Joint Obfuscation of Location and Its Semantic Information for Privacy Protection

Behnaz Bostanipour*, George Theodorakopoulos

School of Computer Science and Informatics
Cardiff University, Cardiff, United Kingdom

Abstract

Location-based social networks (LBSNs) such as Foursquare and Facebook enable users to share with each other, their (geographical) locations together with the semantic information associated with their locations. The semantic information captures the type of a location and is usually represented by a semantic tag like "restaurant", "museum", "school", etc. Semantic tag sharing increases the threat to users' *location privacy* (which is already at risk because of location sharing) and it also puts users' *semantic location privacy* at risk. The existing solution to protect the location privacy and the semantic location privacy of users in such LBSNs is to obfuscate the location and the semantic tag independently of each other in a so called disjoint obfuscation approach. Thus, in this approach, the semantic tag is obfuscated i.e., replaced by a more general tag. Also, the location is obfuscated i.e., replaced by a generalized area (called *the cloaking area*) made of the actual location and some of its nearby locations. However, since in this approach the location obfuscation is performed in a semantic-oblivious manner, an adversary can still increase his chance to infer the actual location and the actual semantic tag by filtering out the locations in the cloaking area that are not semantically compatible with the obfuscated semantic tag. In this work, we address this issue by proposing a *joint obfuscation approach* in which the location obfuscation is performed based on the result of the semantic tag obfuscation. We also provide a formal framework for evaluation and comparison of our joint approach with the disjoint approach. By running an experimental evaluation on a dataset of real-world user traces collected from six different cities, we show that in almost all cases (i.e., in different cities and with different obfuscation parameters), the joint approach outperforms the disjoint approach in terms of location privacy protection and the semantic location privacy protection. Based on the evaluation results, we also discuss how different obfuscation parameters and the choice of the city can affect the performance of the obfuscation approaches. In particular, we show how changing these parameters can improve the performance of the joint approach.

Keywords: Privacy, Social Networks, Location-based Services, Semantics.

[*] Corresponding author

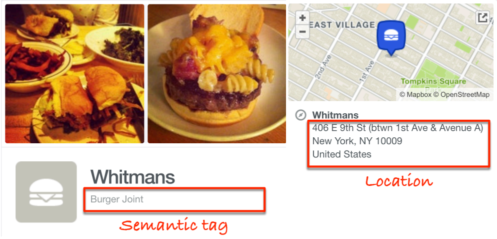

Figure 1: **A check-in to a burger joint called "Whitmans" on Foursquare.** The important information (i.e., the location and the semantic tag of the venue) is highlighted by the red bounding boxes.

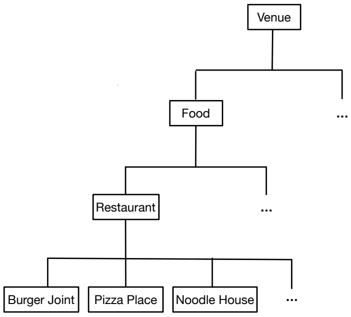

Figure 2: **Example of a semantic tag hierarchical tree.** Each node of the tree is a semantic tag and its parent is a more generalized semantic tag. The root of the tree is the most generalized semantic tag "venue". Note that because of the lack of space, the figure displays only some nodes of the tree. The rest is omitted using the three dots.

1. Introduction

In recent years, the combination of location-based services (LBSs) with online social networks has led to the emergence of location-based social networks (LBSNs) such as Foursquare [19] and Facebook [18]. In these networks, users can share with each other, their (geographical) locations together with the semantic information associated with their locations. For instance, by checking-in to venue "Whitmans" on Foursquare, a user implicitly accepts to share with her friends, the address of the venue together with its type (category), which is represented in the form of a semantic tag "burger joint" (See Fig. 1). A venue's semantic tag usually belongs to a predefined set of tags, where the set of tags form a hierarchical tree in which the "burger joint" tag could be a descendant of the "restaurant" tag and the "restaurant" tag could be a descendant of the "food" tag, and so forth [7, 1] (See Fig. 2).

It is known that by disclosing their locations in LBSNs (and in LBSs, in general), users put their *location privacy* at risk. In fact, an adversary (e.g., a curious service provider or a user of the social network who observes the disclosed locations) can use a collection of users' disclosed locations to re-identify their pseudonymous location traces or to infer their locations at given time instants [53, 54, 50]. As shown in [1], revealing semantic tags together with locations, creates a still more powerful threat to the users' location privacy. Intuitively, this is because the mo-

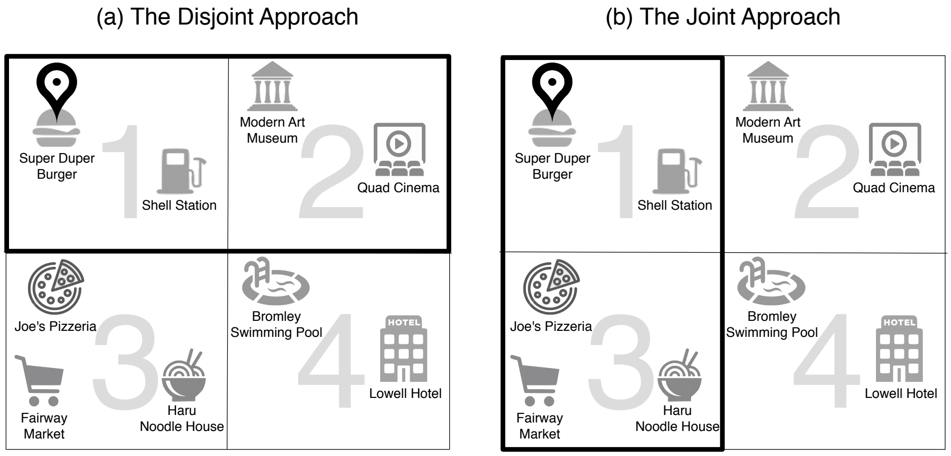

Figure 3: **Toy Examples of the obfuscation approaches.** In both subfigures, a user Alice wants to check-in to venue "Super Duper Burger" on a privacy-aware LBSN. (a) The disjoint semantic tag-location obfuscation approach replaces her location's semantic tag (i.e., "burger joint") by the generalized tag "restaurant" and her location (i.e., region 1) by the cloaking area $\{region\ 1, region\ 2\}$. (b) The joint semantic tag-location obfuscation approach replaces her location's semantic tag (i.e., "burger joint") by the generalized tag "restaurant" and her location (i.e., region 1) by the cloaking area $\{region\ 1, region\ 3\}$, which has the maximum number of semantically compatible regions (locations) with the "restaurant" tag.

bility of users have some regular semantic patterns (e.g., people usually go to the movies after dining in a restaurant), which can be learned and exploited to better track their locations [7, 1]. Moreover, the semantic tags of users' locations describe users' *semantic locations* (i.e., the type of their locations). Accordingly, by revealing the semantic tags of their locations, users also put their *semantic location privacy* at risk [1, 48].

One way to protect the privacy of users is to build *privacy-aware LBSNs* in which users only share obfuscated versions of their locations and semantic tags. Thus, when a user checks-in to a venue on a privacy-aware LBSN, the venue's name, its exact location and its semantic tag are not disclosed to anyone. Instead, an obfuscated version of the location and an obfuscated version of the semantic tag are sent to the service provider and then shared with the user's friends on the LBSN. The existing solution in the literature to build privacy-aware LBSNs consists of obfuscating the location and the semantic tag independently of each other in a so called *disjoint semantic tag-location obfuscation approach* [1]. Fig. 3.a illustrates a toy example of this approach, where a geographical area is partitioned into four square regions (locations) and each region is identified by a number. Let us assume that a user Alice wants to check-in to venue "Super Duper Burger" on a privacy-aware LBSN. Thus, in the semantic tag obfuscation process, her location's semantic tag (i.e., "burger joint") is replaced by a more general tag "restaurant". Also, in the location obfuscation process, her location (i.e., region 1) is replaced by a generalized area (also called a *cloaking area*) made of regions 1 and 2.[1,2] The problem with this approach is that an adversary (e.g., a curious service provider) who knows the semantic tags of the venues in the map can easily filter out region 2 from the cloaking area and infers that Alice is located in region 1. The reason is that region 2 is not *semantically compatible* with the "restaurant" tag i.e., it has no venue whose semantic tag is equal to the "restaurant" tag or is a descendant of the "restaurant" tag in the tag hierarchy. Moreover, since "Super Duper Burger" is the only venue in region 1 whose semantic tag is a descendant of the "restaurant" tag, the adversary infers that the semantic tag of Alice's location is "burger joint".

In this work, we introduce a *joint semantic tag-location obfuscation approach* for building privacy-aware LBSNs. This approach aims to overcome the drawbacks of the disjoint approach by performing the location obfuscation based on the result of the semantic tag obfuscation. More precisely, in the location obfuscation process, the cloaking area is defined so that it has the maximum number of semantically compatible regions with the obfuscated semantic tag among the existing potential cloaking areas. Fig. 3.b illustrates a toy example of this approach. Similar to the toy example of Fig. 3.a, in this example a user Alice wants to check-in to venue "Super Duper Burger" on a privacy-aware LBSN. Thus, in the semantic tag obfuscation process, her location's semantic tag (i.e., "burger joint") is replaced by a more general tag "restaurant". However, in the location obfuscation process, her location (i.e., region 1) is replaced by a cloaking area made of regions 1 and 3. The advantage of merging region 1 with region 3 instead of merging region 1 with region 2, is that region 3 is semantically compatible with the "restaurant" tag since it has two venues (i.e, "Joe's Pizzeria" and "Haru Noodle House") whose semantic tags (i.e., "pizza place" and "noodle house") are descendants of the "restaurant" tag in the tag hierarchy, respectively. Hence, the adversary cannot filter out the region 3 by knowing the "restaurant" tag. Thus, the resulting cloaking area has two semantically compatible regions with the "restaurant" tag, which is the maximum number of semantically compatible regions that can be achieved for the "restaurant" tag and the cloaking area size of two regions.

[1] There exist different types of obfuscation in the literature. For simplicity, in this work we consider only obfuscation by *generalization*, both for locations and semantic tags.

[2] The location obfuscation and the semantic tag obfuscation can reduce user's *perceived quality of service* (also known as *utility*) [7]. However, if the utility loss caused by obfuscation can be predicted using utility models, then the obfuscation levels can be adjusted to best match the user's preferences in terms of utility and privacy [7]. We discuss this in more detail in Section 6.

Contributions. We introduce a joint semantic tag-location obfuscation approach for protecting the location privacy and the semantic location privacy of users in LBSNs (and in LBSs, in general). Our approach aims to overcome the drawbacks of the disjoint approach by performing the location obfuscation based on the result of the semantic tag obfuscation. Thereby, in our approach, the resulting cloaking area is defined so that it has the maximum number of semantically compatible regions with the obfuscated semantic tag among the existing potential cloaking areas. We also provide a formal framework that can be used for evaluation and comparison of our approach with the disjoint approach. More precisely, we consider a privacy protection mechanism (PPM) that can be defined to use one of the joint or the disjoint obfuscation approaches. We formalize the both approaches using probability distribution functions. We consider an adversary model and we formalize the adversary's attacks, where the attacks include the semantic tag inference attack and the location inference attack. We present an implementation of the attacks based on *dynamic bayesian network* (*DBN*) models. We also provide the metrics that are used to quantify the location privacy and the semantic location privacy of users who are subjected to the attacks. Using a dataset of real-world user mobility traces collected from six different cities, we perform an experimental evaluation for comparison of the joint and the disjoint approaches in terms of location privacy and semantic location privacy. In particular, we introduce algorithms that implement the obfuscation approaches and which can be used in the evaluation. We discuss the evaluation results for different values of the obfuscation parameters and for different cities. The results show that in almost all cases (i.e., in different cities and with different obfuscation parameters) the joint approach outperforms the disjoint approach in terms of location privacy protection as well as the semantic location privacy protection. We also study how different parameters (i.e., the obfuscation parameters and the distribution of the number of venues per region in different cities) can affect the performance of the obfuscation approaches. In particular, we show how changing these parameters can improve the performance of the joint approach. Finally, we present a discussion regarding the performance of the both obfuscation approaches in terms of utility.

To the best of our knowledge, this is the first work in which the location obfuscation is performed based on the result of the semantic tag obfuscation. It is also the first work that uses the concept of semantic compatibility and performs location generalization (location cloaking) based on it. However, the most important contribution of our work is introducing *joint obfuscation* as a new type of obfuscation, in which some private attributes of a user are obfuscated based on the result of the obfuscation of some of her other private attributes. Using the example of the joint obfuscation approach presented in this paper, we show how joint obfuscation can increase privacy for all private attributes that are involved in the joint obfuscation compared to the disjoint obfuscation where private attributes are obfuscated independently of each other. Accordingly, our work can be used as a model for more advanced obfuscation schemes that jointly obfuscate a greater number of private attributes.

