# Preserving location and absence privacy in geo-social networks

**Conference Paper** · January 2010

DOI: 10.1145/1871437.1871480 · Source: DBLP



**5 authors**, including:

Sergio Mascetti
University of Milan
**61** PUBLICATIONS **858** CITATIONS

SEE PROFILE

Claudio Bettini
University of Milan
**216** PUBLICATIONS **5,211** CITATIONS

SEE PROFILE

Christian Jensen
Aalborg University
**634** PUBLICATIONS **18,520** CITATIONS

SEE PROFILE

# Preserving Location and Absence Privacy in Geo-Social Networks

Dario Freni[1] Carmen Ruiz Vicente[2] Sergio Mascetti[1] Claudio Bettini[1] Christian S. Jensen[3]

[1]EveryWare Lab, DICo
Università degli Studi di Milano
Italy
{freni,mascetti,bettini}
@dico.unimi.it

[2]Dept. of Computer Science
Aalborg University
Denmark
carmrui@cs.aau.dk

[3]Dept. of Computer Science
Aarhus University
Denmark
csj@cs.au.dk

## ABSTRACT

Online social networks often involve very large numbers of users who share very large volumes of content. This content is increasingly being tagged with geo-spatial and temporal coordinates that may then be used in services. For example, a service may retrieve photos taken in a certain region. The resulting geo-aware social networks (GeoSNs) pose privacy threats beyond those found in location-based services. Content published in a GeoSN is often associated with references to multiple users, without the publisher being aware of the privacy preferences of those users. Moreover, this content is often accessible to multiple users. This renders it difficult for GeoSN users to control which information about them is available and to whom it is available. This paper addresses two privacy threats that occur in GeoSNs: *location privacy* and *absence privacy*. The former concerns the availability of information about the presence of users in specific locations at given times, while the latter concerns the availability of information about the absence of an individual from specific locations during given periods of time. The challenge addressed is that of supporting privacy while still enabling useful services. We believe this is the first paper to formalize these two notions of privacy and to propose techniques for enforcing them. The techniques offer privacy guarantees, and the paper reports on empirical performance studies of the techniques.

## Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Spatial databases and GIS*; K.4.1 [**Computers and Society**]: Public Policy Issues—*Privacy*

## General Terms

Algorithms

## Keywords

Social networks, Location privacy, Absence privacy


*CIKM'10,* October 26–30, 2010, Toronto, Ontario, Canada.


## 1. INTRODUCTION

Geo-aware social networks (GeoSNs) are enabled by the availability of social network services, mobile devices with Internet connectivity, and geo-location capabilities. GeoSN users generate and share very large volumes of content, or resources, tagged with geo-location. Thus, resources such as status messages, photos, and "check-ins" are tagged with the location in which they were generated. Further, resources may reference other users—for example, this occurs when a user tags a photo with the people in the photo.

A variety of services exist and can be envisioned that exploit GeoSN resources. For example, the Google Picasa service lets users share and search photos that are both geo-tagged and tagged with other users. Dozens of other such commercial applications exist, including Brightkite, Flickr, Foursquare, Google Buzz, Google Latitude, Gowalla, Loopt, Twitter, and Whrrl. Indications are that geo-tagging is also coming to Facebook[1]. In addition, some GeoSN services enable third-party services that exploit GeoSN resources.

Privacy in social network services has become a hot topic, and reports indicate that users are leaving social network services due to privacy concerns. In GeoSNs, it is possible for exact locations of users to be exposed to untrusted entities that may in turn utilize these to infer sensitive information about the users. For example, the presence of a user in certain locations, e.g., a hospital or a night club, may reveal sensitive information about the user. And because GeoSN resources are easily spread among users in real time, additional threats such as stalking or assault are possible. In addition, the instant publishing of GeoSN resources can enable an adversary to infer how far users are from their homes (or other locations) and hence how long these are possibly unattended. This information represents an absence privacy violation and may be used to plan a burglary[2].

It may be argued that users should be aware of the privacy implications of making resources available and should simply behave responsibly, and thus should not publish resources that may cause privacy concerns. However, this arrangement is undesirable for two main reasons. First, GeoSN service providers are generally interested in as much content as possible being available, as this attracts users and thus increases advertising revenue. Second, in most GeoSN services, users can reference other users in resources; and it is

[1]See `www.facebook.com/policy.php`; `tcrn.ch/99VSSN`
[2]See `PleaseRobMe.com`

generally not possible for a user to control the resources published by another user. For instance, imagine that Alice and Bob are having a drink together in a pub. Alice updates her GeoSN status, writing "having a beer with Bob" and tagging the post with 10:05 p.m., Manhattan. A bit later, Bob sees his friend Charlie and invites him to join them at the table. Charlie then updates his status, writing "just met Bob at a pub!" and tagging the post with 10:10 p.m., 24 West 35th Street. A user with access to both posts (e.g., a friend of Bob) can infer that Alice is at a pub with Charlie as well as the exact address of the pub, although Alice used a coarser location in her tag.

For these reasons, techniques that enable users to specify their privacy preferences and then enforce these preferences are desirable for service providers and users alike. The objective of this paper is to provide such techniques.

The algorithms proposed combine meta-data generalization over spatio-temporal granularities with an analysis of dynamic constraints on inter-dependent resources, also taking into account constraints on the maximum velocity of user movement. Absence privacy is enforced by computing appropriate temporal delays in resource publication based on the above analysis.

Although privacy has been studied extensively in location-based services and social networks, we are not aware of any studies that consider location and absence privacy in the GeoSN setting.

Research on privacy in social networks focuses on profile information, which differs from dynamic geo-spatial information such as a user's current location. In the GeoSN setting, specifically in relation to friend-finder services, privacy preserving proximity computation has been studied [6–8], since friend-finders are typical GeoSN services. However, the solutions proposed do not apply to resource publication.

Research on privacy in location-based services (LBSs) is also not directly applicable to the GeoSN setting. Many techniques proposed for LBSs assume that users are anonymous and offer anonymity by extending the principle of $k$-anonymity to the geo-spatial setting, thus avoiding user re-identification through location data [4,5]. These techniques are not suitable for our problem, since the assumption of user anonymity is not realistic for most existing social networks.

Other techniques for LBSs assume that the identities of the users are known and offer location privacy by obfuscating the users' locations [3]. We adopt this *location privacy* model that is also the reference model for privacy-preserving proximity services. Among the LBS solutions adopting this model, the proposals by Ghinita et al. [2] are arguably the most related since they also consider *dynamic* privacy attacks based on the maximum velocity of the users. However, like other LBS proposals, these proposals do not take into account resources related to multiple users at the same time as in GeoSNs, and they do not consider temporal uncertainty and absence as privacy preferences. Finally, some existing GeoSN services offer some form of control of the geo-tags of resources, e.g., by enabling tags at coarse granularities such as the city level (e.g., Google Latitude), but much finer controls are necessary to avoid the privacy threats considered in this paper.

