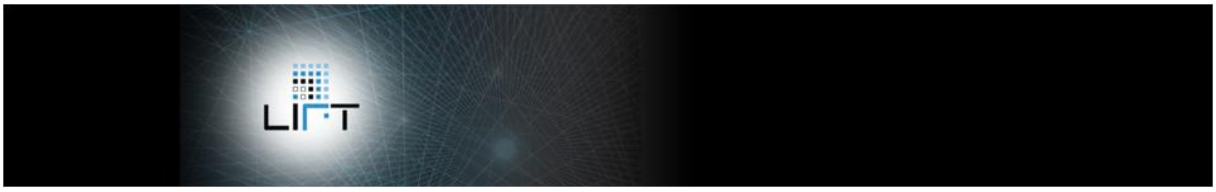


ICT, FET Open

LIFT ICT-FP7-255957

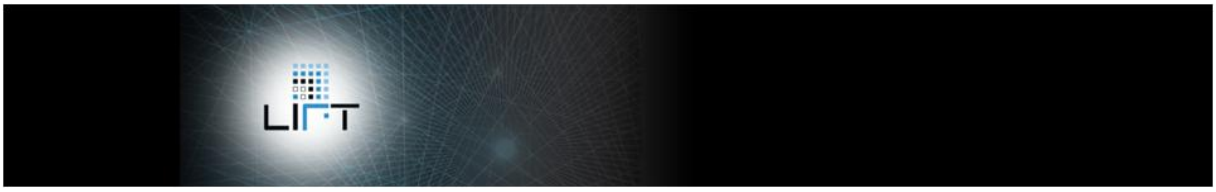
Using Local Inference in Massively Distributed Systems

Collaborative Project

D 2.1

Requirements for Privacy and Anonymity
1.10.2010 – 30.09.2011

Contractual Date of Delivery:	30.09.2011
Actual Date of Delivery:	30.09.2011
Author(s):	Anna Monreale, Wendy Wang, Dino Pedreschi, Christine Körner, Michael May
Institution:	CNR
Workpackage:	WP 2
Security:	PU
Nature:	R
Total number of pages:	59



Project coordinator name: Michael May

Project coordinator organisation name:
Fraunhofer Institute for Intelligent Analysis
and Information Systems (IAIS)
Schloss Birlinghoven, 53754 Sankt Augustin, Germany
URL: http://www.iais.fraunhofer.de

Revision: 3

Abstract:

This document is the LIFT deliverable of WP2 for the first review period (01.10.2010 – 30.09.2011). The document presents the state-of-the-art in privacy and anonymity, adapts the privacy paradigm for applications in LIFT and specifies privacy requirements for example applications of LIFT.

Revision history

Administration Status		
Project acronym: LIFT		**ID:** ICT-FP7-255957
Document identifier:	D 2.1 Requirements for Privacy and Anonymity (01.10.2010 – 30.09.2011)	
Leading Partner:	CNR	
Report version:	3	
Report preparation date:	20.09.2011	
Classification:	PU	
Nature:	REPORT	
Author(s) and contributors:	Anna Monreale (CNR), Wendy Wang (CNR), Dino Pedreschi (CNR), Christine Körner (FHG), Michael May (FHG)	
Status:	-	Plan
	-	Draft
	-	Working
	-	Final
	x	Submitted

Copyright

Project funded by the European Community
under the Information and Communication
Technologies Programme
Contract ICT-FP7-255957

Project Deliverable D2.1
Requirements for Privacy and Anonymity

Anna Monreale, Wendy Wang, Christine Körner, Dino Pedreschi and Michael May

Contents

1 Introduction **2**

2 State of the Art **3**
2.1 Individual Privacy Protection 3
2.2 Privacy and Anonymity for Data Publishing and Mining 4
2.3 Statistical Disclosure Control 10
2.4 Anonymity in Complex Data 13
2.5 Privacy by Design 15
2.6 Privacy by Design for Data Publishing 16

3 Privacy by Design for LIFT-based Systems **25**
3.1 Privacy Requirements for LIFT-based Systems 25
3.2 Properties of a Countermeasure 27

4 Privacy in Application Scenarios **29**
4.1 Privacy in Distributed Density Map Computation 29
4.2 Privacy for Measuring Customer-Location Interactions 39
4.3 Privacy for Counting Distinct Entities in a Region 43

5 Conclusion and Roadmap **48**

Chapter 1

Introduction

This report consists of the collaborative work pursued during the first year of the LIFT project by the participants in the work package WP2, entitled *Privacy & Anonymity*.