Note that the work presented in this paper is an extension of the primary work presented as a workshop paper in [9]. In both papers, we introduce a joint semantic tag-location obfuscation approach in which location obfuscation is performed based on the result of the semantic tag obfuscation. In both papers, we present a formal framework and an experimental evaluation to compare the performance of our joint obfuscation approach with the performance of the disjoint obfuscation approach. However, the work presented in the workshop paper is limited in many ways. The present paper addresses the issues that are not covered by the workshop paper and contains a significant amount of new material. In summary, the present paper contains the following additional contributions compared to the workshop paper[3]: (1) In the workshop paper, we show that the joint obfuscation approach outperforms the disjoint obfuscation approach only in terms of *location privacy*, whereas in the present paper, we show that the joint obfuscation approach outperforms the disjoint obfuscation approach both in terms of *location privacy* and *semantic location privacy*. This is an important contribution since it indicates that the joint obfuscation can increase privacy for all private attributes that are involved in the joint obfuscation process and not only for some of them. This also caused a change in the entire content of the present paper compared to the workshop paper. More specifically, all sections of the present paper (including the theoretical parts as well as the experimental evaluation) are re-written and extended to contain information regarding *semantic location privacy*. For instance, the adversary model in Section 2.5.2 is extended to perform also an attack against the semantic location privacy and new metrics related to semantic location privacy are added in Section 4.2 and Section 5.2. Moreover, new figures and discussions related to semantic location privacy are added to the experimental evaluation results in Section 5.2. (2) The present paper contains a new section (i.e. Section 5.2.2) which discusses the experimental evaluation results for different cities and presents new figures. In particular, this section studies how the distribution of the number of venues per region in a city can affect the performance of the obfuscation approaches in that city. (3) The present paper contains a new section (i.e., Section 6) which discusses the performance of the joint obfuscation approach in terms of the

[3] For a more detailed comparison between the present paper and the workshop paper [9], as well as an exhaustive list of the new material added to the present paper, see Appendix A.

user's perceived quality of service (also known as *utility*) and compares it with the performance of the disjoint approach. (4) The present paper contains a new section (i.e., Section 3.2) which discusses *the dynamic bayesian network (DBN) inference algorithms* in detail and recommends the inference algorithms that can be used for the adversary's attacks based on the available resources. (5) In the present paper, we added pseudo-codes for the obfuscation algorithms (i.e., Algorithm 1 and Algorithm 2) to Section 5.1.3. We also added an example walkthrough of the location obfuscation algorithm and its corresponding figure to Section 5.1.3. We believe that these additions can help the readers to get a better understanding of how these algorithms work. (6) The related work in Section 7 of the present paper is extended to include new discussions regarding the interdependent location privacy, semantic tag labelling and semantic location privacy. We also added some ideas for future work in Section 8. (7) We improved all sections of the present paper by better explaining our ideas and by adding more details compared to the workshop paper. In particular, we added 5 new figures and 2 subfigures, 12 new footnotes and a table that summarizes the notations used throughout the paper.

Road map. The remainder of the paper is organized as follows. In Section 2, we describe the system model and introduce some definitions. In particular, we present a privacy protection mechanism (PPM) that can be defined to use one of the joint or disjoint obfuscation approaches. We also present the adversary model and describe the adversary's knowledge and attacks. In Section 3, we introduce an implementation of the attacks based on dynamic bayesian networks (DBNs). In Section 4, we present the privacy metrics that are used to measure the privacy of the users who are subjected to the attacks. In Section 5, we perform an experimental evaluation to compare the joint and the disjoint approaches in terms of location privacy protection and semantic location privacy protection and we discuss the results. In Section 6, we present a discussion on the performance of the both obfuscation approaches in terms of utility. In Section 7, we discuss the related work. Finally, in Section 8, we conclude and discuss the future work. The paper has also an Appendix A, which provides a detailed discussion regarding the additional contributions of the present paper compared to the primary work published as a workshop paper in [9].

2. System Model

In this section, we present the system model. Our model is built upon the framework proposed by Shokri et al. for quantifying location privacy [53, 54, 50] and its extension for semantic location privacy by Ağir et al. [1]. Accordingly, the notations used in this work are similar to those used in [53, 54, 50, 1]. Table 1 summarizes the main notations used throughout this work.

2.1. Regions and Semantic Tags

We assume that the users move in a geographical area that is partitioned into a set $\mathcal{R}$ of M distinct regions. We use the terms *region*, *geographical location* and *location* interchangeably. Each region has a unique identifier and contains a set of venues. A venue is characterized by its type, which is represented in the form of a semantic tag. The semantic tag of a venue belongs to a set $\mathcal{S}$ of all possible semantic tags. We assume that $\mathcal{S}$ can be represented as a tree data structure where each node is a semantic tag and the parent of a given node is a more general semantic tag with respect to a specified tag hierarchy.

Below, we present some definitions and notations that capture the semantic characteristics of venues and regions.

- Let v be a venue in a region in $\mathcal{R}$ and s be a semantic tag in $\mathcal{S}$. Then, we say v is *semantically compatible* with s, if v's semantic tag is equal to s or descendant of s in the semantic tag tree.

- Let r be a region in $\mathcal{R}$ and s be a semantic tag in $\mathcal{S}$. Then, $NV_{\mathcal{S}}(r)$ denotes the number of venues in r whose semantic tags are equal to s. Also, $NDV_{\mathcal{S}}(r)$ denotes the number of venues in r whose semantic tags are descendants of s in the semantic tag tree. Finally, $NCV_{\mathcal{S}}(r)$ denotes the number of venues in r that are semantically compatible with s. Thus, $NCV_{\mathcal{S}}(r)$=$NV_{\mathcal{S}}(r)$+$NDV_{\mathcal{S}}(r)$.

- Let r be a region in $\mathcal{R}$ and s be a semantic tag in $\mathcal{S}$. Then, we say that r is *semantically compatible* with s if r contains at least one venue which is semantically compatible with s, i.e., $NCV_{\mathcal{S}}(r) > 0$.

2.2. Time

Time is discrete and the set of time instants when the users may be observed is $\mathcal{T} = \{1, ..., T\}$. The set $\mathcal{T}$ is called *the observation interval.*

2.3. Users

We assume a set $\mathcal{U}$ of N users, where each user has a unique identifier. The mobility of a user is characterized by her events and her traces. More specifically, the fact that a user u is at location r with semantic tag s at time t, can be represented by a tuple $< u, r, s, t >$. We call this tuple an *event.* Note that the semantic tag of location of u at time t refers in fact to the semantic tag of the location's venue where u is located at time t. The location trace and the

Table 1: Summary of Notations

Symbol	**Meaning**
$u, \mathcal{U}, N$	A user, Set of users, Number of users ($N = \|\mathcal{U}\|$).
t, $\mathcal{T}$, T	A time instant, Set of time instants, Number of time instants ($T = \|\mathcal{T}\|$).
r, $\mathcal{R}$, M	A location (region), Set of locations, Number of locations ($M = \|\mathcal{R}\|$).
s, $\tilde{s}$, $\mathcal{S}$	A semantic tag, A pseudo-semantic tag, Set of semantic tags that form a semantic tag tree.
$\tilde{r}$, $\widetilde{\mathcal{R}}$	A pseudo-location, Set of possible pseudo-locations.
$NV_s(.)$	Number of venues whose semantic tags are equal to s in a given location.
$NDV_s(.)$	Number of venues whose semantic tags are descendants of semantic tag s in a given location.
$NCV_s(.)$	Number of semantically compatible venues with semantic tag s in a given location.
$NCR_{\tilde{s}}(.)$	Number of semantically compatible locations (regions) with pseudo-semantic tag $\tilde{s}$ in a given cloaking area.
$SumNCV_{\tilde{s}}(.)$	Sum of $NCV_{\tilde{s}}$ values over all locations in a given cloaking area.
MNV	The median of the number of venues per region in a given city.
$< u, r, s, t >$	Actual event: user u is at location r with semantic tag s at time t.
$< u, \tilde{r}, \tilde{s}, t >$	Obfuscated event: indicates that the pseudo-location of user u at time t is $\tilde{r}$ and the pseudo-semantic tag associated to location of u at time t is $\tilde{s}$.
r_u^t, s_u^t	Location of user u at time t, Semantic tag of location of user u at time t.
R_u^t, S_u^t	Random variable associated with location of user u at time t, Random variable associated with semantic tag of location of user u at time t.
$r_u^{1:T}$, $s_u^{1:T}$	Location trace of user u: $r_u^{1:T} \triangleq \{r_u^1, ..., r_u^T\}$, Semantic tag trace of user u: $s_u^{1:T} \triangleq \{s_u^1, ..., s_u^T\}$.
$R_u^{1:T}$, $S_u^{1:T}$	$R_u^{1:T} \triangleq \{R_u^1, ..., R_u^T\}$, $S_u^{1:T} \triangleq \{S_u^1, ..., S_u^T\}$.
$\tilde{r}_u^t$, $\tilde{s}_u^t$	Pseudo-location of user u at time t, Pseudo-semantic tag of location of user u at time t.
$\widetilde{R}_u^t$, $\widetilde{S}_u^t$	Random variable associated with pseudo-location of user u at time t, Random variable associated with pseudo-semantic tag of location of user u at time t.
$\tilde{r}_u^{1:T}$, $\tilde{s}_u^{1:T}$	Obfuscated location trace of user u: $\tilde{r}_u^{1:T} \triangleq \{\tilde{r}_u^1, ..., \tilde{r}_u^T\}$, Obfuscated semantic tag trace of user u: $\tilde{s}_u^{1:T} \triangleq \{\tilde{s}_u^1, ..., \tilde{s}_u^T\}$.
$\widetilde{R}_u^{1:T}$, $\widetilde{S}_u^{1:T}$	$\widetilde{R}_u^{1:T} \triangleq \{\widetilde{R}_u^1, ..., \widetilde{R}_u^T\}$, $\widetilde{S}_u^{1:T} \triangleq \{\widetilde{S}_u^1, ..., \widetilde{S}_u^T\}$.
g, f	Location obfuscation function of PPM, Semantic tag obfuscation function of PPM.
$d_{loc}\ (.)$, $d_{sem}\ (.)$	A distance function between locations, A distance function between semantic tags.
o_{loc}, o_{sem}, λ	Location obfuscation level, Semantic tag obfuscation level, Hiding probability.
loc-priv-ratio	Ratio of the location privacy mean obtained for the joint approach to the location privacy mean obtained for the disjoint approach.
sem-priv-ratio	Ratio of the semantic location privacy mean obtained for the joint approach to the semantic location privacy mean obtained for the disjoint approach.

semantic tag trace of user u can then be obtained based on the set of her events over the entire observation interval. Thus, *the location trace* of u is defined as $r_u^{1:T} \triangleq \{r_u^1, ..., r_u^T\}$, where r_u^t with $t \in \mathcal{T}$, denotes the location of u at time t. We assume that r_u^t is an instantiation of random variable R_u^t that takes values in $\mathcal{R}$. Moreover, *the semantic tag trace* of u is defined as $s_u^{1:T} \triangleq \{s_u^1, ..., s_u^T\}$, where s_u^t with $t \in \mathcal{T}$, denotes the semantic tag of location of u at time t. We assume that s_u^t is an instantiation of random variable S_u^t that takes values in $\mathcal{S}$.

2.4. Privacy Protection Mechanism (PPM)

In order to protect their location privacy and their semantic location privacy, users rely on the privacy-protection mechanism (PPM). This mechanism obfuscates user's locations and their corresponding semantic tags before reporting them to the online service provider. More specifically, the privacy-protection mechanism (PPM) that we consider in this work transforms each actual event $< u, r, s, t >$ to an *obfuscated event* $< u, \tilde{r}, \tilde{s}, t >$, where $\tilde{r}$ and $\tilde{s}$ are the obfuscated versions of r and s, respectively.[4]

The obfuscation of r is achieved through *the location obfuscation* process of the PPM. The resulting pseudo-location $\tilde{r}$ is an instantiation of random variable $\widetilde{R}_u^t$ that takes values in set $\widetilde{\mathcal{R}}$, where $\widetilde{\mathcal{R}}$ is the power set of $\mathcal{R}$. We use the terms *pseudo-location* and *obfuscated location* interchangeably. In the literature, there exist various types of location obfuscation (see Section 7). In this work, we assume that the PPM performs a type of location obfuscation called *location generalization*. The goal of location generalization is to reduce the precision of the location that is reported to the online service provider. Accordingly, r is merged with its nearby regions to form an extended region (also called *a cloaking area* (*CA*)) that is represented by $\tilde{r}$. We also assume the existence of a parameter o_{loc} called *the location obfuscation level*. In this work, o_{loc} defines the number of regions in $\tilde{r}$. Thus, formally, $\tilde{r}$ represents a set that is composed of r and the other merged regions and has a cardinality of o_{loc}.

The obfuscation of s is achieved through *the semantic tag obfuscation* process of the PPM. The resulting pseudo-semantic tag $\tilde{s}$ is an instantiation of random variable $\widetilde{S}_u^t$ that takes values in set $\mathcal{S}$. We use the terms *pseudo-semantic tag* and *obfuscated semantic tag* interchangeably. One can consider different types of semantic tag obfuscation. In this work, we assume that the PPM performs a type of semantic tag obfuscation called *semantic tag generalization*, in which s is replaced by a more general semantic tag in the semantic tag tree. The level of

[4] For simplicity's sake, in this work we consider a PPM that only performs location and semantic tag obfuscation. However, a more advanced PPM can also perform time obfuscation and user anonymization.

generalization is defined by a parameter o_{sem} called *the semantic tag obfuscation level.* Thus, formally, $\tilde{s}$ is the ancestor of s that is o_{sem} level(s) above s in the semantic tag tree.

Based on what we have described, the location obfuscation and the semantic tag obfuscation can each be modeled by a probability distribution function. By applying these functions on a user's events over time, the PPM creates *the obfuscated traces* of the user from her actual traces. Thus, *the obfuscated location trace* of a user u is defined as $\tilde{r}_u^{1:T} \triangleq \{\tilde{r}_u^1, ..., \tilde{r}_u^T\}$, where $\tilde{r}_u^t$ with $t \in \mathcal{T}$, denotes the pseudo-location of u at time t and is an instantiation of $\widetilde{R}_u^t$. Moreover, *the obfuscated semantic tag trace* of user u is defined as $\tilde{s}_u^{1:T} \triangleq \{\tilde{s}_u^1, ..., \tilde{s}_u^T\}$, where $\tilde{s}_u^t$ with $t \in \mathcal{T}$, denotes the pseudo-semantic tag of location of u at time t and is an instantiation of $\widetilde{S}_u^t$.

The definition of the probability distribution functions associated to the obfuscation processes depends on the obfuscation approach used by the PPM. We discuss this in more detail hereafter.

2.4.1. Obfuscation Approaches

Formally, a PPM is defined as a pair $(f,\ g)$ where f and g are probability distribution functions that model the semantic tag obfuscation and the location obfuscation, respectively. The definition of these functions depends on the obfuscation approach used by the PPM. In the following, we introduce two obfuscation approaches and give the definition of the probability distribution functions for each approach.[5]

Let $e =< u, r, s, t >$ and $\tilde{e} =< u, \tilde{r}, \tilde{s}, t >$ be the actual and the obfuscated events of user u at time t, respectively. Then, there exist two obfuscation approaches for transforming e to $\tilde{e}$:

- **Disjoint semantic tag-location obfuscation approach.** In this approach, the location obfuscation and the semantic tag obfuscation are performed independently of each other. Thus, the probability distribution functions in this approach are defined as follows.

$$f_u(s, \tilde{s}) = \Pr\big(\widetilde{S}_u^t = \tilde{s} \mid S_u^t = s\big) \tag{1}$$

$$g_u(r, \tilde{r}) = \Pr\big(\widetilde{R}_u^t = \tilde{r} \mid R_u^t = r\big) \tag{2}$$

[5] Note that in this work we consider a PPM that does not change its obfuscation approach over time, i.e., if the PPM is defined to use an obfuscation approach it always uses that approach.

where the semantic obfuscation function f maps the semantic tag of location of u at time t to random variable $\widetilde{S}_u^t$ that takes values in $\mathcal{S}$ and the location obfuscation function g maps the location of u at time t to random variable $\widetilde{R}_u^t$ that takes values in $\widetilde{\mathcal{R}}$.