The contributions of the paper are the following:

- Formalization of location and absence privacy attacks in the GeoSN setting.
- Proposals of means of expressing privacy preferences.
- A privacy preserving technique with formal guarantees on the enforcement of user preferences.
- Empirical studies that suggest that the proposed methods are applicable in realistic scenarios.

The rest of the paper is organized as follows. Section 2 formally characterizes the privacy problems in GeoSN resource publication and also covers how users can specify their privacy preferences; Section 3 describes the architecture of the proposed GeoSN privacy preservation technique; Section 4 reports the technical details of the algorithms that compose our technique; Section 5 describes experimental results, and Section 6 concludes the paper.

# 2. PROBLEM FORMALIZATION

We formally describe the assumed GeoSN service setting and the privacy threats we address. We then define means for the users to express their privacy preferences, the adversary model, and sufficient conditions for satisfying a user's privacy preferences.

## 2.1 The GeoSN service setting

A GeoSN service allows its users to publish a resource (e.g., a picture, a text message, a check-in) tagged with the current location and time, as well as a set of users related to the resource. A resource is either tagged automatically (e.g. an integrated GPS can provide location and time), or tagged manually. Since resources and their tags become available to other users as well as to service providers, we are concerned with the privacy violations that the publication can lead to.

Formally, a *resource* $r$ is a tuple:

$$\langle Udata, STdata, Content \rangle,$$

where the first two elements are meta-data tags with $r.Udata$ being a set of identifiers of users, $r.STdata$ being a spatio-temporal tag and $r.Content$ being the resource itself. In the following, when referring to a resource $r$, we also denote with $r.Sdata$ and $r.Tdata$ the spatial and temporal components, respectively, of $r.STdata$. We assume that all the users in $r.Udata$ are in the location $r.Sdata$ at the time $r.Tdata$.

As an example, recall the user Charlie performing a status update informing his friends about his presence in the pub together with Alice and Bob. In our formalization, the update is a resource with Alice, Bob, and Charlie as $r.Udata$, and the location of the pub with the current time as $r.STdata$.

We consider techniques for privacy preservation based on the *generalization* of resources before publication. In particular, we consider generalization functions that generalize the spatio-temporal tag of a resource. Formally, $STdata$ for an *original resource* is a point in the spatio-temporal domain, while $STdata$ for a *generalized resource* is a 3D volume in the spatio-temporal domain that contains the point of the corresponding original resource.[3] In case of generalized resources $r'$, we denote by $r'.T_{max}$ and $r'.T_{min}$ the maximum and minimum time instant of $r'.Tdata$, respectively.

*Definition 1.* Given the domain $OR$ of all possible original resources and the domain $GR$ of all possible generalized

[3]We assume that location and time of the original resource are recorded at the finest available resolution and is approximated by a point.

resources, a partial function $g : OR \rightarrow GR$ is a *generalization function* if, for each $r \in OR$ such that $g(r)$ is defined, $r.Udata = g(r).Udata$, $r.Content = g(r).Content$ and $r.STdata \subseteq g(r).STdata$.

The generalization function $g(r)$ will take into account the privacy preferences of the users involved in $r$, as well as all the generalizations of the original resources submitted before $r$. To avoid the case of two original resources being submitted at the same time, we assume a total ordering among the submission times of original resources.

## 2.2 Privacy concerns and user preferences

When an adversary associates a user's identity with user's private information, a privacy violation has occured. When it is not possible to exclude that the user's identity can be obtained by the adversary (as it is often the case in social networks), privacy must be preserved by obfuscating the private information. We consider location and time as private information (in the following we will for brevity often use "location" to refer to a spatio-temporal location), and we consider two privacy notions: *location privacy* and *absence privacy*.

A *location privacy* concern exists when uncontrolled disclosure of the geographic position of a user at specific times can occur. Most of the currently available GeoSNs suffer from this privacy concern. A typical example is a user who elects to not let people know that he attended a religious ceremony or a political meeting.

An *absence privacy* concern exists when uncontrolled disclosure of the absence of a user from a geographic position at specific times can occur. This concern is conceptually different from location privacy and requires different protection techniques. A typical example is a user not wishing to let people know that he will not be at home for an extended period of time.

These privacy concerns can be addressed by offering the users means of controlling the location information to be disclosed. These means include different kinds of preferences the users can express: (a) for certain regions, the user's location and time should be revealed only at a "sufficiently coarse" granularity, (b) for certain regions, the user's absence should not be revealed. Hence, preferences for location privacy should specify the minimum uncertainty that an adversary should have about the location of the user upon publication of a resource (possibly considering the resource publication history). In contrast, preferences for absence privacy should specify regions and time intervals such that an adversary cannot exclude any point of the region as the user's location during the associated interval, independently of the publication of resources during the interval.

For both type of preferences, we introduce the notion of *minimal uncertainty region* (MUR) as a spatio-temporal region for which an adversary cannot exclude any internal point as the location of the user. In the solution we propose, each user can express location privacy preferences by specifying a partition of the spatio-temporal domain that define the desired MURs. For example, Alice specifies that any resource in which she is tagged should not report the specific campus building where she was at 10:30 a.m. today; in this case, one solution is to define the combination of the entire campus region and the time period "this morning" as one of the MURs. In other words, when defining a MUR, a user accepts that the adversary learns that she is located in that MUR, but requires that no location within the MUR can be excluded as a possible location.

The MURs for a user can be captured formally with the notion of *spatio-temporal granularity*. Intuitively, a spatial (temporal) granularity is a partitioning of the spatial (temporal) domain into a discrete set of non-overlapping granules [1]. Each granule is associated with an index $i$, and the $i$-th granule of a granularity $G$ is denoted as $G(i)$.

For clarity, we consider spatio-temporal granularities where the granules are products of granules of spatial and temporal granularities. Thus, we assume that a user $u$ specifies a spatial granularity $G_u^S$ and a temporal granularity $G_u^T$ as privacy preferences. The spatio-temporal granularity $G_u$ that represents the user's privacy preference is then derived from $G_u^S$ and $G_u^T$ as follows: for each index $j$ of $G_u^S$ and each index $k$ of $G_u^T$, $G_u(\langle j,k\rangle) = G_u^S(j) \times G_u^T(k)$. In this case the set of indexes of $G_u$ is a set of pairs of numbers, and each granule of $G_u(\langle j,k\rangle)$ is the temporal extension of $G_u^S(j)$ for the time interval defined by $G_u^T(k)$. When no confusion arises, we denote the granules of $G_u$ with a single index $i = \langle j,k\rangle$. Since the granularities partition the spatial and temporal domains, for each spatio-temporal point $p$, there exists a granule of the granularity $G_u$ that contains $p$. We denote this granule with $G_u[p]$. Definition 2 formally states when a location privacy preference is enforced.

*Definition 2.* Let $u$ be a user and $r$ be an original resource with $u \in r.Udata$. The location privacy preference of $u$, expressed as the spatio-temporal granularity $G_u$, is *enforced* if the adversary cannot exclude any point of $G_u[r.STdata]$ as the possible location of $u$.