The ultimate mission of the WP2 is to develop privacy-preserving frameworks for guaranteeing privacy protection in LIFT-based systems, and therefore: (1) to define the requirements for privacy and anonymity of the LIFT-based systems; (2) to design, implement and test algorithms for guaranteeing privacy protection in these systems and for satisfying the privacy and anonymity requirements; (3) to investigate the impact of privacy-preserving approaches on the performance of the systems and on the data utility.

During the first year of LIFT, the WP2 participants interacted closely in three main activities:

- The alignment of participants' expertise and knowledge by means of a collective exploration of the state-of-the-art in privacy-preserving methods;
- the definition of a shared road-map of research directions in the design of privacy-preserving framework for LIFT-based systems;
- the concrete exploration of the possible privacy breaches in the LIFT-based systems with the general definition of the typical attacks that an adversary may conduct in the systems to infer sensitive information.

The aim of this report is to describe these three activities.

Chapter 2 presents the state-of-the-art in privacy and anonymity and describes in detail the *Privacy by Design* paradigm. Chapter 3 presents the definition of this paradigm for LIFT-based systems and so introduces the general definition of the privacy attack model to be taken into consideration and general requirements of the countermeasures against these attacks. These requirements will be the base of the privacy-preserving techniques that will be designing during the next years. In Chapter 4 we provide some examples of application scenarios with the definition of specific privacy requirements depending on the application. Here, we show how adequately customizing the attacks model in each specific scenario.

Chapter 2

State of the Art

In this Chapter we introduce the problem of the individual privacy protection in the context of data publication studied extensively in two different communities: in data mining and in statistics. After a general introduction in Section 2.1 we provide a survey of the main privacy and anonymity techniques proposed by the two different communities in Sections 2.2 and 2.3, analyzing them from the two perspectives. We proceed with anonymity in complex data in Section 2.4 and introduce the concept of *Privacy by Design* in Section 2.5. We conclude this chapter by specifying attack models for publishing of sequence and movement data in Section 2.6.

2.1 Individual Privacy Protection

In the last years, the importance of the privacy protection is rising thanks to the availability of large amounts of data. These data collections can be gathered from various channels. Typically, the data collector or data holder can releases these data to data miners and analysts who can conduct on them statistical and data mining analysis. The published data collections could contain personal information about users and their *individual privacy* could be compromised during analytical processes.

In recent years, individual privacy has been one of the most discussed jurisdictional issues in many countries. Citizens are increasingly concerned about what companies and institutions do with their data, and ask for clear positions and policies from both the governments and the data owners. Despite this increasing need, there is not a unified view on privacy laws across countries.

The European Union regulates privacy by Directive 95/46/EC (Oct. 24, 1995) and Regulation (EC) No 45/2001 (December 18, 2000). The European regulations, as well as other regulations such as the U.S. rules on protected health information (from HIPAA), are based on the notion of "non-identifiability".

The problem of protecting the individual privacy when disclosing information is not trivial and this makes the problem scientifically attractive. It has been studied extensively in two different communities: in data mining, under the general umbrella of *privacy-preserving data mining* (PPDM), and in statistics, under the general umbrella

of *statistical disclosure control* (SDC). Often, the different communities have investigated lines of work which are quite similar, sometimes with little awareness of this strong tie. The Figure 2.1 shows a taxonomy tree that describes our classification of the privacy-preserving techniques.