- **Joint semantic tag-location obfuscation approach.** In this approach, the location obfuscation is performed based on the result of the semantic tag obfuscation. Thus, first $\tilde{s}$ is obtained from s by applying the semantic tag obfuscation process. Then, in the location obfuscation process, the merging of r with nearby locations is performed in a way that the resulting $\tilde{r}$ has the maximum number of semantically compatible regions with $\tilde{s}$. Formally this can be expressed as follows. Let $\mathcal{C}(r)$ be the set of potential cloaking areas for region r and $NCR_{\tilde{S}}(.)$ denote the number of regions that are semantically compatible with $\tilde{s}$ in a given cloaking area. Then, an element $\tilde{r}$ of $\mathcal{C}(r)$ has the maximum number of semantically compatible regions with semantic tag $\tilde{s}$ if $NCR_{\tilde{S}}(\tilde{r}) \geq NCR_{\tilde{S}}(\tilde{\rho})$ for $\forall \tilde{\rho} \in \mathcal{C}(r)$. As we have already discussed, the knowledge of $\tilde{s}$ can help the adversary to filter out the regions that are not semantically compatible with $\tilde{s}$. Thus, by defining $\tilde{r}$ as a cloaking area that has the maximum semantically compatible regions with $\tilde{s}$, we aim to reduce the negative impact that the revelation of $\tilde{s}$ can have on users' location privacy and semantic location privacy. Based on what we have described, the probability distribution functions in this approach are defined as follows.

$$f_u(s, \tilde{s}) = \Pr\big(\widetilde{S}_u^t = \tilde{s} \mid S_u^t = s\big) \tag{3}$$

$$g_u(r, \tilde{r}, \tilde{s}) = \Pr\big(\widetilde{R}_u^t = \tilde{r} \mid R_u^t = r, \widetilde{S}_u^t = \tilde{s}\big) \tag{4}$$

where the semantic tag obfuscation function f maps the semantic tag of location of u at time t to random variable $\widetilde{S}_u^t$ that takes values in $\mathcal{S}$ and the location obfuscation function g maps both the location and the pseudo-semantic tag of location of u at time t to random variable $\widetilde{R}_u^t$ that takes values in $\widetilde{\mathcal{R}}$.

2.5. Adversary

Typically, the adversary is a curious service provider (or an external observer with the access to the same information), who observes the obfuscated traces of the users and seeks to infer the locations of users and their corresponding semantic tags at given time instants. Formally, we model the adversary by his *knowledge* and his *attacks*.

2.5.1. Knowledge

The adversary has full knowledge of regions (including their venues and their semantic tags) and the semantic tag tree. He knows which obfuscation approach is used by the PPM and also knows the semantic tag obfuscation function (f) and the location obfuscation function (g) of PPM in both obfuscation approaches. We assume that the adversary performs his attacks *a posteriori*, meaning that the adversary has access to the obfuscated traces of the users over the complete observation interval. In addition, he has access to some of the past semantic tag traces and past location traces of the users. We refer to this as his *prior information.* As we further discuss in Section 3.1.1, the adversary uses the prior information to build a basic dynamic bayesian network (DBN) for each user, where the basic DBN of a user models her mobility.

2.5.2. Attacks

The adversary performs the two following attacks, where each attack is defined as a statistical inference problem. We present an implementation of these attacks in Section 3.

- **Location-Inference Attack.** In this attack, the goal of the adversary is to find the location of a user u at time t, given the obfuscated location trace and the obfuscated semantic tag trace of u. The attack can be formalized as finding the following posterior probability distribution over set $\mathcal{R}$ of regions:

$$\Pr\big(R_u^t = r \mid \tilde{r}_u^{1:T}, \tilde{s}_u^{1:T}\big) \tag{5}$$

- **Semantic tag-Inference Attack.** In this attack, the goal of the adversary is to find the semantic tag associated with the location of a user u at time t, given the obfuscated location trace and the obfuscated semantic tag trace of u. The attack can be formalized as finding the following posterior probability distribution over set $\mathcal{S}$ of semantic tags:

$$\Pr\big(S_u^t = s \mid \tilde{r}_u^{1:T}, \tilde{s}_u^{1:T}\big) \tag{6}$$

3. Implementation of the Attacks

The adversary can use different methods to implement his attacks. In this work, we assume an implementation which is based on *dynamic bayesian networks* (*DBNs*). More precisely, we assume that to implement the attacks, the adversary first builds a *dynamic bayesian network* (*DBN*) model for each user based on his knowledge. Roughly speaking, the DBN model for a user encodes the probabilistic dependencies between the random variables involved in the inference attacks against that user. Once a DBN is built for a user, the adversary can perform his attacks against the user by applying the existing DBN inference algorithms. In the following, we first discuss the DBN models. We then discuss the DBN inference algorithms that can be used by the adversary.

3.1. The Dynamic Bayesian Network (DBN) Models

Based on his knowledge, the adversary builds a *dynamic bayesian network* (*DBN*) model for each user. A DBN is a probabilistic graphical model. It belongs to a wider class of probabilistic graphical models known as *bayesian networks* (*BNs*). In fact, a DBN is a BN which is used to model time series, sequential data [42, 29].

The DBN model of a user u built by the adversary, presents a joint distribution over random variables $R_u^{1:T}, S_u^{1:T}, \widetilde{R}_u^{1:T}, \widetilde{S}_u^{1:T}$, where $R_u^{1:T} \triangleq \{R_u^1, ..., R_u^T\}$, $S_u^{1:T} \triangleq \{S_u^1, ..., S_u^T\}$, $\widetilde{R}_u^{1:T} \triangleq \{\widetilde{R}_u^1, ..., \widetilde{R}_u^T\}$ and $\widetilde{S}_u^{1:T} \triangleq \{\widetilde{S}_u^1, ..., \widetilde{S}_u^T\}$. These random variables can be divided into two categories: (1) *Observed variables.* These are the variables that are directly observed and whose values are known by the adversary. They include $\widetilde{R}_u^{1:T}$ and $\widetilde{S}_u^{1:T}$; (2) *Unobserved variables* (also called *hidden variables*). These are the variables that are not directly observed and whose values are supposed to be inferred from the observed variables. They include $R_u^{1:T}$ and $S_u^{1:T}$. The graphical structure of the DBN specifies all probabilistic dependencies between the hidden variables, between the hidden and the observed variables and between the observed variables.[6]

The probabilistic dependencies between the hidden and the observed variables as well as between the observed variables themselves, depend on the obfuscation approach used by the PPM. Accordingly, the DBN of a user in the case where the disjoint obfuscation approach is used differs from her DBN in the case where the joint obfuscation approach is used. However, in both cases the probabilistic dependencies between the hidden variables remain the same. Thus, in the following we first present a basic DBN for user u that encodes only the probabilistic dependencies between the hidden variables. Then, we present the DBNs of u for the

[6] Note that the adversary can as well use a *hidden Markov model* (*HMM*) to model the data. In fact, DBNs generalize HMMs. The main difference between DBNs and HMMs is that in DBNs the hidden state can be represented by any number of random variables whereas in HMMs it can only be represented by one random variable. This means that in a HMM model, the adversary needs to represent the couple (location, location's semantic tag) by one random variable. However, as discussed in [42, 39, 1], this simplification makes the inferences more complex and less efficient.

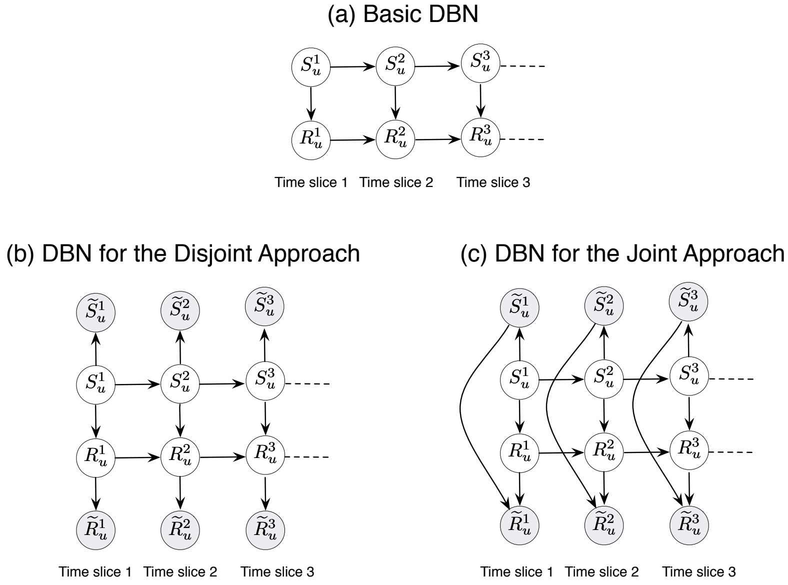

Figure 4: The Dynamic Bayesian Network (DBN) Models.

disjoint and the joint obfuscation cases. These DBN models are made by adding the corresponding observed variables of each case to the basic DBN.

3.1.1. The Basic DBN

This model encodes the probabilistic dependencies between the hidden variables associated to user u, namely $R_u^{1:T}, S_u^{1:T}$ (see Fig. 4.a). Since these random variables characterize the mobility of user u, we can say that the basic DBN models the mobility of u. The adversary builds this model based on the following assumption on user mobility: to move to the next location, a user first decides on the type (i.e., semantic tag) of the next location based on the type (i.e., semantic tag) of her current location. Once the next location type is decided, the user can choose her next (geographical) location based on her current (geographical) location and the next location type. For instance, a user is in a restaurant and decides to go to the movies, as she usually does after going to a restaurant. Thus, considering her current geographical location, she chooses the movie theater that is most convenient to her (e.g., the closest movie theater to the restaurant) [1].

Let us take a closer look at the model. Since a DBN is a type of bayesian network (BN), the model exhibits the general properties of BNs. More precisely, it is a directed acyclic graph in which nodes represent random variables and the edges model conditional dependencies between variables. In addition, each node has a conditional probability distribution (CPD) associated to it, which is the CPD of the variable represented by the node, given the parent variables of the node (by parent variables of a node, we mean the variables that are represented by the parent nodes of that node in the graph). For instance, in each time slice t of Fig. 4.a, to represent the fact that S_u^t depends on S_u^{t-1}, an edge connects the corresponding nodes and the associated CPD is $\Pr\big(S_u^t \bigm| S_u^{t-1}\big)$. Moreover, to represent the fact that R_u^t depends on S_u^t and R_u^{t-1}, two edges connect the corresponding nodes and the associated CPD is $\Pr\big(R_u^t \bigm| S_u^t, R_u^{t-1}\big)$. The model has also some properties that are specific to DBNs. Firstly, it has a structure which is repeated over time. Secondly, the model is *first order Markovian*, i.e., the random variables in each time slice t are independent of all random variables from time slices 1 to $t-2$, given the random variables in time slice $t-1$. Finally, the model is *time-invariant* (also called *stationary* or *homogenous*), i.e., the CPDs of the model do not change as a function of time. As a consequence of the Markov and the time-invariance properties of the model, $R_u^{1:T}$ and $S_u^{1:T}$ each form a time-invariant first order Markov chain.

Parameters. The model is fully specified by the following probability distributions.

- **The transition distributions: $\boldsymbol{\Pr\big(S_u^t \bigm| S_u^{t-1}\big)}$ and $\boldsymbol{\Pr\big(R_u^t \bigm| S_u^t, R_u^{t-1}\big)}$.** These are the CPDs that define the transition between any two consecutive time slices $t-1$ and t in the model. According to [1], the distribution $\Pr\big(R_u^t \bigm| S_u^t, R_u^{t-1}\big)$ can be computed as follows:

$$\Pr\big(R_u^t = r \bigm| S_u^t = s, R_u^{t-1} = r'\big) = \begin{cases} 0, & \text{if } NV_S(r) = 0 \\ \alpha \dfrac{\Pr\big(R_u^t = r \bigm| R_u^{t-1} = r'\big)}{\sum\limits_{\rho \in \mathcal{E}} \Pr\big(R_u^t = \rho \bigm| R_u^{t-1} = r'\big)} & \text{otherwise} \\ +(1-\alpha) \cdot \Pr\big(R_u^t = r \bigm| S_u^t = s\big), & \end{cases} \tag{7}$$

  where $\mathcal{E} = \{\rho \in \mathcal{R} : NV_S(\rho) > 0\}$ and α is a real-valued parameter that is used to set the weight of each term in the equation. The distributions $\Pr\big(S_u^t \bigm| S_u^{t-1}\big)$ and $\Pr\big(R_u^t \bigm| R_u^{t-1}\big)$ can be learned from the prior traces by applying maximum likelihood estimation (if the traces are complete) or by using algorithms such as Gibbs sampling (if the traces have missing locations or if they are noisy) [53, 50, 8]. The distribution $\Pr\big(R_u^t \bigm| S_u^t\big)$ can also be learned from the prior traces. More precisely, $\Pr\big(R_u^t = r \bigm| S_u^t = s\big)$ can be estimated by counting in the user's prior traces, the number of visits to a region r given the semantic tag s [1]. Note that in the experimental evaluation in [1], Ağir et al. set $\alpha = 0.5$ to accord the same importance to both terms of the equation. In this paper, we follow the same convention and set $\alpha = 0.5$ for the experimental evaluation.