Next, a user can express an absence privacy preference in terms of a set of granules of a spatio-temporal granularity, and we call each granule an *Absence Privacy Region* (APR). The intuitive semantics is the following: no location information should ever be disclosed about the user such that, for any point $p$ in an APR, $p$ can be excluded as a possible location of the user. Note that the difference with respect to location privacy is that the user requires that, independently of the previous or current publication of resources in locations outside or inside the APR, no point of an APR can be excluded as a possible location of the user. Definition 3 formally states when an absence privacy preference is enforced.

*Definition 3.* Let $u$ be a user and $A$ be the set of APRs chosen by $u$. The absence privacy requirement of $u$ is *enforced* if, for every point $p$ of each APR in $A$, the adversary, when considering all published resources, cannot exclude that user $u$ is located at $p$.

## 2.3 Adversary model

We assume that the adversary has access to all the resources published by all the users. This is a conservative approach, and in some cases, it would be possible to exploit Access Control (AC) techniques to identify classes of adversaries that can only access a subset of the resources published by the users. The above assumption accounts for the case of collusion among adversaries in different classes.

We assume that the adversary knows the technique used to generalize resources before publication. Also, since we do not assume that the users' privacy preferences are kept secret, we conservatively assume that the adversary can obtain

the privacy preferences. In general, we assume that the adversary has the knowledge to compute $g(r)$ for each $r \in OR$. We also assume that the adversary knows, for each user, the maximum velocity $v$ at which that user can move. While users can have different maximum velocities, we assume for simplicity that a single velocity $v$ applies to all users.

As opposed to adversary models that assume a uniform distribution of user locations, our defense techniques also allow adversaries having a non-uniform a-priori probabilistic distribution function $P$ that gives for each user $u$ the probability that at a given time instant $t$, the user is in a certain spatial location $s$. Since we assume that a user can publish a resource any time and from any location the user is at, the same probability distribution applies to the publication of resources: for each potential original resource $r$, $P[r.STdata = p]$ denotes the a-priori probability that $r$ is issued from a spatio-temporal point $p$. Note that if the adversary knows from the a-priori knowledge that a user has zero probability of being located in a certain location at a given time (and hence no resource tagging him can be published from there) then it is not possible to guarantee the location privacy of that user. Hence, for each potential original resource $r$ and each spatio-temporal point $p$, we assume $P[r.STdata = p] > 0$.

To summarize, we assume that the adversary has exactly the following knowledge:

- the set $R'$ of all the published resources
- $g(r)$ for each $r \in OR$
- the users' maximum velocity $v$
- the a-priori probabilistic distribution function of original resources $P[r.STdata = p] > 0$.

Finally, we note that how to infer spatio-temporal information from the content of a resource, e.g., a photo, is an orthogonal research topic. If such information is made explicit, it can be treated as a tag and thus handled by the paper's proposal.

## 2.4 Location privacy preservation

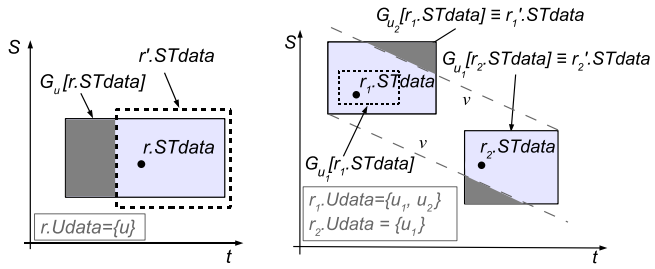
A necessary condition to guarantee location privacy is that each original resource $r$ is generalized into a resource $r'$ such that, for each user $u$ tagged in the resource, the MUR of $u$ that contains $r.STdata$ must be contained in $r'.STdata$. Formally:

$$\forall u \in r.Udata\ (G_u[r.STdata] \subseteq r'.STdata) \qquad (1)$$

When Equation 1 holds, we say that $r'$ *covers its users' MURs.*

As an intuition for this necessary condition, consider Figure 1(a) that, for the sake of simplicity, shows a one-dimensional spatial domain. The attribute $r.STdata$ is generalized into a spatio-temporal region $r'.STdata$ that only partially covers the granule representing the MUR required by the user $u$ that is tagged in $r$. Consequently, the adversary can exclude $u$ from the dark gray region, hence violating the location privacy requirement of $u$.

The fact that a generalized resource $r'$ covers its users' MURs is a necessary but insufficient condition to guarantee location privacy. Consider the example in Figure 1(b) that shows two generalized resources $r_1$ and $r_2$. User $u_1$ is tagged in both resources, while user $u_2$ is tagged in $r_1$ only. Resource $r_1$ and $r_2$ is generalized into $r'_1$ and $r'_2$, respectively. It is easily seen that each generalized resource covers




(a) Single resource (b) Set of resources

**Figure 1: Example of privacy violations**

its users' MURs. However, location privacy of neither $u_1$ nor $u_2$ is enforced if we take into account the maximum velocity $v$. First, the adversary knows that during time period $r'_1.Tdata$, user $u_1$ is in $r'_1.Sdata$. Now, if resource $r_1$ is issued from the bottom left corner of $r'_1.STdata$, it is impossible for user $u_1$ to reach, with the maximum velocity $v$, the dark gray region of $r'_2.STdata$. In other words, not all points of $r'_2.STdata$ are *reachable* from $r'_1.STdata$, and hence some points can be excluded as possible locations of $u_1$. A similar attack can be performed to exclude some parts of $r'_1.STdata$. Indeed, no point of $r'_2.STdata$ can be reached from any point of the dark gray region in $r'_1.STdata$. Since the adversary knows that at the time of $r_1$ the two users are in the same location, the adversary can exclude $u_2$ from the dark gray region of $r'_1$, hence violating the location privacy of $u_2$.

The above example shows that two resources can be *dependent* if they have at least one user in common and if their temporal distance is small when compared with the spatial distance. In order to capture this intuition formally, we first define the notion of *reachability* and then the notion of *independence* of resources.

*Definition 4.* Given a velocity $v$ and two spatio-temporal regions $str_1$ and $str_2$, we say that $str_1$ is *reachable* from $str_2$ if

$$\forall p_1 \in str_1\ (\exists p_2 \in str_2\ (d_s(p_1, p_2) \leq v \cdot d_t(p_1, p_2))), \qquad (2)$$

where $d_s$ and $d_t$ denotes the spatial and temporal distance between two points, respectively.

In the following, we say that a resource $r_1$ is reachable from a resource $r_2$ if $r_1.STdata$ is reachable from $r_2.STdata$. The transitivity property applies to reachability in case resources are totally ordered as defined in Property 1.

PROPERTY 1. *(Transitivity) Let $r_1$, $r_2$, and $r_3$ be generalized resources such that: $r_1.T_{max} \leq r_2.T_{min}$ and $r_2.T_{max} \leq r_3.T_{min}$. Then,*
*(a) if $r_3$ is reachable from $r_2$ and $r_2$ is reachable from $r_1$ then $r_3$ is reachable from $r_1$;*
*(b) if $r_1$ is reachable from $r_2$ and $r_2$ is reachable from $r_3$ then $r_1$ is reachable from $r_3$.*

We can now define independent resources as follows.