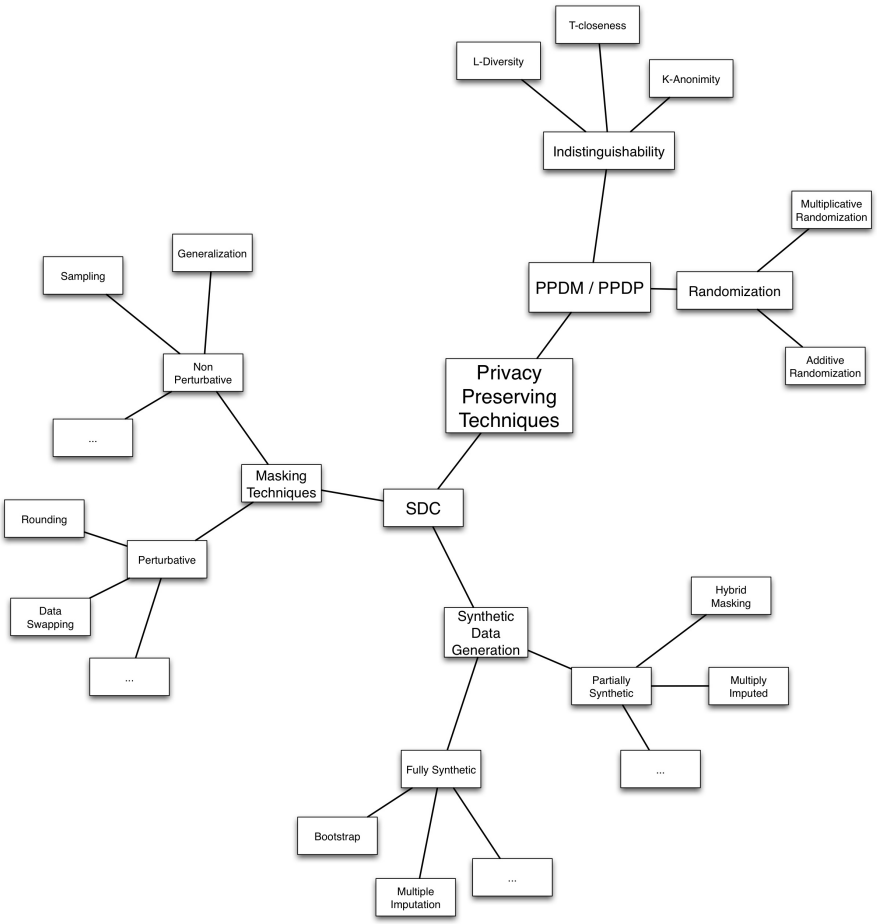


Figure 2.1: Taxonomy of privacy-preserving techniques

2.2 Privacy and Anonymity for Data Publishing and Mining

The importance of privacy-preserving data publishing (PPDP) and mining (PPDM) is growing thanks to the increasing capability of storing and processing large amounts of

data. In literature, many privacy-preserving techniques has been proposed by the data mining community and in this section we provide an overview of them.

2.2.1 Anonymity by Randomization

Randomization methods are used to modify data at aim of preserving the privacy of sensitive information. They were traditionally used for statistical disclosure control [5] and later have been extended to the privacy-preserving data mining problem[10]. Randomization is a technique for privacy-preserving data mining using a noise quantity in order to perturb the data. The algorithms belonging to this group of techniques first of all modify the data by using randomization techniques. Then, from the perturbed data it is still possible to extract patterns and models. In the following we present the most famous random perturbation techniques.

Additive Random Perturbation

In this section, we will discuss the method of *additive random perturbation* and its applications in data mining problem. This method can be described as follows. Denote by $X = \{x_1 \dots x_m\}$ the original dataset. The new distorted dataset, denoted by $Z = \{z_1 \dots z_m\}$, is obtained drawing independently from the probability distribution a noise quantity n_i and adding it to each record $x_i \in X$. The set of noise components is denoted by $N = \{n_1, \dots, n_m\}$. The original record values cannot be easily guessed from the distorted data as the variance of the noise is assumed enough large. Instead, the distribution of the dataset can be easily recovered. Indeed, if X is the random variable representing the data distribution for the original dataset, N is the random variable denoting the noise distribution, and Z is the random variable describing the perturbed dataset, we have:

$$Z = X + N$$
$$X = Z - N$$

Notice that, both m instantiations of the probability distribution Z and the distribution N are known. In particular, the distribution N is known publicly. Therefore, by using one of the methods discussed in [10, 8], we can compute a good approximation of the distribution Z, by using a large enough number of values of m. Then, by subtracting N from the approximated distribution of Z, we can compute N approximation of X. At the end of this process individual records are not available, while obtain a distribution only along individual dimensions describing the behavior of the original dataset X.

The additive perturbation method has been extended to several data mining problems. But, it is evident that traditional data mining algorithms are not adequate as based on statistics extracted from individual records or multivariate distributions. Therefore, new data mining approaches have to be devised to work with aggregate distributions of the data in order to obtain mining results. This can sometimes be a challenge. In the works presented in [10, 94, 95] authors propose new techniques based on the randomization approach in order to perturb data and then, we build classification models over randomized data. In particular, the work in [10] is based on the fact that the probability

distribution is sufficient in order to construct data mining models as classifiers. Authors show that the data distribution can be reconstructed with an iterative algorithm. Later, in [8] Agrawal and Aggarwal show that the choice of the reconstruction algorithm affects the accuracy of the original probability distribution. Furthermore, they propose a method that converges to the maximum likelihood estimate of the data distribution. Authors in [94, 95] introduce methods to build a Naive Bayesian classifier over perturbed data. Randomization approaches are also applied to solve the privacy-preserving association rules mining problem as in [78, 37]. In particular, the paper [78] presents a scheme attempting to maximize the privacy to the user and to maintain a high accuracy in the results obtained with the association rule mining. While, in [37] authors present a framework for mining association rules from randomized data. They propose a class of randomization operators more effective than uniform distribution and a data mining approach to recover itemset supports from distorted data.