- **The initial state distributions: $\mathbf{Pr}\big(\boldsymbol{R_u^1} \mid \boldsymbol{S_u^1}\big)$ and $\mathbf{Pr}\big(\boldsymbol{S_u^1}\big)$.** These are the distributions associated to the nodes of the first time slice of the model. For the estimation of $\Pr\big(R_u^1 \mid S_u^1\big)$, we refer the reader to the previous point, where we discuss the estimation of $\Pr\big(R_u^t \mid S_u^t\big)$ from the prior traces (recall that the model is time-invariant). Moreover, we assume that $\Pr\big(S_u^1\big)$ is equal to the stationary distribution of the Markov chain $S_u^{1:T}$. Accordingly, it can be found based on $\Pr\big(S_u^t \mid S_u^{t-1}\big)$, which is the transition distribution of the chain. We refer the reader to the previous point where we discuss the estimation of $\Pr\big(S_u^t \mid S_u^{t-1}\big)$ from the prior traces.

3.1.2. The DBN for the Disjoint Obfuscation Case

This is the DBN built for user u in the case where the PPM uses the disjoint obfuscation approach. It is made by adding the observed variables $\widetilde{R}_u^{1:T}$ and $\widetilde{S}_u^{1:T}$ to the basic DBN, where the observed variables correspond to the disjoint obfuscation case (See Fig. 4.b where observed variables are indicated in gray).

Parameters. The model is fully specified by the parameters of the basic DBN plus the following CPDs.

- **The observation distributions: $\mathbf{Pr}\big(\widetilde{\boldsymbol{S}}_{\boldsymbol{u}}^{\boldsymbol{t}} \mid \boldsymbol{S_u^t}\big)$ and $\mathbf{Pr}\big(\widetilde{\boldsymbol{R}}_{\boldsymbol{u}}^{\boldsymbol{t}} \mid \boldsymbol{R_u^t}\big)$.** These are the CPDs that define the probabilistic dependencies between the hidden and the observed variables in any time slice t in the model. These CPDs are in fact the obfuscation functions of the PPM in the disjoint obfuscation approach (see Eqs. 1 and 2), and hence known by the adversary.

3.1.3. The DBN for the Joint Obfuscation Case

This is the DBN built for user u in the case where the PPM uses the joint obfuscation approach. It is made by adding the observed variables $\widetilde{R}_u^{1:T}$ and $\widetilde{S}_u^{1:T}$ to the basic DBN, where the observed variables correspond to the joint obfuscation case (See Fig. 4.c where observed variables are indicated in gray). In particular, to represent the fact that in the joint obfuscation case $\widetilde{R}_u^t$ depends also on $\widetilde{S}_u^t$, an edge connects the corresponding nodes in each time slice t of the model.

Parameters. The model is fully specified by the parameters of the basic DBN plus the following CPDs.

- **The observation distributions: $\mathbf{Pr}(\widetilde{\boldsymbol{S}}_{\boldsymbol{u}}^{\boldsymbol{t}} \mid \boldsymbol{S}_{\boldsymbol{u}}^{\boldsymbol{t}})$ and $\mathbf{Pr}(\widetilde{\boldsymbol{R}}_{\boldsymbol{u}}^{\boldsymbol{t}} \mid \boldsymbol{R}_{\boldsymbol{u}}^{\boldsymbol{t}}, \widetilde{\boldsymbol{S}}_{\boldsymbol{u}}^{\boldsymbol{t}})$.** These are the CPDs that define the probabilistic dependencies between the hidden and the observed variables in any time slice t in the model. These CPDs are in fact the obfuscation functions of the PPM in the joint obfuscation approach (see Eqs. 3 and 4), and hence known by the adversary.

3.2. The DBN Inference Algorithms

The adversary performs his attacks against a user by applying an existing bayesian network inference algorithm to the DBN of that user. One of the widely-used inference algorithms for bayesian networks is the *belief propagation* (*BP*) algorithm [45]. The belief propagation (BP) algorithm is an iterative message passing algorithm. Roughly speaking, this means that at each iteration, nodes perform a set of local computations and then relay the results to their neighbors in the form of so called messages [44]. One of the main advantages of the BP algorithm is that it computes all the (conditional) marginal distributions simultaneously. Thus, one can run the algorithm once for multiple inference queries instead of re-running it for each single query. However, the BP algorithm has a limitation: it computes the exact inference solution only for graphs with no undirected cycles (loops). Recall that the DBN that is built for the joint obfuscation case has loops (See Fig. 4.c). Thereby, the BP algorithm can not directly be used by the adversary for the exact inference.

There exist, however, two algorithms in the literature, that are based on the BP algorithm and can be applied on general graphs (i.e., with or without loops). These algorithms are *the junction tree algorithm* and *the loopy belief propagation algorithm* . The junction tree algorithm (also known as *the clique tree algorithm*) [29, 42], finds the exact inference solution but has a high space and time complexity. The loopy belief propagation has a less inference cost compared to the junction tree but it finds the approximate inference solution [45, 43]. The adversary can choose one of these two algorithms for performing the inference attacks based on the resources that he has at his disposal. If the inference cost is not very important to him, he can apply the junction tree algorithm. However, if he has limited resources, he can apply the loopy belief propagation algorithm which provides a good approximation of the exact inference solutions [43, 42].

4. Privacy Metrics

In this section, we present the metrics to measure the location privacy and the semantic location privacy of a user with respect to the inference attacks.

4.1. Location Privacy Metric

The location privacy of a user u a time t is measured by the expected error of the adversary when performing the location-inference attack [53]. The expected error of the adversary is computed as:

$$\sum_{r \in \mathcal{R}} \Pr\big(R_u^t = r \mid \tilde{r}_u^{1:T}, \tilde{s}_u^{1:T}\big) \cdot d_{loc}(r, r_u^t) \tag{8}$$

where $\Pr\big(R_u^t = r \mid \tilde{r}_u^{1:T}, \tilde{s}_u^{1:T}\big)$ over set $\mathcal{R}$, is the output of the location-inference attack defined in Section 2.5.2 and $d_{loc}(\cdot,\cdot)$ denotes a distance function on the set $\mathcal{R}$ of regions. Here, we assume that $d_{loc}(\cdot,\cdot)$ is *the Haversine distance* between the centers of the two regions [39].

4.2. Semantic Location Privacy Metric

The semantic location privacy of a user u a time t is measured by the expected error of the adversary when performing the semantic tag-inference attack [53, 1]. The expected error of the adversary is computed as:

$$\sum_{s \in \mathcal{S}} \Pr\big(S_u^t = s \mid \tilde{r}_u^{1:T}, \tilde{s}_u^{1:T}\big) \cdot d_{sem}(s, s_u^t) \tag{9}$$

where $\Pr\big(S_u^t = s \mid \tilde{r}_u^{1:T}, \tilde{s}_u^{1:T}\big)$ over set $\mathcal{S}$, is the output of the semantic tag-inference attack defined in Section 2.5.2 and $d_{sem}(\cdot,\cdot)$ denotes a distance function on the set $\mathcal{S}$ of semantic tags. Given two semantic tags s and s', we use the method introduced in [1] to compute $d_{sem}(s,\, s')$ as below:

$$d_{sem}(s, s') = \frac{d_{graph}(s, s')}{d_{graph}(s, root) + d_{graph}(s', root)} \tag{10}$$

where $d_{graph}(\cdot,\cdot)$ denotes *the graph distance* between the two given semantic tags (nodes) in the semantic tag tree and is measured by the number of edges in the shortest path connecting them. Moreover, *root* denotes the root of the semantic tag tree.

5. Experimental Evaluation

Using a dataset of real-world user mobility traces, we perform an experimental evaluation to compare the performance of the joint obfuscation approach with the performance of the disjoint approach in terms of location privacy and semantic location privacy. We also study how different parameters can affect the performance of the obfuscation approaches. More precisely, we first obfuscate the traces of the users under the disjoint and the joint approaches using different combinations of the obfuscation parameters. Then, we perform the inference attacks on the obfuscated traces and measure the privacy of the users in both approaches based on the results of the attacks. Finally, we compare the privacy results obtained for each approach.

5.1. Evaluation Setup

In this section, we describe the experimental evaluation's setup.

5.1.1. Dataset

We use the dataset that is collected between January 2015 and July 2015 by Ağir et al. for the research purposes and used for evaluation of the disjoint obfuscation approach in [1]. It comprises the semantically-annotated location traces of Foursquare check-ins (collected through Twitter's public stream) of total of 1065 users distributed across six large cities in North America and Europe, namely Boston, Chicago, Istanbul, London, New York and San Francisco. The location information in the traces is presented as GPS coordinates. The dataset also contains a snapshot of Foursquare category tree at the time of data collection.

5.1.2. Space Discretization

We use the same space discretization described in [1]. More precisely, within each city in the dataset, a geographical area of size $\sim$ 2.4 km $\times$ 1.6 km that contains the largest number of check-ins is selected. Then, each selected area is partitioned into 96 locations (cells) by using a 12 $\times$ 8 regular square grid. Each grid cell has a unique ID. Once the partitioning is done, the GPS coordinates in user traces are translated into the location (i.e., the grid cell) they fall into. Moreover, for each grid cell, the Foursquare semantic tags of the venues that are located in that cell are identified and stored in an associative array. Thus, the associative array contains the key-value pairs, where in each pair the key is a grid cell ID and the value is the set of the semantic tags of the venues located in that cell. The associative array can be used by the location obfuscation algorithm (i.e., Algorithm 2 described in Section 5.1.3) in the case of joint obfuscation.

5.1.3. Obfuscation

In the following, we first introduce the pseudocode of the algorithms that we use for the semantic tag obfuscation and the location obfuscation in our evaluation. We then describe the process of building the obfuscated traces from the real traces using these algorithms. Note that for our experimental evaluation, the algorithms presented hereafter are implemented in Python.

Algorithm 1: Semantic Tag Obfuscation Algorithm

```
input  : a set $\mathcal{S}$ of semantic tags that form a semantic tag tree, a semantic tag $s$ in $\mathcal{S}$, a semantic tag obfuscation level $o_{sem}$
output: a pseudo-semantic tag $\tilde{s}$ for $s$
begin
    $\tilde{s} \leftarrow$ getAncestor($\mathcal{S}$, $s$, $o_{sem}$)
    return $\tilde{s}$
```

Algorithm 2: Location Obfuscation Algorithm

```
input  : a grid mainGrid, a cell (location) $r$ of mainGrid, a location obfuscation level $o_{loc}$, an obfuscation approach
         approach, the semantic tag tree $\mathcal{S}$ that is used as input by Algorithm 1, the pseudo-semantic tag $\tilde{s}$ that is output
         by Algorithm 1, an associative array SemanticTagsTable that contains key-value pairs where in each pair the key is
         a main grid cell ID and the value is the set of the semantic tags of the venues located in that cell ($\mathcal{S}$, $\tilde{s}$ and
         SemanticTagsTable should only be input if a joint obfuscation approach is used).
output: a cloaking area $\tilde{r}$
begin
    potentialCloakingAreas $\leftarrow$ getPotentialCloakingAreas(mainGrid, $r$, $o_{loc}$)
    if approach=disjoint then
        $\tilde{r} \leftarrow$ selectCloackingAreaForDisjointApproach(potentialCloakingAreas)
    if approach=joint then
        $\tilde{r} \leftarrow$ selectCloackingAreaForJointApproach(potentialCloakingAreas, $\mathcal{S}$, $\tilde{s}$, SemanticTagsTable)
    return $\tilde{r}$
function getPotentialCloakingAreas(mainGrid, $r$, $o_{loc}$)
    cloakingGrids $\leftarrow$ getCloakingGrids(mainGrid, $o_{loc}$)
    potentialCloakingAreas $\leftarrow \{\}$
    for cloakingGrid in cloakingGrids do
        potentialCloakingAreas $\leftarrow$ potentialCloakingAreas $\cup$ getCloackingGridCell(cloakingGrid, $r$)
    return potentialCloakingAreas
function selectCloackingAreaForDisjointApproach(potentialCloakingAreas)
    selectedCloackingArea $\leftarrow$ selectAreaRandomly(potentialCloakingAreas)
    return selectedCloackingArea
function selectCloackingAreaForJointApproach(potentialCloakingAreas, $\mathcal{S}$, $\tilde{s}$, SemanticTagsTable)
    AreasWithMaxNCR $\leftarrow$ selectAreasWithMaxNCR(potentialCloakingAreas, $\mathcal{S}$, $\tilde{s}$, SemanticTagsTable)
    if |AreasWithMaxNCR| = 1 then
        selectedCloackingArea $\leftarrow$ AreasWithMaxNCR
    else
        AreasWithMaxSumNCV $\leftarrow$ selectAreasWithMaxSumNCV(AreasWithMaxNCR, $\mathcal{S}$, $\tilde{s}$, SemanticTagsTable)
        if $|AreasWithMaxSumNCV| = 1$ then
            selectedCloackingArea $\leftarrow$ AreasWithMaxSumNCV
        else
            selectedCloackingArea $\leftarrow$ selectAreaRandomly(AreasWithMaxSumNCV)
    return selectedCloackingArea
```

Semantic Tag Obfuscation Algorithm. The semantic tag obfuscation in both disjoint and joint obfuscation approaches is performed by Algorithm 1. The algorithm gets as input a set $\mathcal{S}$ of semantic tags that form a semantic tag tree, a semantic tag s in $\mathcal{S}$ and a semantic tag obfuscation level o_{sem}. It returns as output a pseudo-semantic tag $\tilde{s}$, where $\tilde{s}$ is the ancestor of s that is o_{sem} level(s) above s in the semantic tag tree (see the function GETANCESTOR in line 2). To better understand how the algorithm works, consider the following example. Assume that the "burger joint" tag is given as input to the algorithm. Assume also that the "burger joint" tag is a child and a grandchild of the "restaurant" and the "food" tags in the semantic tag tree, respectively. Then, the algorithm returns as output the "restaurant" tag if $o_{sem} = 1$ and it returns the "food" tag if $o_{sem} = 2$. In the case where the depth of semantic tag s in the semantic tag tree is smaller than o_{sem}, the function GETANCESTOR returns the root of the the semantic tag tree as $\tilde{s}$. Note that in our evaluation, we use the Foursquare category tree (which is included in the dataset) as the input $\mathcal{S}$ of the algorithm.