*Definition 5.* Two resources $r_1$ and $r_2$ are *independent* if at least one of the following conditions hold:
a) $r_1.Udata \cap r_2.Udata = \emptyset$;
b) $r_1$ is reachable from $r_2$, and $r_2$ is reachable from $r_1$.

Intuitively, Definition 5 states that two resources having at least one user in common are independent if they are reachable from each other.

One last necessary condition to guarantee location privacy is related to the generalization function $g()$. Consider the following example, in which we exploit the fact that the adversary can compute $g(r)$ for each potential original resource $r \in OR$. Let $r'$ be a published generalized resource, $p$ a point in $r'.STdata$, and $r$ a potential original resource that is identical to $r'$ except $r.STdata = p$. Also assume that $g(r).STdata \neq r'.STdata$. The adversary can exclude $p$ as a possible location of the users in $r'.Udata$ since if the users posted the resource from $p$ then it would be generalized to something different from $r'$. Clearly, this can violate the location privacy requirement.

In order to guard against this kind of attack, the generalization function must have the property known in the literature as "non-invertibility" or "reciprocal" [4, 5]. This property is stated in Definition 6.

*Definition 6.* A generalization function $g()$ is *non-invertible* if for each pair of original resources $r_1$ and $r_2$ such that $r_2.Udata = r_1.Udata$, $r_2.Content = r_1.Content$, and $r_2.STdata \in g(r_1).STdata$, it holds that $g(r_1).STdata = g(r_2).STdata$.

We can now define a set of conditions that are sufficient to guarantee location privacy.

THEOREM 1. *Let $R' = \{g(r)|r \in R$ and $g(r)$ is defined$\}$ be the set of all published resources, where $R$ is the set of corresponding original resources, and $g()$ is a generalization function. If all the following conditions hold then the location privacy of all the users tagged in these resources is enforced:*
*(a) for each $r'$ in $R'$, $r'$ covers its users' MURs*
*(b) each pair of distinct resources $r'_1$ and $r'_2$ in $R'$ are independent*
*(c) $g()$ is non-invertible*

If this is satisfied, we say that $R'$ is a *safe* set.

The proofs of Theorem 1 and two following theorems are omitted due to space limitations.

## 2.5 Absence privacy preservation

We consider the class of absence privacy preferences that model the following situation: at each time instant, a user wants to prevent the adversary from learning, that, *at that time*, the user is not in the APRs. This protects from the problem of unattended locations (like a user's home) that arises in the case of instant publishing of resources.

In our model, this privacy preference can be specified easily. Intuitively, the user defines a spatial region $s$. Then, at each time instant $t$, the APR corresponding to $s$ is the spatio-temporal region formed by $s$ at time $t$. This can be easily extended to a set $S$ of spatial regions, and we denote by $S_u$ the set of spatial regions specified by user $u$ to protect absence privacy.

We now define a set of conditions that are sufficient to guarantee that absence privacy is enforced. Intuitively, if all the published resources are such that location privacy is enforced then by definition, for each resource $r'$ in which a user $u$ is tagged, no location in $r'.STdata$ can be excluded as a possible location of $u$. Then, in order to guarantee that absence privacy is enforced, we postpone the publication of the resources until each APR of $u$ is reachable from the $STdata$ of each resource in which $u$ is tagged. Theorem 2 captures this intuition.

THEOREM 2. *Let $R'$ be the set of published resources and $u$ be a user. Then absence privacy is enforced if the following two condition hold:*
*(a) $R'$ is such that location privacy is enforced*
*(b) the publication time $t$ of each $r' \in R'$ is such that $\forall s \in S_u$, the spatio-temporal region given by $s \times t$ is reachable from $r'.STdata$*

# 3. ARCHITECTURE

Figure 2 shows the architecture of the privacy enforcing system, including the interaction of the users with the system. We assume the presence of a centralized trusted entity that is in charge of processing the original resource and publishing it to the GeoSN after the generalization process described in the following.

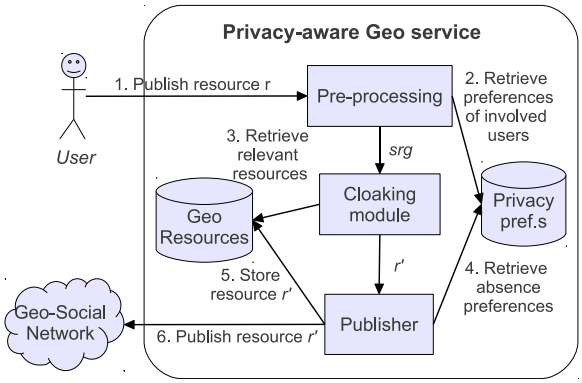



**Figure 2: System architecture**

When a user wants to publish a resource $r$, possibly related to multiple users, the user sends the resource to the trusted system. A pre-processing module retrieves the privacy preferences of the involved users and computes the generalization $srg$ that covers its users' MURs.

The cloaking module checks whether the disclosure of $srg$ introduces privacy violations when related to other resources previously processed. If so, the module applies a generalization algorithm in an attempt to produce a more general resource that eliminates the violation. We show in Section 4 that there are cases in which such generalization will not be possible, and hence the publication should be denied. In all the other cases, the cloaking module finds a safe generalization $r'$ and sends it to the Publisher module.

The Publisher module checks whether the users' absence privacy preferences are violated by the immediate publication of the resource and computes a publication time after which it is safe to publish the resource. The final resource is then stored in the local database, since it must be used to check new resources for possible violations. After the publication time has passed, the resource is published.

The privacy-enforcing trusted system needs to know the $Udata$ and $STdata$ of each resource to be processed, but it does not rely fundamentally on the content of a resource. Thus, a variant of the system can return instructions to a user for how to safely publish a resource when simply given the $Udata$ and $STdata$ of the resource. This variant is attractive if the user does not want to reveal the content of a resource, e.g., a photo, to the system.

This also makes it possible to implement our techniques as a third-party application for existing GeoSNs. Indeed,

many existing GeoSNs allow third-party applications to have limited access to the users' data and to post resources on behalf of the users upon being authorized to do so.

# 4. THE WYSE TECHNIQUE

This section presents WYSE (Watch Your Social stEp), our proposed privacy-preservation technique.

## 4.1 Overview

The general idea is the following. Given a resource $r$ to be published and a stored, safe set $R'$ of generalized resources, the technique produces a generalized resource $r'$ such that $\{r'\} \cup R'$ remains safe. As will be explained, it may happen that it is impossible to find a proper $r'$. In this case, publication is denied.

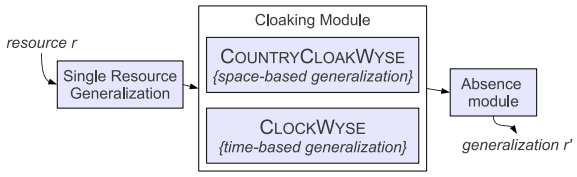



**Figure 3: Steps of Wyse**

Figure 3 presents the different steps of the technique. First, the resource is generalized to a resource covering its users' MURs. This step is performed by the Single Resource Generalization (SRG) module. Then the resource is sent to a cloaking module, where an additional spatial or temporal generalization is applied, if necessary. Finally, the resource passes through the absence module, where the publication time of the resource is defined so that absence privacy is enforced. We proceed to cover these steps in detail.