Multiplicative Random Perturbation

For privacy-preserving data mining, *multiplicative random perturbation* techniques can also be used. There exist two types of multiplicative noise. The first one applies a logarithmic transformation on the data, and generates a random noise that follows a multivariate normal distribution with mean equal to zero and constant variance. Then, this noise is added to each element of the transformed data. Finally, the antilog of the noise-added data is taken. The second approach generates random noise by truncated normal distribution with mean equal to 1 and small variance, and then multiplies this noise by the original data. This method preserves the inter-record distances approximately. Therefore, in this case it is possible to reconstruct both aggregate distributions and some record-specific information as distance. This means that the multiplicative random perturbation method is suitable for many data mining applications. For example, in the work presented in [23] authors showed that this technique can be applied for the problem of classification. Moreover, the technique is suitable for the problem of privacy-preserving clustering [72, 74]. The work in [72] introduces a family of geometric data transformation methods (GDTMs) that distort confidential numerical attributes in order to meet privacy protection in clustering analysis. Oliveira et al. in [74] address the problem to guarantee privacy requirements while preserving valid clustering results. To achieve this dual goal, the authors introduce a novel spatial data transformation method called Rotation-Based Transformation (RBT). Multiplicative perturbations can also be used and for distributed privacy-preserving data mining as shown in [61]. The main techniques of multiplicative perturbation are based on the work presented in [52].

Differential Privacy

Differential privacy is a privacy notion introduced in [36] by Dwork. It is based on the fact that the privacy risks should not increase for a respondent as a result of occurring in a statistical database. Dwork in this work proposes to compare the risk with and without the record respondent's data in the published data. This privacy model, called $\epsilon-$*differential privacy*, assures a record owner that he/she may submit his/her personal

information to the database securely in the knowledge that nothing, or almost nothing, can be discovered from the database with his/her information that could not have been discovered without his/her information. Moreover, in [36] is formally proved that $\epsilon-$differential privacy can provide a guarantee against adversaries with arbitrary background knowledge. This strong guarantee is achieved by comparison with and without the record owner's data in the published data.

2.2.2 Anonymity by Indistinguishability

As said in the previous section randomization method has some weaknesses. The main problem is that it is not safe in case of attacks with prior knowledge. When the process of data transformation for privacy-preserving has not to be performed at data-collection time, it is better to apply methods that reduce the probability of record identification by public information. In literature three techniques have been proposed: k*-anonymity*, l*-diversity* and t*-closeness*. These techniques differ from the randomization methods as they are not data-independent.

k-Anonymity

One approach to privacy-preserving data publishing is *suppression* of some of the data values, while releasing the remaining data values exactly. However, suppressing just the identifying attributes is not enough to protect privacy because other kinds of attributes, that are available in public such as age, zip-code and sex can be used in order to accurately identify the records. This kind of attributes are known as *quasi-identifiers* [84]. In [83] it has been observed that for 87% of the population in the United States, the combination of Zip Code, Gender and Date of Birth corresponded to a unique person. This is called *record linkage*. In this work, authors proposed **k-anonymity** in order to avoid the record linkage. This approach became popular in privacy-preserving data publishing. The goal of k-anonymity is to guarantee that every individual object is hidden in a crowd of size k. A dataset satisfies the property of k-anonymity if each released record has at least $(k-1)$ other records also visible in the release whose values are indistinct over the quasi-identifiers. In k-anonymity techniques, methods such as *generalization* and *suppression* are usually employed to reduce the granularity of representation of quasi-identifiers. The method of *generalization* generalizes the attribute values to a range in order to reduce the granularity of representation. For instance, the city could be generalized to the region. Instead, the method of *suppression*, removes the value of an attribute. It is evident that these methods guarantee the privacy but also reduce the accuracy of applications on the transformed data.

The work proposed in [80] is based on the construction of tables that satisfy the k-anonymity property by using domain generalization hierarchies of the quasi-identifiers. The main problem of the k-anonymity is to find the minimum level of generalization that allows us to guarantees high privacy and a good data precision. Indeed, in [66], Meyerson and Williams showed that the problem of optimal k-anonymization is NP-hard. Fortunately, many efforts have been done in this field and many heuristic approaches have been designed as those in [59, 53]. LeFevre et al. in [59] propose a