Location Obfuscation Algorithm. The location obfuscation in both disjoint and joint obfuscation approaches is performed by Algorithm 2. Note that this algorithm is one of the various possibilities for implementing the location obfuscation. The algorithm takes as input a grid (that we call the main grid for the sake of precision and which is denoted by *mainGrid*), a cell r of the main grid, a location obfuscation level o_{loc} and an obfuscation approach (denoted by *approach*). In the case of joint obfuscation, in addition to what has been described, the following inputs should also be provided: the semantic tag tree $\mathcal{S}$ that is used as input by Algorithm 1, the pseudo-semantic tag $\tilde{s}$ that is output by Algorithm 1 and an associative array *SemanticTagsTable* that contains key-value pairs where in each pair the key is a main grid cell ID and the value is the set of the semantic tags of the venues located in that cell. The algorithm returns as output a cloaking area $\tilde{r}$ for r. Note that, in our evaluation, we perform the location obfuscation for user traces of all 6 cities in the dataset. Thus, for each city, the *mainGrid* and the *SemanticTagsTable* which are input by the algorithm, are the ones obtained from the space discretization process described in Section 5.1.2.

The main idea behind the algorithm is to first find a set of potential cloaking areas for r (denoted by *potentialCloakingAreas*) and then based on the obfuscation approach, select an area among the potential cloaking areas and return it as $\tilde{r}$ (lines 2-7).

The algorithm finds the potential cloaking areas by building a set of cloaking grids (denoted by *cloakingGrids*) (line 9). A cloaking grid is an alternative tessellation for the same surface presented by the main grid. It has two properties:

(1) each cell of a cloaking grid is made of o_{loc} distinct cells of the main grid; (2) the number of rows and the number of columns of a cloaking grid are factors of the number of rows and the number of columns of the main grid, respectively. Recall that in our evaluations, the main grid has square cells. Accordingly, in our evaluations, a cloaking grid can have either square or rectangular cells where each cell covers o_{loc} distinct cells of the main grid. Each cloaking grid can be used to find a potential cloaking area for r. More precisely, the cell of a cloaking grid that contains r, is a potential cloaking area for r and can be added to the set of potential cloaking areas (lines 10-13).

Once the potential cloaking areas are found, an area among them is selected and returned as $\tilde{r}$. The selection is made based on the obfuscation approach. More precisely, in the case of the disjoint obfuscation, the algorithm calls the function selectCloackingAreaForDisjointApproach (line 4), which selects an area uniformly at random among the potential cloaking areas by calling the function selectAreaRandomly (line 15). In the case of the joint obfuscation, the function selectCloackingAreaForJointApproach (line 6) is called. This function first calls the function selectAreasWithMaxNCR which looks for the areas with the maximum $NCR_{\tilde{\mathcal{S}}}$ value among the potential cloaking areas (line 18). The results are then stored in the set *AreasWithMaxNCR*. If only one area with the maximum $NCR_{\tilde{\mathcal{S}}}$ value is found (i.e., $|AreasWithMaxNCR| = 1$), it is returned as the selected cloaking area (line 20). Otherwise, the function selectAreasWithMaxSumNCV is called which looks for the areas with the maximum $SumNCV_{\tilde{\mathcal{S}}}$ value among the elements of *AreasWithMaxNCR* (line 22). The results are then stored in the set *AreasWithMaxSumNCV*. Note that the $SumNCV_{\tilde{\mathcal{S}}}$ of an area is in fact the sum of $NCV_{\tilde{\mathcal{S}}}$ values over all the main grid cells in that area. If only one area with the maximum $SumNCV_{\tilde{\mathcal{S}}}$ value is found (i.e., $|AreasWithMaxSumNCV| = 1$), it is returned as the selected cloaking area (line 24). Otherwise, an area is selected uniformly at random among the elements of *AreasWithMaxSumNCV* by calling the function selectAreaRandomly and the result is returned as the selected cloaking area (line 26).[7]

As described, the function selectCloackingAreaForJointApproach does not only select a cloaking area with the maximum $NCR_{\tilde{\mathcal{S}}}$ value among the potential cloaking areas. But, in the case that more than one potential cloaking area with the maximum $NCR_{\tilde{\mathcal{S}}}$ value exist, this function selects the area that has the maximum $SumNCV_{\tilde{\mathcal{S}}}$ value among the areas with the maximum $NCR_{\tilde{\mathcal{S}}}$ value. Selecting

[7] Note that the inputs $\mathcal{S}$, $\tilde{s}$ and *SemanticTagsTable* are provided to the functions selectAreasWithMaxNCR and selectAreasWithMaxSumNCV so that they can be used by these functions for the calculation of $NCR_{\tilde{\mathcal{S}}}$ and $SumNCV_{\tilde{\mathcal{S}}}$ of the cloaking areas.

the area with the maximum $SumNCV_{\tilde{s}}$ value is an additional mechanism that we use to enhance the resistance of the joint obfuscation against the privacy attacks. Roughly speaking, by selecting the cloaking area with the maximum $NCR_{\tilde{s}}$ value, we decrease the number of locations that can be filtered out by the adversary from the cloaking area and by selecting the area with the maximum $SumNCV_{\tilde{s}}$ value (i.e., the area with the maximum number of semantically compatible venues), we potentially increase the number of locations and semantic tags that can be guessed by the adversary as the actual location and semantic tag.

Fig. 5, illustrates an example walkthrough of Algorithm 2. The main grid is a 4×4 grid.[8] The number at the bottom-right of a cell represents its identifier and the number in the black square at the top-left of each cell represents its $NCV_{\tilde{s}}$, where $\tilde{s}$ is the pseudo-semantic tag returned by Algorithm 1. The $NCV_{\tilde{s}}$ of cells will be used in our example for explaining the joint obfuscation approach and comparing its output with the output of the disjoint approach in terms of semantic compatibility.[9] We assume that the cell to be obfuscated is cell 6 and $o_{loc} = 4$ (see Fig. 5.a).

The algorithm first creates a set of 4×1, 2×2 and 1×4 cloaking grids. These cloaking grids are shown in Fig. 5.b from top to bottom, respectively. Note that since o_{loc} is equal to 4, each cell of these cloaking grids contains 4 distinct cells of the main grid. Moreover, since the main grid is 4×4, the number of rows and the number of columns of all these cloaking grids are the factors of 4. Once the cloaking grids are built, the algorithm finds the potential cloaking areas. More precisely, in each cloaking grid the cell that contains the cell 6 of the main grid is identified as a potential cloaking area (see Fig. 5.c where potential cloaking areas are represented in gray). Thus, the resulting set of potential cloaking areas includes three areas: the area made of the main grid cells $\{2, 6, 10, 14\}$ with $NCR_{\tilde{s}} = 3$ and $SumNCV_{\tilde{s}} = 20$, the area made of the main grid cells $\{1, 2, 5, 6\}$ with $NCR_{\tilde{s}} = 4$ and $SumNCV_{\tilde{s}} = 18$ and the area made of the main grid cells $\{5, 6, 7, 8\}$ with $NCR_{\tilde{s}} = 4$ and $SumNCV_{\tilde{s}} = 12$.

If a disjoint obfuscation approach is used, the algorithm selects uniformly at random an area among the potential cloaking areas. Let us assume that the algorithm selects the area $\{2, 6, 10, 14\}$ and returns it as $\tilde{r}$ (see Fig. 5.d). By verifying $NCR_{\tilde{s}}$ of this area, we realize that it has the minimum value of $NCR_{\tilde{s}}$ (i.e., $NCR_{\tilde{s}} = 3$) among the potential cloaking areas. However, if a joint obfuscation approach is used, the algorithm selects the area $\{1, 2, 5, 6\}$ and returns it as $\tilde{r}$ (see

[8] In this paper, a grid is defined by *the number of its columns* × *the number of its rows.*

[9] Recall that $NCR_{\tilde{s}}$ and $SumNCV_{\tilde{s}}$ can both be calculated based on the value of $NCV_{\tilde{s}}$. In fact, $NCR_{\tilde{s}}$ is equal to the number of cells that are semantically compatible with $\tilde{s}$ i.e., the cells that have $NCV_{\tilde{s}} > 0$. Also, $SumNCV_{\tilde{s}}$ is the sum of $NCV_{\tilde{s}}$ over all cells in the cloaking area.

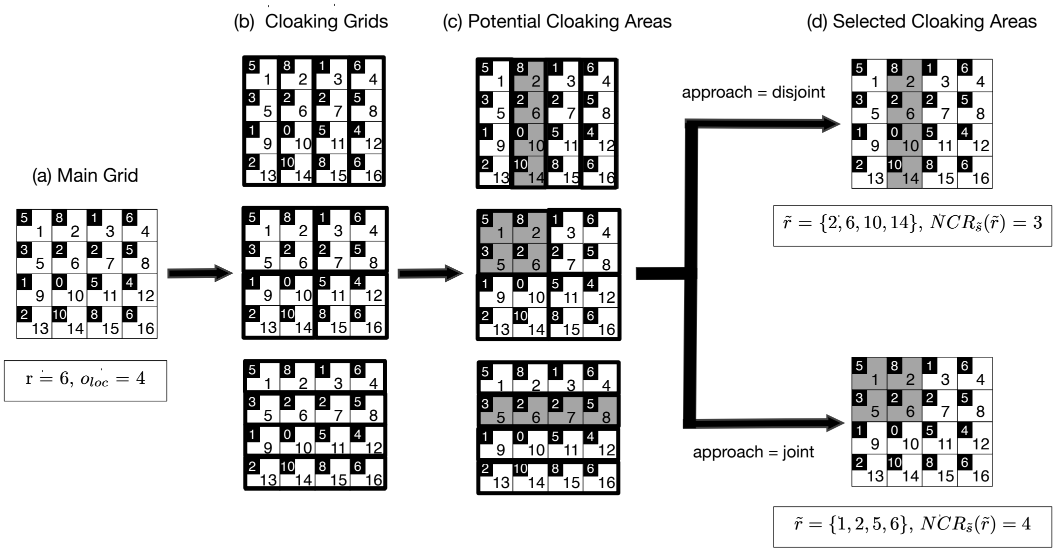
(a) Main Grid
(b) Cloaking Grids
(c) Potential Cloaking Areas
(d) Selected Cloaking Areas
r = 6, o_{loc} = 4
approach = disjoint
approach = joint
$\tilde{r} = \{2, 6, 10, 14\}$, $NCR_{\tilde{s}}(\tilde{r}) = 3$
$\tilde{r} = \{1, 2, 5, 6\}$, $NCR_{\tilde{s}}(\tilde{r}) = 4$

Figure 5: **Example walkthrough of the location obfuscation algorithm (Algorithm 2).** In the figure, the cell to be obfuscated is cell 6 and the cloaking areas are represented in gray. Moreover, the number at the bottom-right of a cell represents its identifier and the number in the black square at the top-left of a cell represents its $NCV_{\tilde{s}}$, where $\tilde{s}$ is the pseudo-semantic tag returned by Algorithm 1.

Fig. 5.d). This cloaking area has the maximum value of $NCR_{\tilde{s}}$ (i.e., $NCR_{\tilde{s}} = 4$). Note that the area $\{5, 6, 7, 8\}$ has also $NCR_{\tilde{s}} = 4$. However, its $SumNCV_{\tilde{s}} = 12$ which is less than the $SumNCV_{\tilde{s}}$ value of the area $\{1, 2, 5, 6\}$ i.e., $SumNCV_{\tilde{s}} = 18$. That is why the area $\{1, 2, 5, 6\}$ is selected and returned as $\tilde{r}$.

Building the Obfuscated Traces. For each city in the dataset, we choose the location traces and the semantic tag traces of 20 randomly chosen users. These traces are then obfuscated under the disjoint and the joint obfuscation approaches using Algorithms 1 and 2. To better capture the fact that the users do not share their locations and their corresponding semantic tags all the time on LBSNs, we apply the obfuscation algorithms with an additional *hiding process.* More precisely, we assume that at each time instant in the observation interval, both the user's location and its semantic tag can be hidden from the LBSN with *the hiding probability* λ or shared on the LBSN (and accordingly obfuscated by the algorithms under the disjoint and the joint approaches) with the probability $1 - \lambda$. The hidden locations and the hidden semantic tags are appeared in the obfuscated traces as *hidden*, denoted by $r_\perp$ and $s_\perp$ symbols, respectively. To build the obfuscated traces for each approach, we use all combinations of the following parameters:

the location obfuscation level (o_{loc}), the semantic tag obfuscation level (o_{sem}) and the hiding probability (λ), where $o_{loc} \in \{1, 2, 4, 8, 16\}$ and $o_{sem} \in \{0, 1, 2\}$ and $\lambda \in \{0, 0.2, 0.4, 0.6, 0.8\}$. For simplicity's sake, in what follows, we use the term *the obfuscation parameters* to refer to these parameters.[10]

5.1.4. Attacks and Privacy Evaluation

We implement the DBN models defined in Section 3, in Python by using *the pomegranate package* for the probabilistic models [47] and the Bayesian Belief Networks library provided by eBay [5]. The probability distributions of the basic DBN model are learned from the user traces using the Location-Privacy and Mobility Meter tool [36]. For the attacks, we apply the loopy belief propagation inference algorithm [43]. We perform the attacks for the observation interval of length 3. We then use the metrics defined in Section 4 to measure the location privacy and the semantic location privacy of the users.

5.2. Evaluation Results

We consider the obfuscation parameters and the choice of the city[11] as the parameters that can affect the performance of an obfuscation approach. Accordingly, in this section we present the results for different values of these parameters. In this way, we can compare the performance of the two obfuscation approaches under different values of these parameters and we can also show how changing these parameters affects the performance of the obfuscation approaches. Note that in addition to the privacy metrics presented in Section 4, to discuss the results, we use two additional metrics:

- **Ratio of the location privacy means** (denoted by *loc-priv-ratio.*) This is the ratio of the location privacy mean obtained for the joint approach to the location privacy mean obtained for the disjoint approach.
- **Ratio of the semantic location privacy means** (denoted by *sem-priv-ratio*). This is the ratio of the semantic location privacy mean obtained for the joint approach to the semantic location privacy mean obtained for the disjoint approach.

[10] In the dataset, the majority of the venues' semantic tags have a depth of 2 with respect to the Foursquare category tree. Thereby, we limit the maximum o_{sem} value to 2 in our evaluations.

[11] As we further discuss in Section 5.2.2, the main parameter in a city that affects the performance of an obfuscation approach is the distribution of the number of venues per region in that city.