## 4.2 Single resource generalization

The SRG module performs a generalization of the original resource $r$ according to the privacy preferences of each user associated with the resource and produce as output a generalized resource *srg* that covers its users' MURs.

To obtain this result, a granularity $G_r$ is computed such that each granule of $G_r$ is the union of a set of granules of $G_u$ for each $u$ in $r.Udata$. Then, the resource *srg* is produced such that $srg.STdata = G_r[r.STdata]$.

In order to guarantee that a proper granularity $G_r$ exists for each possible set $r.Udata$, we limit the possible spatial and temporal granularities each user can choose as location privacy preferences. In brief, we assume that each user $u$ can choose a spatial granularity $G_u^S$ in a set of granularities that are totally ordered. If a granularity $G_1$ precedes $G_2$ with respect to this order then every granule of $G_2$ is equal to the union of some granules of $G_1$. The same applies for temporal granularities. With these assumptions, the granularity $G_r$ exists and is obtained as the combination of the coarser spatial granularity $G_u^S$ for each user $u \in r.Udata$ and the coarser temporal granularity $G_u^T$ for each user $u \in r.Udata$, denoted in the following with $G_r^S$ and $G_r^T$, respectively. As $srg.STdata$ is the granule $G_r[r.STdata]$, for simplicity we refer to this as the granule of *srg*, and we denote it as $G_r[srg.STdata]$.

## 4.3 Cloaking

As observed in Section 2, privacy violations occur when resources are not independent. The aim of the cloaking module is to find a generalized resource, derived from *srg* and $G_r$, that does not cause a privacy violation when combined with resources in $R'$. The resulting generalized region is equivalent to or larger than $srg.STdata$. We propose two different cloaking algorithms, called CountryCloakWyse and ClockWyse, that apply a generalization only on the spatial or temporal dimensions, respectively.

In order to reduce the number of resources belonging to $R'$ that need to be considered during the generalization process, both algorithms exploit the transitivity property of reachability (Property 1). Indeed, to obtain the independence with respect to all resources, it is sufficient to identify a set of *relevant resources* and to enforce independence with respect to the resources in this set. To formally define the relevant resource, we need the following notation: resource $r_1$ "precedes" resource $r_2$ if $r_1.T_{max} \leq r_2.T_{min}$ and resource $r_3$ "succeeds" resource $r_4$ if $r_3.T_{min} \geq r_4.T_{max}$. The set of relevant resources can then be defined as the union, for each user $u \in r.Udata$, of the resources $r^i \in R'$ such that $u \in r^i.Udata$ and one of the following conditions holds:

- (*concurrency*): $r^i.Tdata \cap srg.Tdata \neq \emptyset$,
- (*preceding*): $r^i$ precedes *srg* and there does not exist any resource that precedes *srg* and succeeds $r^i$,
- (*succeeding*): $r^i$ succeeds *srg* and there does not exist any resource that succeeds *srg* and precedes $r^i$.

In addition, in order to efficiently compute the independence of the relevant resources, we exploit Property 2 that reduces the problem of evaluating reachability to the computation of a single inequality.

PROPERTY 2. *Given a velocity $v$ and two resources $r_1$ and $r_2$, $r_1$ is reachable from $r_2$ if and only if the following condition holds:*

$$h_s(r_1.STdata, r_2.STdata) \leq v \cdot \max\left(\frac{|r_2.Tdata|}{2}, h_t(r_2.STdata, r_1.STdata)\right),$$

*where $|r_2.Tdata| = r_2.T_{max} - r_2.T_{min}$ and $h_s$ and $h_t$ are the direct spatial and temporal Hausdorff distances, respectively.*

The direct spatial (temporal) Hausdorff distance between $r_1.STdata$ and $r_2.STdata$ is the maximum distance between any point in $r_1.Sdata$ ($Tdata$) to its closest point in $r_2.Sdata$ ($Tdata$). The spatial Hausdorff distance has been exploited in literature to describe attacks based on the knowledge of maximum velocity [2]. Property 2 is an extension for spatio-temporal regions that also considers the direct temporal Hausdorff distance.

The next two sections illustrate our two proposed generalization algorithms.

## 4.4 Spatial generalization: CountryCloakWyse

CountryCloakWyse generalizes $Sdata$ with respect to the *srg* while preserving $Tdata$. This kind of generalization is thus suitable for services in which a high time accuracy is desirable, such as microblogging services.

The pseudocode can be found in Algorithm 1. Overall, the idea of the algorithm is the following. First, from the relevant resources, two sets of granules of $G_r$ are identified, among the set $G_{srg.Tdata}$ of the granules obtained as the product of each spatial granule of $G_r^S$ with $G_r^T[srg.Tdata]$, The set $S_{safe}$ contains granules such that if the granule of *srg*

**Algorithm 1** CountryCloakWyse

**Input**: The resource *srg*, the granularity $G_r$ and a safe set of resources $R'$.
**Output**: A resource $r'$ such that $R' \cup \{r'\}$ is a safe set.

1: compute $G_{srg.Tdata}$
2: $S_{safe}, S_{gener} \leftarrow G_{srg.Tdata}$
3: $R_R = \{$ set of resources in $R'$ that are relevant to *srg*$\}$
4: **for** each $r_R \in R_R$ **do**
5: $\quad SI_{gener} = \{g \in G_{srg.Tdata} : g$ is reachable from $r_R\}$
6: $\quad SI_{safe} = \{g \in G_{srg.Tdata} : g$ is independent from $r_R\}$
7: $\quad S_{gener} = S_{gener} \cap SI_{gener}$
8: $\quad S_{safe} = S_{safe} \cap SI_{safe}$
9: **end for**
10: $S_{gener} = S_{gener} \setminus S_{safe}$
11: **if** $(G_r[srg.STdata] \notin S_{gener} \cup S_{safe})$ **or** $(S_{safe} = \emptyset)$ **then**
12: $\quad$ **return null** {Deny the publication}
13: **end if**
14: $cg =$ NN of *srg* in $S_{safe}$
15: $r'.Sdata = \bigcup\{$ RNN of $cg$ in $S_{gener}\}$
16: $r'.Sdata = r'.Sdata \cup cg$
17: $\langle r'.Tdata, r'.Content\rangle \leftarrow \langle srg.Tdata, srg.Content\rangle$

is one of these granules then it would not require any generalization. The set $S_{gener}$ contains the granules such that if the granule of *srg* is one of these granules then a generalization is required to guarantee independence (lines 2–10).

