freni, D., Bettini, C., Wang, X.S., Jajodia, S.: On the impact of user movement simulations in the evaluation of LBS privacy-preserving techniques. In: Proc. of the International Workshop on Privacy in Location-Based Applications, Malaga, Spain. CEUR-WS, vol. 397, pp. 61–80 (2008)
35. Vyahhi, N., Bakiras, S., Kalnis, P., Ghinita, G.: Tracking moving objects in anonymized trajectories. In: Bhowmick, S.S., Küng, J., Wagner, R. (eds.) DEXA 2008. LNCS, vol. 5181, pp. 158–171. Springer, Heidelberg (2008)
36. Golle, P., Partridge, K.: On the anonymity of home/work location pairs. In: Tokuda, H., Beigl, M., Friday, A., Bernheim Brush, A.J., Tobe, Y. (eds.) Pervasive 2009. LNCS, vol. 5538, pp. 390–397. Springer, Heidelberg (2009)