The experimental evaluation results are depicted by Fig. 6, Fig. 7, Fig. 8 and Fig. 9. More precisely, Fig. 6 and Fig. 7, represent the location privacy results and the semantic location privacy results in the form of boxplots (i.e., first quartile, median, third quartile and outliers), respectively. Note that the location privacy in Fig. 6 is expressed in kilometres. Also, Fig. 8 and Fig. 9 represent the ratios of the location privacy means and the ratios of the semantic location privacy means in the form of scatterplots, respectively. Each figure has four subfigures (a), (b), (c) and (d). Each subfigure represents the aggregated results for different values of a given parameter, where the aggregation is performed over the results obtained for all users, all values of the obfuscation parameters and all cities. In the following, we first discuss the results for different obfuscation parameters. We then discuss the results for different cities.

5.2.1. Results for Different Obfuscation Parameters

We have three main observations regarding these results. Thus, in the following we first describe the observations. Then, we describe the reason behind the observations.

1. As the values of o_{loc}, o_{sem} and λ increase, the median location privacy and the median semantic location privacy for the both obfuscation approaches increase (see subfigures (a), (b), (c) of Fig. 6 and Fig. 7).

2. Under all values of o_{loc}, o_{sem} and λ, the median location privacy and the median semantic location privacy obtained for the joint approach are higher than the median location privacy and the median semantic location privacy obtained for the disjoint approach, respectively (see subfigures (a), (b), (c) of Fig. 6 and Fig. 7). There exist two exceptions to this observation. The first exception is the case where $o_{loc} = 1$. In this case, the median location privacy is the same for the both obfuscation approaches (See Fig. 6.a). Also, the median semantic location privacy is the same for the both obfuscation approaches (see Fig. 7.a). In fact, in the case where $o_{loc} = 1$, no location obfuscation is performed since the cloaking area can contain only one location (region) which is the actual location of the user. We know that the main difference between the obfuscation approaches is in the way that they obfuscate the actual location. Accordingly, in the case where no location obfuscation is performed, there is no difference between the performance of the two approaches. The second exception is the case where $o_{sem} = 0$. In this case, the median semantic location privacy is the same for the both obfuscation approaches (See Fig. 7.b). Recall that in the case where $o_{sem} = 0$, there is no semantic tag obfuscation. Accordingly, the median

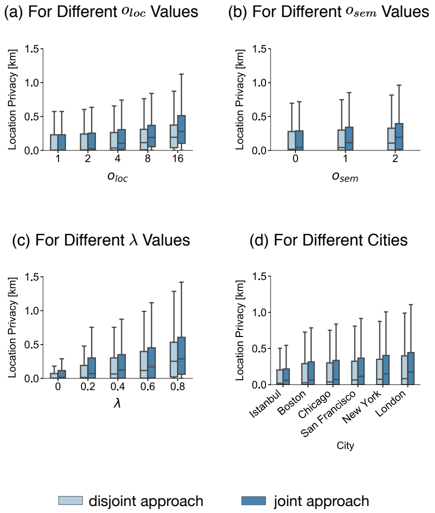

Figure 6: Location privacy results.

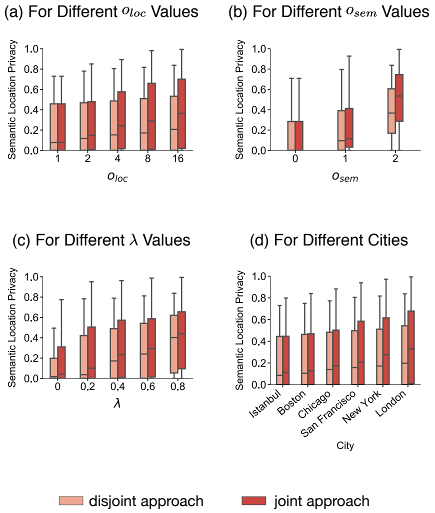

Figure 7: Semantic location privacy results.

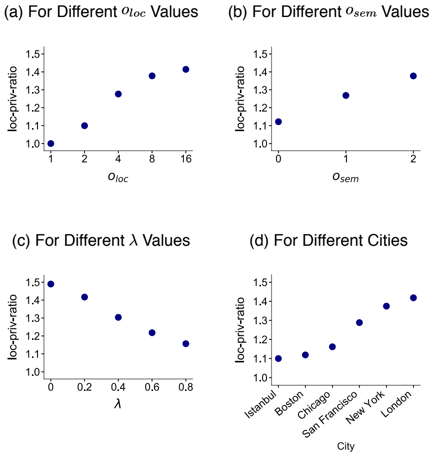

Figure 8: *loc-priv-ratio* for different parameters. Note that *loc-priv-ratio* is the ratio of the location privacy mean obtained for the joint approach to the location privacy mean obtained for the disjoint approach.

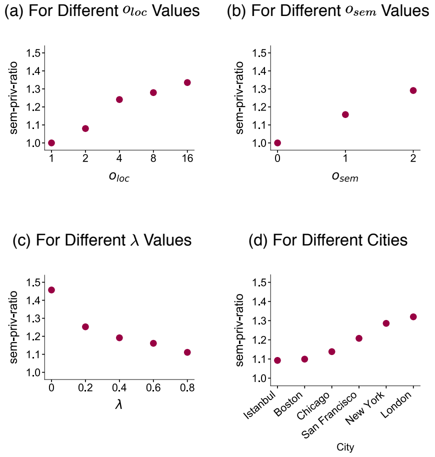

Figure 9: *sem-priv-ratio* for different parameters. Note that *sem-priv-ratio* is the ratio of the semantic location privacy mean obtained for the joint approach to the semantic location privacy mean obtained for the disjoint approach.

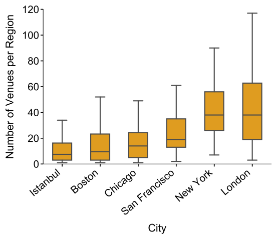

Figure 10: Distribution of number of venues per region (location) in different cities.

semantic location privacy is the same regardless of the obfuscation approach.[12]

3. As the value of o_{loc} increases, the values of *loc-priv-ratio* and *sem-priv-ratio* also increase (see Fig. 8.a and Fig. 9.a). Similarly, as the value of o_{sem} increases, the values of *loc-priv-ratio* and *sem-priv-ratio* increase (see Fig. 8.b and Fig. 9.b). However, as the value of λ increases, the values of *loc-priv-ratio* and *sem-priv-ratio* decrease (see Fig. 8.c and Fig. 9.c)

To explain these observations, we apply the following reasoning. As the value of o_{loc} increases, the number of regions (locations) in the cloaking area increases. Thus, by increasing o_{loc}, the performance of the both approaches improve. Also, as the value of o_{sem} increases, the number of semantic tags that can be semantically compatible with the obfuscated semantic tag increases. This, in turn, increases the chance of having more semantically compatible regions with the obfuscated semantic tag in every potential cloaking area. Thus, by increasing o_{sem} the performance of the both approaches improve. Moreover, we observe that by increasing o_{loc} and o_{sem}, the values of *loc-priv-ratio* and *sem-priv-ratio* also increase. For instance, in the case where $o_{loc} = 2$, we have *loc-priv-ratio* $= 1.09$ and *sem-priv-ratio* $= 1.07$, whereas in the case where $o_{loc} = 16$, we have *loc-priv-ratio* $= 1.41$ and *sem-priv-ratio* $= 1.33$ (see Fig. 8.a and Fig. 9.a). Also, in the case where $o_{sem} = 1$, we have *loc-priv-ratio* $= 1.26$ and *sem-priv-ratio* $= 1.15$, whereas in the case where $o_{sem} = 2$, we have *loc-priv-ratio* $= 1.37$ and *sem-priv-ratio* $= 1.29$ (see Fig. 8.b and Fig. 9.b). Roughly speaking, this means that the joint approach shows a much better performance compared to the disjoint approach under higher values of o_{loc} and o_{sem}. In fact, as the value of o_{loc} increases, the number of candidate regions for being in the cloaking area also increases. This, in turn, increases the chance that a greater number of the candidate regions are semantically compatible with the obfuscated semantic tag. Similarly, as the value of o_{sem} increases, the chance that a greater number of candidate regions are semantically compatible with the obfuscated semantic tag increases. The joint approach takes advantage of this increase, i.e., as the number of semantically compatible candidate regions increases, the joint approach selects a cloaking area with a greater number of semantically compatible regions and semantically compatible venues, whereas the disjoint approach is oblivious to the concept of semantic compatibility. Accordingly, the performance

[12] Note that in Fig. 6.a in the case where $o_{loc} = 1$, the location privacy results are not equal to zero. Also, in Fig. 7.b in the case where $o_{sem} = 0$, the semantic location privacy results are not equal to zero. The reason is that our plots aggregate all results including the results for the values of λ (i.e., the hiding probability) that are not equal to zero. Accordingly, even in the absence of location obfuscation or semantic tag obfuscation, the privacy results in the figures are not equal to zero.

of the disjoint approach does not improve as much as the performance of the joint approach by increasing the values of o_{loc} and o_{sem}. We also observe that as the value of λ increases, the performance of the both approaches improves. However, by increasing λ, the values of *loc-priv-ratio* and *sem-priv-ratio* decrease. Roughly speaking, this means that by increasing λ, the difference between the performance of the both approaches becomes less significant. Intuitively, this is because by increasing λ, we increase the number of hidden locations and hidden semantic tags compared to the number of the obfuscated locations and the obfuscated semantic tags in the obfuscated traces. This, in turn, increases the privacies resulting for the both approaches but it also decreases the importance of the obfuscation approach in defining the amount of the resulting privacies.

5.2.2. Results for Different Cities

The main parameter in a city that affects the performance of an obfuscation approach is the distribution of the number of venues per region in that city. Accordingly, in the following, we first discuss the distribution of the number of venues per region in each city, we then present the evaluation results for each city.

Fig. 10 depicts the distribution of the number of venues per region (location) in different cities. More precisely, in this figure for each city, we draw one boxlpot to aggregate the number of venues per region for all regions in that city. Moreover, let *MNV* of a city denote the median of the number of venues per region for that city, then the cities in the City axis of Fig. 10 are ordered in an ascending order with respect to their *MNV* values. Note that in Fig. 10, the horizontal line drawn in the middle of each city's boxplot corresponds to the *MNV* of that city. Thus, the lowest *MNV* corresponds to the city of Istanbul with $MNV = 7.5$ and the highest *MNV* corresponds to the city of London with $MNV = 38$. To better illustrate the dependency of the privacy results in each city to the number of venues per region, we also arrange the cities in the City axes of Fig. 6.d, Fig. 7.d, Fig. 8.d and Fig. 9.d in the same order as Fig. 10's City axis. Again, we have three observations regarding the results:

1. As *MNV* increases across the cities, the median location privacy and the median semantic location privacy also increase for the both obfuscation approaches (see Fig. 6.d, Fig. 7.d, respectively).

2. In all cities, the median location privacy and the median semantic location privacy obtained for the joint approach are higher than the median location privacy and the median semantic location privacy obtained for the disjoint approach, respectively (see Fig. 6.d, Fig. 7.d, respectively).

3. As *MNV* increases across the cities, the values of *loc-priv-ratio* and *sem-priv-ratio* increase (see Fig. 8.d, Fig. 9.d, respectively). So, the lowest values of *loc-priv-ratio* and *sem-priv-ratio* correspond to the city of Istanbul (with *loc-priv-ratio* = 1.10 and *sem-priv-ratio* = 1.09) and the highest values of *loc-priv-ratio* and *sem-priv-ratio* correspond to the city of London (with *loc-priv-ratio* = 1.41 and *sem-priv-ratio* = 1.32).

The reason behind these observations is as follows. Intuitively, the greater is the number of the venues in a region (location), the higher is the chance that a number of these venues are semantically compatible with a given obfuscated semantic tag. Thereby, the chance that the regions of a city are semantically compatible with a given obfuscated semantic tag increases as *MNV* in that city increases. Thus, regardless of the obfuscation approach, a cloaking area made in a city with a higher *MNV* has a better chance to have more semantically compatible regions with a given obfuscated semantic tag. Accordingly, as *MNV* increases, the performance of the both obfuscation approaches improve. However, the disjoint obfuscation approach is oblivious to the concept of semantic compatibility. Thus, as *MNV* increases, the disjoint approach cannot take advantage of the increase in the number of semantically compatible regions as much as the joint approach and its performance does not improve as much as the performance of the joint approach. Hence, as *MNV* increases, the values of *loc-priv-ratio* and *sem-priv-ratio* also increase.

6. Utility Discussion

As discussed in this paper, the location obfuscation and the semantic tag obfuscation increase user's privacy. However, they can at the same time reduce user's *perceived quality of service* (also known as *utility*). In this section, we first present the definition of utility and how it can be taken into consideration while building privacy protection mechanisms. We then present our conjecture about the performance of the obfuscation approaches in terms of utility. We intend to test the correctness of our conjecture in a future work.

To the best of our knowledge, the only work in the literature that presents a methodology to quantify the utility loss caused by obfuscation is the one presented in [7, 25]. According to the authors, a user's purpose behind a check-in plays a significant role in determining the utility of the check-in. Thus, they define utility as "*the extent to which the initial purpose of a check-in is still met after an obfuscation function is applied*" [7, 25]. To identify the main purposes behind user check-ins and to see how location obfuscation and semantic tag obfuscation affect the utility of check-ins, they run a targeted user study on a group of Foursquare users. The results of the study show that a user's purpose behind a check-in is not always to communicate her exact location and its semantic tag but can be related to higher level social goals such as informing about her current activity or mood. For instance, by checking-in to a restaurant in a city, a user might only want to inform her friends about her activity (i.e., eating in a restaurant in that city) without really wanting to reveal the exact type of the restaurant and its full address [7, 25]. This means that obfuscating a user's location and its semantic tag does not always drastically reduce utility. Thus, if a user's purpose behind a check-in can be inferred (e.g., by machine learning algorithms) and the utility loss for different obfuscation levels can be predicted using the inferred purpose, then the privacy protection mechanism can adjust the obfuscation levels to best match the user's preferences in terms of utility and privacy. The authors present an implementation of the building blocks of such privacy protection mechanism. More specifically, they present machine learning algorithms that can infer the purpose of check-ins based on some check-in features (e.g., textual information) and user behaviors. They also present a utility model that predicts the utility loss caused by obfuscation given the inferred purpose of the check-in and the obfuscation level [7, 25].