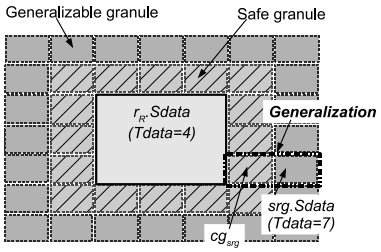



**Figure 4: Spatial generalization**

If the granule of *srg* is neither in $S_{gener}$ nor $S_{gener}$, or if $S_{safe}$ is empty, then the publication of the resource is denied (lines 11–12). Otherwise, due to Property 3, if $srg.STdata$ is a granule of $S_{gener}$, then, to guarantee the independence of the resource, it is sufficient to generalize $srg.STdata$ so that a granule of $S_{safe}$ is included. In order to achieve this with a non-invertible algorithm, the following is performed: the granule $cg$ is computed as the granule in $S_{safe}$ that is the nearest neighbor (NN) of $srg.STdata$ (this can be $srg.STdata$ itself, if it is a granule in $S_{safe}$). Then the generalization is computed as the union of $cg$ with the set of granules of $S_{gener}$ that considers $cg$ as the closest granule in $S_{safe}$. This is computed with a reverse nearest neighbor (RNN) query (lines 14–15). Both the NN and the RNN queries are computed considering the centers of the granules as source and targets.

Property 3. *Let A, B, and C be spatio-temporal regions such that A is reachable from C and A is not reachable from B. Then A is reachable from $B \cup C$.*

An example of the algorithm is illustrated in Figure 4. The figure represents the safe and generalizable granules for *srg* and a related resource $r_R$, the closest safe granule to *srg* and the resulting generalization. In general, the generalizable granules are partitioned with respect to their closer safe granule, and *all* the granules that belong to a partition are retrieved as the output of the generalization.

## 4.5 Temporal generalization: ClockWyse

ClockWyse performs temporal generalization technique and is suitable for services in which it is desirable to preserve the accuracy of the spatial location, like in check-ins, rather than maintaining an accurate temporal interval. In this case, a generalization is computed considering the set $G_{srg.Sdata}$ of the granules obtained as the product of each temporal granule of $G_r^T$ with $G_r^S[srg.Sdata]$.

The pseudocode is shown in Algorithm 2. The idea of the algorithm is the following: as inCountryCloakWyse, the two set of granules $S_{safe}$ and $S_{gener}$ are computed from the relevant resources. However, in this case, since the generalization is performed on a single dimension (the temporal dimension) $S_{safe}$ and $S_{gener}$ are intervals of time granules and hence can be specified in terms of the intervals' boundaries. Another difference with respect to CountryCloakWyse is that to compute the two sets, it is necessary to consider the three cases in which a relevant resource is preceding, succeeding, or concurrent wrt. *srg* (lines 3–16).

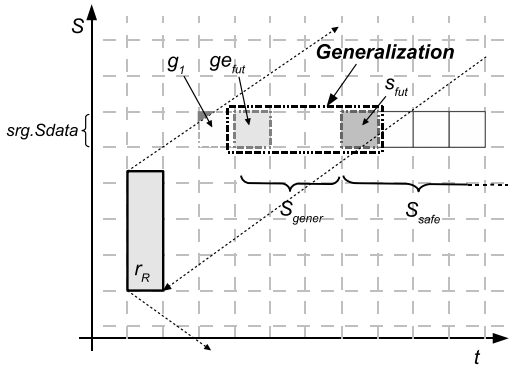



**Figure 5: Temporal generalization (one-dimensional space)**

We explain the algorithm for the case in which a relevant resource $r_R$ precedes *srg*; the other cases are analogous. See Figure 5 that shows, for the sake of simplicity, a single spatial dimension. If $g_1$ is the granule of *srg* then $srg.STdata$ is not reachable from $r_R$. Hence, even if *srg* is generalized, it is still unreachable from $r_R$, and the publication of the resource has to be denied (lines 17–18). All of the granules to the right of $g_1$ are reachable from $r_R$. However $r_R$ is not reachable from all of them. Indeed, $r_R$ is not reachable from the granules to the left of $s_{fut}$. According to Property 3, these granules can be generalized together with a granule in $S_{safe}$ to enforce location privacy. The solution adopted to achieve this with a non-invertible algorithm is the following: if the granule of *srg* is $s_{fut}$ or in $S_{gener}$ then the temporal domain is generalized to the "Generalization" box, i.e., the union of $s_{fut}$ and all the granules in $S_{gener}$ (lines 19–20). Otherwise, if the granule of *srg* is in $S_{safe}$, but is not $s_{fut}$, no temporal generalization is applied (line 23–24).

A numerical example showing a two-dimensional space can be seen in Figure 6. The striped lines centered in each resource $r'$ with radius $t = x$ enclose, for a user who is inside $r'$ and can travel at a maximum velocity of $v$, the possible locations after $x$ time instants. Graphically, these regions

**Algorithm 2** ClockWyse

**Input**: The resource *srg*, the granularity $G_r$ and a safe set of resources $R'$.
**Output**: A resource $r'$ such that $R' \cup \{r'\}$ is a safe set.

compute $G_{srg.Sdata}$
$R_R = \{$ set of resources in $R'$ that are relevant to *srg*$\}$
**for all** $r_R \in R_R$ **do**
  **if** (*srg* is concurrent with $r_R$) **or** $(h_s(r_R, srg) \leq v \cdot \frac{|srg.Tdata|}{2})$ **then**
    $s_{past}, s_{fut} \leftarrow$ past/future safety boundaries
    $ge_{fut} \leftarrow$ future generalization boundary
    $ge_{past} \leftarrow$ Previous granule of $ge_{fut} - 1$
  **else if** $srg.T_{min} \geq r_R.T_{max}$ **then**
    $ge_{fut} \leftarrow$ future generalizable boundary
    $s_{fut} \leftarrow$ future safety boundary
  **else**
    $ge_{past}, \leftarrow$ past generalizable boundary
    $s_{past} \leftarrow$ past safety boundary
  **end if**
  Update generalizable and safety boundaries
**end for**
**if** $(Ge_{past} < srg < Ge_{fut})$ **or** $(S_{fut} > S_{past})$ **then**
  **return null** {Deny the publication}
**else if** $Ge_{fut} \leq srg \leq S_{fut}$ **then**
  $r'.Tdata = \bigcup\{g \in G_{srg.Sdata} : Ge_{fut} \leq g \leq S_{fut}\}$
**else if** $S_{past} \leq g \leq Ge_{past}$ **then**
  $r'.Tdata = \bigcup\{g \in G_{srg.Sdata} : S_{past} \leq g \leq Ge_{past}\}$
**else**
  $r'.Tdata = srg.Tdata$ {No generalization}
**end if**
$\langle r'.Sdata, r'.Content\rangle \leftarrow \langle srg.Sdata, srg.Content\rangle$

correspond to the Minkowski sum of $r'.Sdata$ and a circular region having radius $v \cdot x$ (to simplify the notation, in the figure we write $t = x$ to refer to $t = v \cdot x$).

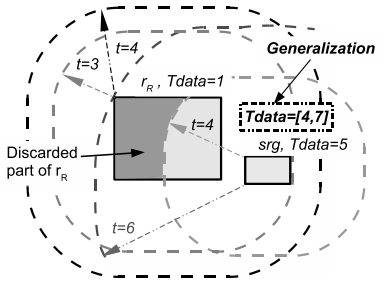



**Figure 6: Example of temporal generalization (two-dimensional space)**

The figure shows the Minkowski sum of $r_R.Sdata$ and a circular region having radius $t = 4$, which is the time difference between $srg.Tdata$ and $r_R.Tdata$. As this region contains *srg*, it means that it is possible to travel from $r_R$ to any point of *srg* in 4 time instances. Therefore, *srg* is reachable from $r_R$ and the resource can be generalized by our technique (if this first condition does not hold, our algorithm denies publication). However, as can be observed in the figure, the opposite does not hold: it is not possible to reach, from *srg*, all the points in $r_R$ in 4 time instants. Thus, $r_R$ is not reachable from *srg*. Intuitively, if *srg* is published, there is a part of $r_R$ that could be discarded as the origin of the resource $r_R$ (a user must have been in the light gray region of $r_R$ in order to be able to reach *srg* by time $t = 5$). Therefore, a generalization technique must be applied. Our temporal generalization obtains the following values: (a) $Ge_{fut} = 3 + 1$, which intuitively corresponds to the minimum radius of a circular region such that the Minkowski sum of $r_R.Sdata$ and that circular region fully covers *srg*, plus $r_R.Tdata$, and (b) $S_{fut} = 6 + 1$, which intuitively corresponds to the minimum radius of a circular region such that the Minkowski sum of $srg.Sdata$ and that circular region fully covers $r_R$, plus $r_R.Tdata$. Then, $srg.Tdata$ is generalized to the interval $[Ge_{fut}, S_{fut}] = [4, 7]$.

## 4.6 Absence privacy enforcement and publication

A resource $r'$ generalized by CountryCloakWyse or ClockWyse is guaranteed to be independent from every other existing resource, but this does not offer any guarantees with respect to the absence privacy preferences $S_u$ of the users (see Section 2.5). To enforce absence privacy, the publication time of the resource must be computed in a way such that every spatial region in the $S_u$ of the users in *Udata* is reachable from $r'$ at that time. Algorithm 3 computes the publication time of a resource $r'$ considering absence privacy preferences.

**Algorithm 3** MinimumPublicationTime

**Input**: A resource $r'$ generalized by the cloaking module
**Output**: The minimum publication time for $r'$

$delay = 0$
**for all** $u \in r'.Udata$ **do**
  $S_u$ = the set of spatial regions chosen by $u$ in the absence privacy preference
  **for all** $apr \in S_u$ **do**
    $delay = \max(delay, \frac{h_s(apr, r'.Sdata)}{v})$
  **end for**
**end for**
**return** $delay + r'.T_{min}$

To determine the publication time of the resource, the maximum value between $r'.T_{max}$ and the value of MinimumPublicationTime is first stored in a temporary attribute $r'.pTime$. The resource is then stored in the database of generalized resources, and the Publisher component is responsible for releasing the generalized resource to the GeoSN once the time instant $r'.pTime$ is reached. This guarantees that (a) it is not possible to exclude any of the points of $r'.Tdata$ as possible *Tdata* of the original resource, and (b) after *pTime* has passed, each user in *Udata* can be located in any of the spatial regions indicated by the absence privacy preference.

However, if another resource $r''$ relevant to $r'$ is submitted, processed and published before $r'.pTime$, in some particular cases, an adversary may detect, by looking at $r''$, that there is a resource waiting to be published in the system. In the worst case, the adversary can obtain accurate information about *STdata* and *Udata* of $r'$, and this could enable the adversary to determine that a user in *Udata* can not be located in one of the spatial regions indicated in the absence privacy preference. To ensure that this attack cannot be performed, we modify our cloaking algorithms so that, during the generalization of a resource $r'$, the maximum *pTime* of the resources that were considered for the generalization

is stored in a variable called $rel_{pTime}$. The final publication time of $r'$ then becomes the maximum among $rel_{pTime}$, MinimumPublicationTime($r'$), and $r'.T_{max}$.

## 4.7 Correctness of privacy enforcement

The WYSE algorithm enforces privacy because it guarantees the sufficient properties of Theorems 1 and 2. This is formally stated in Theorem 3.

THEOREM 3. *The* WYSE *algorithm enforces location and absence privacy.*

The correctness of the theorem follows from the following reasoning. To offer location privacy, the SRG module generalizes each resource so that it covers its users' MURs (condition (a) of Theorem 1). Also, the cloaking module guarantees that each resource is independent with respect to its relevant resources which, according to Property 1, means that the resource is independent of all the other resources (condition (b) of Theorem 1). As a result, the WYSE algorithm is not invertible (condition (c) of Theorem 1). Intuitively, this holds because the WYSE algorithm implicitly computes a partition of the spatio-temporal domain. Then, the spatio-temporal attribute of each generalized resource $g(r)$ is computed as the block containing $r.STdata$.

According to Theorem 2, it is sufficient to guarantee two conditions in order to enforce absence privacy. The first, which concerns the location privacy of the published resources, follows from the above reasoning. The second, which relates to the publication time, is guaranteed by the Absence module that delays the publication of resources according to user's preferences.

# 5. EMPIRICAL STUDY

We conducted experiments to measure the impact of the WYSE technique on the quality of service, comparing the CLOCKWYSE and COUNTRYCLOAKWYSE variants.

## 5.1 Setting

The experimental evaluation was performed on a survey-driven synthetic dataset of user movements, which was obtained using the MilanoByNight simulation[4]. We carefully tuned the simulator in order to reflect a typical deployment scenario of a GeoSN: 100,000 potential users moving between their homes and one or more entertainment places in the city of Milan during a weekend night. Locations are sampled every minute. The total size of the map is 215 km$^2$, and the average density is 465 users/km$^2$. The simulation also models the time spent at the entertainment places, i.e., when no movement occurs, following probability distributions extracted from user surveys. The observed maximum velocity is 55.05 km/h, which is used as the velocity parameter for the generalization techniques.

We extended the simulation by specifying how often the resources are generated by users. When a resource is generated by a user, the other users that are located in the same place are associated with the resource, and the resource is submitted. However, as it is unlikely that all the users in the same place are associated with each resource generated in that place, we limit the number of users that can be associated with the same resource.

[4]See `everywarelab.dico.unimi.it/lbs-datasim`

The experiments use regular granularities, i.e., all the granules of a granularity have the same size and shape. Each temporal granularity is identified by the duration of time covered by one single granule. Each spatial granularity is identified by the size of the edge of one cell of a grid. The conversions from a spatio-temporal location to the respective spatial and temporal granules are performed in constant time.