Considering the utility definition described above, it is hard to conjecture about the joint approach's performance in terms of utility and compare it with the disjoint approach's performance. The reason is that in both approaches the location obfuscation level and the semantic tag obfuscation level are the input parameters whose values cannot be changed by the obfuscation algorithms. Thus, given a location obfuscation level and a semantic tag obfuscation level, our joint approach uses the same location obfuscation level and the same semantic tag obfuscation level as the disjoint approach to obfuscate the actual location and its semantic tag. The only difference between the two approaches is in the way that they select the cloaking area. Thus, based on the way that the cloaking areas are selected in these approaches, we can conjecture that in check-in cases where the semantic dimension of the venue's location is important to a user, the joint approach can cause a smaller utility loss compared to the disjoint approach. In fact, as already discussed in this paper, the semantic dimension of a venue's location is not only captured by the semantic tag of the venue but also by the cloaking area that is selected for obfuscating the location of that venue. Our joint approach tries to better capture the semantic dimension by selecting a cloaking area that has the maximum number of semantically compatible regions with the obfuscated semantic tag. To better illustrate this point, we use again the toy examples in Fig. 3 and we add the following assumption to them: region 1 and region 3 are located in a neighborhood which is known for its good restaurants and shops (called neighborhood A), whereas region 2 is located in a neighborhood which is known for its museums and offices (called neighborhood B). Recall that in both toy examples, a user Alice wants to check-in to venue “Super Duper Burger” on a privacy-aware LBSN. So, in both toy examples her location’s semantic tag (i.e., “burger joint”) is replaced by the generalized tag “restaurant” during the semantic tag obfuscation process. In the toy example of Fig. 3.a, by applying the disjoint approach, her location (i.e., region 1) is replaced by the cloaking area made of region 1 and region 2. This cloaking area is located partly in neighborhood A and partly in neighborhood B. On the other hand, in the toy example of Fig. 3.b, by applying the joint approach, her location (i.e., region 1) is replaced by the cloaking area made of region 1 and region 3. This cloaking area is located entirely in neighborhood A. Intuitively, it makes more sense for Alice to inform her friends that she eats in a restaurant located in neighborhood A (which is known for its good restaurants), than in a restaurant located either in neighborhood A or in neighborhood B. Accordingly, the utility loss caused by the joint obfuscation approach is smaller than the utility loss caused by the disjoint obfuscation approach unless we assume that the semantic dimension of the neighborhood where the restaurant is located is not important to Alice. In addition, if the Alice’s purpose behind the check-in is not sharing the exact location of the restaurant but is sharing the fact that she eats in a restaurant in neighborhood A, then her purpose is still preserved even after the location obfuscation is performed by the joint approach and therefore, in this case the joint obfuscation causes no utility loss.

The fact that our joint approach tries to better capture the semantic dimension of the venue’s location compared to the disjoint approach is particularly important if we consider another important result obtained from the user study in [7, 25]. According to this result, the semantic tag obfuscation has a significantly larger negative impact on utility compared to the location obfuscation for the majority of check-ins. This implies that the semantic dimension of a venue’s location is more important to a user than its geographical dimension for the majority of cases. Accordingly, we expect that the use of the joint approach instead of the disjoint approach will make a change in terms of utility for many check-ins and decrease the utility loss.

7. Related Work

The problem of protecting location privacy of users in LBSNs (and in LBSs, in general) has been extensively studied in the literature and various protection mechanisms are proposed. Some location privacy protection mechanisms use cryptographic operations [15, 24, 10]. However, these schemes usually require technical modifications of the service. Many of the location privacy protection mechanisms apply location obfuscation. The popularity of the location obfuscation lies in the fact that it does not require changing the infrastructure, as it can be performed entirely on the user's side [55]. There exist different methods to obfuscate a location, for instance, by *hiding the location* from the LBS [6, 20], by *perturbing the location* (e.g., by adding noise to the location coordinates) [2], by *generalizing the location* (e.g., by merging the location with nearby locations using a cloaking algorithm) [22, 21, 41, 11, 27, 58, 59, 62] and by *adding fake (dummy) locations to the actual location* [12, 28, 30, 8].[13] Our work differs from these works by the fact that it considers not only the obfuscation of location but also the obfuscation of the semantic information, where the obfuscation of the semantic information is used to protect not only the semantic location privacy but also the location privacy. In addition, the location obfuscation in our work is performed with respect to the obfuscated semantic information, whereas the location obfuscation in these works is semantic-oblivious.

There exist several works that propose metrics for quantifying location privacy and define requirements that should be met to ensure successful location obfuscation [32, 14, 16, 53, 54, 50]. In particular, in [53, 54, 50] Shokri et al. propose a framework to quantify location privacy. The framework proposed in this paper is an extension of Shokri's framework. However, there exist two main differences between our framework and the Shokri's framework. Firstly, contrary to Shokri's framework that focuses only on users' locations and their location privacy, our framework takes into consideration also the locations' semantic tags and the users' semantic location privacy. Secondly, our framework relies on Dynamic Bayesian Network (DBN) models for implementing the attacks whereas Shokri's framework relies on Hidden Markov Models for implementing the attacks. In the case of Shokri's framework, relying on Hidden Markov Models for implementing the attacks is fine since the hidden state involves only one type of a user private attribute (i.e., user's location). However, as already discussed in Section 3.1, if the hidden state involves not only location but also other private attributes such as location's semantic tag, using Hidden Markov Models for implementing the attacks makes the inferences more complex and less efficient and it is better to use DBN models.

There exist also a number of works in the literature regarding the interdependent location privacy problem [38, 39, 49, 56, 23]. This problem is a subcategory of interdependent privacy problems. Roughly speaking, interdependent privacy

[13] We refer the interested reader to [31, 57, 51, 52] for detailed surveys on location obfuscation methods.

risks arise from the situations where the privacy of an individual is affected by the actions of other individuals or the data shared by other individuals [26]. For instance, according to the results of a survey, Foursquare users frequently report their co-locations with their friends (e.g., by tagging their friends' names) while checking-in to venues [39]. Thus by doing so, they put at risk not only their own location privacy but also the location privacy of their friends. In [38, 39], Olteanu et al. propose a framework to quantify the impact of co-location information on location privacy. By running different inference attacks, they show that the co-location information decreases the location privacy of individuals considerably. They also propose some countermeasures that mitigate the impact of co-location information on the location privacy. There are some similarities between their work and our work. For instance, like our work, their work is built upon Shokri's framework for quantifying location privacy [53, 54, 50]). Also, to perform some privacy attacks they rely on bayesian network models. Thus, by combining our work with their work, one can build a richer framework for quantifying privacy and can possibly propose more advanced privacy protection mechanisms.

The semantic dimension of location information has been studied in various works [33, 37, 17, 61, 4, 35]. For instance, in [33], authors propose an algorithm that by using machine learning techniques generates semantic tags (labels) for places based on the timing of visits to those places, nearby businesses and demographics of the users. In [37], the authors propose a location type classification model that classifies location types based on the content of users' check-in tweets. In [17], the authors present the semantic annotation of location history as a way to better understand the purpose of a mobile user for visiting the locations. Accordingly, different methods for semantic annotation of location history are studied. These methods use the spatiotemporal documents collected from social media (e.g., geotagged tweets) for semantic annotation. In [61], an approach is introduced which predicts the next location of mobile users based on both the geographic and semantic features of users' trajectories where the semantic features are extracted by mining the semantic trajectory patterns for each user. In [4], the authors rely on semantic labelling techniques to protect users' identity and their location privacy in LBSs and LBSNs. More precisely, they argue that many applications are not interested in (geographical) location data and they can instead use the semantic information about the locations to perform their tasks. For instance, a virtual assistant application such as Apple's Siri or Amazon's Alexa can be set to notify the user when she reaches her home or work place and is indifferent as to the geographical locations of the home and work place. Accordingly, the authors propose an approach called *semantic cloaking* in which the user's geographical locations are replaced by their semantic labels and then used by different applications. The semantic labelling is performed by applying machine learning techniques and by using several temporal features such as duration of stay at a location and day of week. As described, the goal of the work in [4] is to provide a mechanism to protect the users' identity and location privacy and contrary to our work, it does not provide a mechanism to protect semantic location privacy.

The problem of protecting semantic location privacy is also studied in the literature, although to a lesser extent than the problem of protecting location privacy. For instance, in [60] and [34], cloaking algorithms are proposed that cloak the actual location by merging it with semantically heterogenous locations (these algorithms are based on the concept of *l-diversity* originally proposed in [40]). In [13], the authors propose a personalized and semantic-aware location cloaking algorithm. The algorithm adjusts the spatial resolution of the cloaking area based on the sensitivity of the locations for the user, e.g., near a sensitive location such as a hospital, it coarsens the spatial resolution more. Our work differs from these works mainly because of the following fact: to protect the semantic location privacy, our work considers the obfuscation of both the location information and the semantic information whereas these works only consider the obfuscation of the location information.

The disjoint obfuscation approach discussed in this paper, was originally introduced in [1]. Our work is close to the work presented in [1], in the sense that it assumes a similar system model and adversary model. In fact, our work and the work in [1] are both built upon the Shokri's framework for quantifying location privacy [53, 54, 50]) and they both rely on bayesian network models for implementing the inference attacks. However, as already discussed in this paper, our work tries to improve the work in [1], by proposing a joint obfuscation approach.

8. Conclusion and Future Work

In this paper, we have introduced a joint semantic tag-location obfuscation approach for protecting the location privacy and the semantic location privacy of users in LBSNs (and in LBSs, in general). This approach is designed to overcome the drawbacks of the existing disjoint approach, by performing the location obfuscation based on the result of the semantic tag obfuscation. We provided a formal framework that can be used for evaluation and comparison of the joint approach with the disjoint approach. Using a dataset of real-world user mobility traces collected from six different cities, we performed an experimental evaluation to compare the joint approach's performance with the disjoint approach's performance in terms of location privacy protection and semantic location privacy protection. According to our experimental results, in almost all cases (i.e., in different cities and with different obfuscation parameters), the joint approach outperforms the disjoint approach. We also studied how different parameters (i.e., the obfuscation parameters and the distribution of the number of venues per region in different cities) can affect the performance of the obfuscation approaches. In particular, we showed that compared to the disjoint approach, the joint approach can take better advantage of higher values of the location obfuscation level, the semantic tag obfuscation level and the number of venues per region in a city and exhibits even more satisfactory performance under higher values of these parameters.

As a potential future work, we consider to perform the following improvements on the current work:

- **Adding the Time Periods to the System Model.** Venues have opening hours which are defined based on the time periods (i.e., morning, afternoon, evening, night). For instance, a restaurant is closed in the morning and is open in the afternoon. Accordingly, the adversary can filter out the venues in the cloaking area based on time periods and increase his chance to infer the actual location and its semantic tag. Thus, to prevent such attacks, the time periods should also be taken into consideration while building the cloaking areas [1].

- **Considering the Semantic Tag Obfuscation Level and the Location Obfuscation Level as the instantiations of Random Variables.** In the current work, the semantic tag obfuscation level and the location obfuscation level are considered as parameters whose values remain constant during the whole observation interval. In a future work, we can assume that these obfuscation levels are the instantiations of random variables that can take different values at each time instant based on user's utility and privacy preferences. In fact, as discussed in detail in Section 6, the privacy protection mechanism (PPM) can use a utility model that for each check-in, predicts the utility loss caused by using different obfuscation levels. Then, based on the predicted utility losses, the PPM can select the obfuscation levels that best match the user's utility and privacy preferences for that check-in.

- **Performance Optimization in Terms of Time and Space Complexity.** The goal of this paper is to present the idea of joint obfuscation and to show how it can improve the privacy protection compared to the existing disjoint approach. Thereby, it is beyond the scope of this paper to discuss the time and space complexity of the obfuscation approaches. Thus, another issue that can be investigated as future work is to optimize the joint obfuscation algorithm and compare its performance in terms of time and space complexity with the disjoint approach.

Acknowledgements

This research is partially funded by a UNIL/CHUV postdoc mobility grant disbursed by University of Lausanne and Lausanne University Hospital. We also gratefully thank Berker Ağir for his comments and help regarding the disjoint obfuscation approach.

Appendix A. A thorough comparison between the present work and the primary work published as a workshop paper

The work presented in this paper is an extension of the primary work presented as a workshop paper in [9]. In both papers, we introduce a joint semantic tag-location obfuscation approach in which location obfuscation is performed based on the result of the semantic tag obfuscation. In both papers, we present a formal framework and an experimental evaluation to compare the performance of our joint obfuscation approach with the performance of the existing solution in the literature, i.e., the disjoint obfuscation approach. However, the work presented in the workshop paper is limited in many ways and does not address various issues. The present paper addresses the issues that are not covered by the workshop paper. Accordingly, the present paper is a distinct paper from the workshop paper and contains a significant amount of new material. Below, we list the main additional contributions of the present paper compared to the workshop paper and we describe the importance of each contribution:

1. In the workshop paper, we show that the joint obfuscation approach outperforms the disjoint obfuscation approach only in terms of *location privacy*, whereas in the present paper, we show that the joint obfuscation approach outperforms the disjoint obfuscation approach both in terms of *location privacy* and *semantic location privacy*. As discussed in the present paper, semantic location privacy is an important aspect of user privacy and is a different concept than the location privacy. In fact, one of our main goals for designing the joint obfuscation approach was to show that it could better protect both the location privacy and the semantic location privacy of the users compared to the disjoint obfuscation approach. However, in the primary work presented in the workshop paper, there is no mention of semantic location privacy and we only show that our joint obfuscation approach performs better than the disjoint approach in terms of location privacy. Thus, the present paper completes the work presented in the workshop paper by including the semantic location privacy parts. Accordingly, all sections of the present paper (and not only the experimental evaluation) are extended to contain information regarding the semantic location privacy. More

precisely, we modified the abstract and the introduction of the present paper so that they contain information regarding the semantic location privacy. In the adversary model in Section 2.5.2 on page 13, we added a new type of attack (i.e., *semantic-tag inference attack*) against semantic location privacy of users. Thus, the adversary model in the present paper performs two types of attacks (i.e., *semantic-tag inference attack* and *location-inference attack*), whereas the adversary model in the workshop paper only performs one type of attack (i.e., *location-inference attack*). We also added the definitions of new metrics including *semantic location privacy metric* to Section 4.2 on page 19 and *ratio of the semantic location privacy means* (denoted by *sem-priv-ratio*) to Section 5.2 on page 26. In the experimental evaluation in section 5.2, we added new figures (i.e., Fig. 7 and Fig. 9 on page 28) which depict the semantic location privacy results. We also extended the discussions in Section 5.2 so that they cover also the semantic location privacy results (see pages 26–31). Finally, we extended the related work in Section 7 (see pages 33–36) and the conclusion and future work in Section 8 (see pages 36–37), so that they include discussions regarding the semantic location privacy.