For absence privacy, we assume that each user's home is sensitive. For simplicity, we assume that each user selects randomly, at the beginning of the simulation, a spatio-temporal granularity among the ones we consider. To observe the impact of the preferences, we limit the minimum granularity that a user can choose. The parameters used in the experiments are shown in Table 1, with default values in bold. We implemented our algorithms using the Java language, and the studies were performed on a computer with a 2.5 GHz Intel Xeon processor and 32GB of shared RAM, running 64-bit RedHat Enterprise Linux 5 and using 1GB of allocated memory for the JVM.

| Parameter | Values |
|---|---|
| Resources posted by a user in 1 hour | 1,**2**,6,12 |
| Max number of users in $UData$ | 1,**5**,10,20 |
| Min time interval of $G_u^T$ (minutes) | 1,4,**8**,16,32 |
| Min edge of a cell of $G_u^S$ (meters) | 128,256,**512**,1024,2048 |

**Table 1: Parameters and their values**

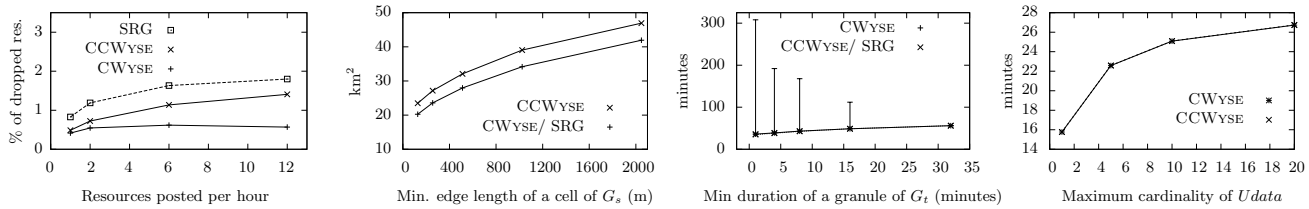
## 5.2 Quality of service

Figure 7(a) shows the percentage of times a technique fails to find a safe generalized resource, considering different resource posting frequencies. In order to observe when the cloaking techniques actually avoid a resource from being dropped, we developed a technique that consists only of the Single Resource Generalization step (SRG) and that drops the resource when the resulting generalization violates privacy. As can be observed, the percentage of dropped resources grows when the resources become more frequent. This is because it then becomes more likely that the relevant resources are close in time and the privacy condition is not satisfied. We can also observe that both the CLOCKWYSE and COUNTRYCLOAKWYSE techniques can often produce a safe generalized resource even when the resource generated by SRG is unsafe. In particular, with our default values, the COUNTRYCLOAKWYSE and CLOCKWYSE algorithms drop about one half of the resources dropped by SRG, and the drop ratio of CLOCKWYSE grows much more slowly than those of the other techniques.

To observe the generalizations applied to the resources, we analyzed the average spatial area, the average temporal duration (i.e., the length of the temporal interval $Tdata$), and the average publication delay.

Figure 7(b) shows the average area using different values for the minimum $G_s$ chosen by users. As it can be observed, the area grows linearly with the edge length of the spatial granules. In addition, the COUNTRYCLOAKWYSE generates, on average, regions having areas larger than the ones generated by SRG. This is expected because the COUNTRYCLOAKWYSE attempts to extend the spatial region to avoid privacy violations and/or dropping.

In Figure 7(c), we measure the average time duration of the generalized resource considering different minimum values of $G_u^T$. With our default parameters and using both algorithms, the average time duration is around 40 minutes,




(a) Drop ratio (b) Area (c) Time interval (d) Publication delay

**Figure 7: Impact on the generalized resource**

with ClockWyse and CountryCloakWyse performing similarly. However, ClockWyse may produce a time interval that is significantly longer than the average. The vertical bars show the maximum time durations observed in the experiments. For this reason, it is suggested to use CountryCloakWyse for those services that perform better with a more accurate time duration.

Figure 7(d) shows the average publication delay of the resources generated by our techniques for varying maximum values of *Udata*. The delay can be caused by both the time generalization due to location privacy enforcement and the delay due to absence privacy enforcement. The results indicate that the larger the number of users in *Udata*, the more delay is added to the resource. This happens mainly because having more users in *Udata* increases the probability that some user has stricter (i.e., coarser) location privacy preferences and also results in more absence privacy preferences that need to be satisfied.

## 5.3 Runtime

We measured the runtime required by each algorithm to process a resource. With our default values, the average computation time for ClockWyse is 23.24 ms, while it is 39.13 ms for CountryCloakWyse. These results suggest that our prototype can support a mid-scale resource publishing service. This is without any optimizations to the code. We believe that the cost of some operations (e.g., the NN and the RNN operations) can be significantly reduced by optimizations, including the use of spatio-temporal indexes.

# 6. CONCLUSIONS

As social networking services continue to proliferate, there is an increasing need for means of affording users privacy. This paper proposes such means for a setting in which social network users can publish resources that have spatio-temporal tags and that reference other users—a prototypical example is a photo with location and time and a listing of who appears in the photo.

This paper addresses two privacy threats in this setting, namely location privacy and absence privacy. The former concerns the availability of information about the presence of users in specific locations at given times, while the latter concerns the availability of information about the absence of an individual from specific locations during given periods of time. The paper is the first to address these threats in this setting.

The paper formalizes the setting, provides a foundation for easily defining privacy preferences, and provides techniques that generalize the tags of resources so that these remain useful while ensuring that the privacy preferences are enforced. These techniques exploit spatial and temporal generalization, and they utilize publication delays. An empirical study characterizes the impact of the proposal on the quality of service, considering the number of resources that are dropped, the extents of the spatio-temporal generalizations, and the publication delays, suggesting that these are relatively modest. Further, the runtime of the techniques is low.

In future research, it is relevant to study how to best enable users to specify their location and absence privacy preferences as supported by the paper's proposal. Next, it may be of interest to consider more general privacy preferences. Finally, it is of interest to consider integration with access control mechanisms.

# Acknowledgments

This work was partially supported by Italian MIUR under grants PRIN-2007F9437X and FIRB-RBFR081L58_002, and by the NSF under grant CNS-0716567. The research was performed when C. S. Jensen was with Aalborg University. C. S. Jensen is an adjunct professor at University of Agder, Norway.

# 7. REFERENCES

[1] C. Bettini, X. S. Wang, and S. Jajodia. *Time Granularities in Databases, Temporal Reasoning, and Data Mining.* Springer, 2000.

[2] G. Ghinita, M. L. Damiani, C. Silvestri, and E. Bertino. Preventing velocity-based linkage attacks in location-aware applications. *Proc. of ACM Int. Symp. on Advances in Geographic Information Systems*, 246–255, 2009.

[3] C. S. Jensen, H. Lu, and M. L. Yiu. Location privacy techniques in client-server architectures. *Privacy in Location-Based Applications*, LNCS 5599, 31–58. Springer, 2009.

[4] 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.

[5] 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.

[6] S. Mascetti, C. Bettini, D. Freni, X. S. Wang, and S. Jajodia. Privacy-aware proximity based services. *Proc. of the 10th Int. Conference on Mobile Data Management*, 31–40, 2009.

[7] L. Šikšnys, J. R. Thomsen, S. Šaltenis, and M. L. Yiu. Private and flexible proximity detection in mobile social networks. *Proc. of the 11th Int. Conference on Mobile Data Management*, 75–84, 2010.

[8] G. Zhong, I. Goldberg, and U. Hengartner. Louis, Lester and Pierre: Three protocols for location privacy. *Privacy Enhancing Technologies*, LNCS 4776, 62–76. Springer, 2007.