2. In the workshop paper, we compare the performance of our joint obfuscation approach with the performance of the disjoint obfuscation approach only under different values of the obfuscation parameters. However, the performance of these approaches can also be influenced by external factors such as the choice of the city. Accordingly, we added a new section (i.e. Section 5.2.2 on pages 30–31) to the present paper, which introduces the evaluation results for different cities and discusses new figures including Fig. 6.d, Fig. 7.d, Fig. 8.d, Fig. 9.d and Fig. 10 on page 28. This section also studies how the distribution of the number of venues per region in a city can affect the performance of the obfuscation approaches in that city and presents a new metric i.e., *the median of the number of venues per region* for a city denoted by *MNV*.

3. In the workshop paper, there is no discussion regarding the user's perceived quality of service (also known as *utility*). However, from a practical point of view, it is important to discuss the performance of our joint obfuscation approach also in terms of utility and compare it with the performance of the disjoint approach. Accordingly, we added to the present paper a new section (i.e., Section 6 on pages 31–33), which compares the performance of the obfuscation approaches in terms of utility.

4. In the workshop paper, we only mention the names of two *dynamic bayesian network (DBN) inference algorithms* that can be used by the adversary for per-

forming his attacks. However, in the present paper we added a new section (i.e., Section 3.2 on page 18), in which we discuss these DBN inference algorithms in detail. In particular, we discuss the origin of these DBN inference algorithms and how they work. We also compare the cost and the performance of these algorithms and recommend the algorithms that can be used for the adversary's attacks based on the available resources.

5. In the workshop paper, the obfuscation algorithms are described briefly without the use of any pseudo-codes or example walkthroughs. In the present paper, we added pseudo-codes for the obfuscation algorithms (i.e., Algorithm 1 and Algorithm 2 on page 21) to Section 5.1.3. We also added an example walkthrough of the location obfuscation algorithm to Section 5.1.3. The example walkthrough begins on the second paragraph of page 24 and ends on the first paragraph of page 25 and has a figure (i.e., Fig. 5 on page 25). We believe that these additions can help the readers to get a better understanding of how these algorithms work.

6. The related work section of the workshop paper is short and limited to a brief comparison between our joint obfuscation approach and the famous location privacy protection mechanisms existing in the literature. In the present paper, new discussions are added to the related work in Section 7 on pages 33–36, in particular, regarding the interdependent location privacy, semantic tag labelling and semantic location privacy. Accordingly, the total number of referenced papers in the present paper is equal to 62 whereas the total number of referenced papers in the workshop paper is equal to 27. We also added some ideas for future work to Section 8 on pages 36–37 of the present paper.

7. We improved all sections of the present paper by better explaining our ideas and by adding more details compared to the workshop paper. In particular, we added a new figure (i.e., Fig. 2 on page 2) to the introduction section to better explain the concept of semantic tag hierarchical tree. We also added Table 1 on page 9 to the system model section. This table summarizes the notations used throughout the paper. Finally, we added 12 footnotes (i.e., Footnotes 2–13) to the present paper.

References

[1] B. Ağir, K. Huguenin, U. Hengartner, and J.-P. Hubaux. On the Privacy Implications of Location Semantics. In *PoPETs Journal*, vol. 2016, no. 4, pp. 165–183, 2016.

[2] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *Proc. ACM SIGSAC'13*, pp. 901–914, 2013.

[3] B. Bamba, L. Liu, P. Pesti, and T. Wang. Supporting Anonymous Location Queries in Mobile Environments with Privacygrid. In *Proc. ACM WWW'08*, pp. 237–246, 2008.

[4] O. Barak, G. Cohen, and E. Toch. Anonymizing Mobility Data using Semantic Cloaking. In *Pervasive and Mobile Computing*, vol. 28, pp. 102–112, 2016.

[5] The Bayesian Belief Networks library. `https://github.com/eBay/bayesian-belief-networks`

[6] A. R. Beresford and F. Stajano. Location Privacy in Pervasive Computing. In *IEEE Pervasive Computing*, vol. 2, no. 1, pp. 46–55, 2003.

[7] I. Bilogrevic, K. Huguenin, S. Mihaila, R. Shokri, and J.-P. Hubaux. Predicting Users Motivations behind Location Check-Ins and Utility Implications of Privacy Protection Mechanisms. In *Proc. NDSS'15*, pp. 1–11, 2015.

[8] V. Bindschaedler and R. Shokri. Synthesizing Plausible Privacy-Preserving Location Traces. In *Proc. IEEE S&P'16*, pp. 546–563, 2016.

[9] B. Bostanipour and G. Theodorakopoulos. Joint Obfuscation for Privacy Protection in Location-Based Social Networks. In *DPM workshop of ESORICS 2020*, 2020.

[10] B. Carbunar, R. Sion, R. Potharaju, and M. Ehsan, The shy mayor: Private Badges in Geosocial Networks, In *Proc. ACNS'12*, pp. 436–454,2012.

[11] R. Cheng, Y. Zhang, E. Bertino, and S. Prabhakar. Preserving User Location Privacy in Mobile Data Management Infrastructures. In *Proc. PETS'06*, 2006.

[12] R. Chow and P. Golle. Faking Contextual Data for Fun, Profit, and Privacy. In *Proc. ACM WPES'09*, pp. 105–108, 2009.

[13] M.L. Damiani, E. Bertino, and C. Silvestri. Protecting Location Privacy against Spatial Inferences: The PROBE Approach. In *Proc. ACM SIGSPATIAL'09*, pp. 32–41, 2009.

[14] M. Decker, Location Privacy–An Overview, In *Proc. IEEE ICMB'08*, pp. 221–230, 2008.

[15] C. Dong and N. Dulay, Longitude: A Privacy-Preserving Location Sharing Protocol for Mobile Applications, In *Proc. IFIPTM'11*, pp. 133–148, 2011.

[16] M. Duckham, Moving forward: Location Privacy and Location Awareness, In *Proc. ACM SPRINGL'10*, pp. 1–3, 2010.

[17] F. Wu, Z. Li, W.-C. Lee, H. Wang, and Z. Huang. Semantic Annotation of Mobility Data using Social Media. In *Proc. WWW '15*, 2015.

[18] Facebook. `http://www.facebook.com/`.

[19] Foursquare. `https://foursquare.com/`.

[20] J. Freudiger, R. Shokri, and J.-P. Hubaux. On the Optimal Placement of Mix Zones. In *Proc. PETS'09*, pp. 216–234, 2009.

[21] B. Gedik. Location Privacy in Mobile Systems: A Personalized Anonymization Model. In *Proc. IEEE ICDCS'05*, 2005.

[22] M. Gruteser and D. Grunwald. Anonymous Usage of Location-Based Services through Spatial and Temporal Cloaking. In *Proc. MobiSys'03*, 2003.

[23] B. Henne, C. Szongott, and M. Smith. Snapme if you can: Privacy threats of other peoples geo-tagged media and what we can do about it. In *WiSec'13*, pp. 95–106, 2013.

[24] M. Herrmann, A. Rial, C. Diaz, and B. Preneel, Practical Privacy-Preserving Location-Sharing Based Services with Aggregate Statistics, In *Proc. ACM WiSec'14*, pp. 87?98, 2014.

[25] K. Huguenin, I. Bilogrevic, J. Soares Machado, S. Mihaila, R. Shokri, I. Dacosta and J.-P. Hubaux. A Predictive Model for User Motivation and Utility Implications of Privacy Protection Mechanisms in Location Check-Ins. n *IEEE Trans. on Mobile Computing*, 2018.

[26] M. Humbert, B. Trubert, K. Huguenin. A Survey on Interdependent Privacy. In *ACM Comput. Surv.*, 2020.

[27] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias. Preventing Location-Based Identity Inference in Anonymous Spatial Queries. In *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 19, no. 12, pp. 1719–1733, 2007.

[28] H. Kido, Y. Yanagisawa, and T. Satoh. An Anonymous Communication Technique Using Dummies for Location-Based Services. In *Proc. ICPS'05*, pp. 88–97, 2005.

[29] D. Koller and N. Friedman. Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning. The MIT Press, 2009.

[30] J. Krumm. Realistic Driving Trips for Location Privacy. In *Proc. IEEE PerCom'09*, pp. 25–41, 2009.

[31] J. Krumm. A Survey of Computational Location Privacy. In *Personal Ubiquitous Comput.*, vol. 13, no. 6, pp. 391–399, 2009.

[32] J. Krumm. Inference Attacks on Location Tracks. In *Pervasive*, pp. 127–143, 2007.

[33] J. Krumm and D. Rouhana, Placer: Semantic place labels from diary data, In *Proc. ACM UbiComp'13*, pp. 163–172, 2013.

[34] B. Lee, J. Oh, H. Yu, and J. Kim. Protecting Location Privacy using Location Semantics. In *Proc. ACM SIGKDD'11*, pp. 1289–1297, 2011.

[35] W. Li, P. Serdyukov, A. P. de Vries, C. Eichko, and M. Larson, The where in the tweet, In *Proc. ACM CIKM*, pp. 2473?2476, 2011.

[36] Location-Privacy and Mobility Meter. `https://icapeople.epfl.ch/rshokri/lpm/doc/`

[37] H. Liu, B. Luo, and D. Lee, Location Type Classification using Tweet Content, In *Proc. ICMLA'12*, vol. 1, pp. 232–237, 2012.

[38] A.-M. Olteanu, K. Huguenin, R. Shokri and J.-P. Hubaux. Quantifying the Effect of Co-Locations on Location Privacy. In *Proc. of PETS*, 2014.

[39] A.-M. Olteanu, K. Huguenin, R. Shokri, M. Humbert, and J.-P. Hubaux. Quantifying Interdependent Privacy Risks with Location Data. In *IEEE Trans. Mob. Comput.* vol. 16, no. 3, pp. 829–842, 2017.

[40] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. l-diversity: Privacy Beyond k-Anonymity. In *ACM Trans. Knowl. Discov. Data* . vol. 1, no.1, 2007.

[41] M. Mokbel, C. Chow, and W. Aref. The New Casper: Query Processing for Location Services without Compromising Privacy. In *Proc. VLDB06*, 2006.

[42] K. P. Murphy. Dynamic Bayesian Networks: Representation, Inference and Learning. Ph.D. Thesis, University of California, Berkeley, 2002.

[43] K. P. Murphy, Y. Weiss, and M. Jordan. Loopy Belief Propagation for Approximate Inference: an Empirical Study. In *UAI*, 1999.

[44] N. Noorshams and M. J. Wainwright. Stochastic Belief Propagation: A Low-Complexity Alternative to the Sum-Product Algorithm. In *IEEE Trans. Information Theory*, vol. 59, no. 4, pp. 1981–2000, 2013.

[45] J. Pearl. Reverend Bayes on Inference Engines: a Distributed Hierarchical Approach. In *Proc. AAAI'82*, pp. 133–136, 1982.

[46] J. Pearl. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, 1988.

[47] Pomegranate. `https://pomegranate.readthedocs.io/en/latest/`.

[48] C. Renso, S. Spaccapietra, and E. Zimnyi. Mobility Data: Modeling, Management, and Understanding. Cambridge University Press, New York, NY, USA, 2013.

[49] A. Sadilek, H. Kautz, and J. P. Bigham. Finding Your Friends and Following Them to Where You Are. In *Proc. of WSDM*, 2012.

[50] R. Shokri. Quantifying and Protecting Location Privacy. Ph.D. Thesis, EPFL, 2012.

[51] R. Shokri, J. Freudiger, M. Jadliwala, and J.-P. Hubaux. A Distortion-Based Metric for Location Privacy. In *Proc. ACM WPES'09*, pp. 21–30, 2009.

[52] R. Shokri, J. Freudiger, and J.-P. Hubaux. A Unified Framework for Location Privacy. In *Proc. HotPETs10*, 2010.

[53] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux. Quantifying Location Privacy. In *Proc. IEEE S&P'11*, pp. 247–262, 2011.

[54] R. Shokri, G. Theodorakopoulos, G. Danezis, J.-P. Hubaux, and J.-Y. Le Boudec. Quantifying Location Privacy: The Case of Sporadic Location Exposure. In *Proc. PETS'11*, pp. 57–76, 2011.

[55] R. Shokri, G. Theodorakopoulos, and C. Troncoso. Privacy Games Along Location Traces: A Game-Theoretic Framework for Optimizing Location Privacy. In *ACM Trans. Priv. Secur*, vol. 19, no. 4, pp.1–31, 2016.

[56] C. Vicente, D. Freni, C. Bettini, and C. S. Jensen. Location-related Privacy in Geosocial Networks. In *IEEE Internet Computing*, vol. 15, no. 3, pp. 20–27, 2011.

[57] M. Wernke, P. Skvortsov, F. Dürr, and Kurt Rothermel. A Classification of Location Privacy Attacks and Approaches. In *Personal Ubiquitous Comput.*, vol. 18, no. 1, pp. 163–175, 2014.

[58] T. Xu and Y. Cai. Exploring Historical Location Data for Anonymity Preservation in Location-Based Services. In *Proc. IEEE INFOCOM'08*, 2008.

[59] T. Xu and Y. Cai. Feeling-Based Location Privacy Protection for Location-Based services. In *Proc. ACM CCS'09*, 2009.

[60] M. Xue, P. Kalnis, and H. K. Pung. Location Diversity: Enhanced Privacy Protection in Location Based Services. In*Proc. LoCA'09*, pp. 70–87, 2009.

[61] J.J.-C. Ying, W.-C. Lee, T.-C. Weng, V.S. Tseng, Semantic Trajectory Mining for Location Prediction, In *Proc. ACM SIGSPATIAL'11*, pp. 34–43,2011.

[62] T.-H. You, W.-C. Peng, and W.-C. Lee. Protecting Moving Trajectories with Dummies. In *Proc. IEEE MDM'07*, pp. 278–282, 2007.