framework to implement a model of k-anonymization, named full-domain generalization. They introduce a set of algorithms, called *Incognito* that allows us to compute a k-minimal generalization. This method generates all possible full-domain generalizations of a given table and thus, uses a bottom-up breadth-first search of the domain generalization hierarchy. In particular, it begins by checking if the single quasi-identifiers attributes satisfy the k-anonymity property and removing all the generalizations that do not satisfy it. In general, for each iteration i the *Incognito* algorithm performs these operations for the subset of quasi-identifiers of size i. Another algorithm, called *k-Optimize* is presented in [53] by Bayardo and Agrawal. This approach determines an optimal k-anonymization of a given dataset. This means that it perturbs the dataset as little as is necessary in order to obtain a dataset satisfying the k-anonymity property. In particular, authors try to solve the problem to find the power-set of a special alphabet of domain values. They propose a top-down search strategy, i.e., a search beginning from the most general to the more specific generalization. In order to reduce the search space *k-Optimize* uses pruning strategies. Another interesting work has been proposed in [88], where a bottom-up generalization approach for k-anonymity is presented. Instead, in [43] the authors introduced a method of top-down specialization for providing an anonymous dataset. Both these algorithms provide masked data that are still useful for building classification models.

The problem of k-anonymization can be seen as a search over a space of possible multi-dimensional solutions. Therefore, some work used heuristic search techniques such as genetic algorithms and simulated annealing [51, 90]. Unfortunately, by applying these approach the quality of the anonymized data is not guaranteed and often they require high computational times.

Aggarwal et al. proposed an approach based on clustering to implement the k-anonymity [6]. k-anonymity is also achievable by micro-aggregation, as shown in [30, 33]. Specifically, [33] shows the connection between masking methods for statistical disclosure control and privacy-preserving data mining. Moreover, it has been studied that some approximation algorithms guarantee the quality of the solution of this problem [66, 7]. In particular, in [7] the authors provide an $O(k)$-approximation algorithm for k-anonymity, that uses a graph representation. By using a notion of approximation authors try to minimize the cost of anonymization, due to the number of entries generalized and the degree of anonymization.

In literature, there exist also applications of the k-anonymity framework in order to preserve the privacy while publishing valid mining models. For example, in [13, 14, 15] the authors focused on the notion of individual privacy protection in frequent itemset mining and shift the concept of k-anonymity from source data to the extracted patterns.

Based on the definition of k-anonymity, new notions such as l-diversity [62] and t-closeness [60] have been proposed to provide improved privacy.

l-Diversity

In literature, there exist many techniques based on the k-anonymity notion. It is due to the fact that k-anonymity is a simple way to reduce the probability of record identification by public information. Unfortunately, the k-anonymity framework in some case can be vulnerable; in particular, it is not safe against homogeneity attack and back-

ground knowledge attack, that allow to infer the values of sensitive attributes. Suppose that we have a k-anonymous dataset containing a group of k entries with the same value for the sensitive attributes. In this case, although the data are k-anonymous, the value of the sensitive attributes can be easily inferred (Homogeneity Attack). Another problem happens when an attacker knows information useful to associate some quasi-identifiers with some sensitive attributes. In this case the attacker can reduce the number of possible value of the sensitive attributes (Background Knowledge Attack). In order to eliminate this weakness of the k-anonymity the technique of l-diversity was proposed [62]. The main aim is to maintain the diversity of sensitive attributes. In particular, the main idea of this method is that every group of individuals that can be isolated by an attacker should contain at least l *well-represented* values for a sensitive attribute. A number of different instantiations for the l-diversity definition are discussed in [62, 92].

t-Closeness

l-diversity is insufficient to prevent attack when the overall distribution is skewed. The attacker can know the global distribution of the attributes and use it to infer the value of sensitive attribute. In this case, the **t-closeness** method introduced in [60] is safe against this kind of attack. This technique requires that the distribution of a sensitive attribute in any equivalence class is close to the distribution of the attribute in the overall table. The distance between the two distributions should be no more than a threshold t [60].

2.2.3 Knowledge Hiding

This approach is also known as *sanitization*. The aim is to hide some kind of knowledge, such as rules or patterns, considered sensitive, which could be inferred from the published data. Clearly, in this context, the data owner wants to share the data and to hide sensitive knowledge.

This methodology has been used in literature in order to hide association rule, classification rule and sequential patterns.

In the context of *association rule hiding* there are a lot of approaches based on heuristics such as [12, 28, 81, 86, 73]; others instead are based on algebraic approaches as that proposed in [58] that tries to hide maximal sensitive patterns using a correlation matrix. An interesting approach is presented in [82], where authors introduced a border-based approach that uses the notion of *border*. The hiding process focuses on preserving the quality of the border, that reflects the quality of the sanitized database that is generated. In *classification rule hiding*, some rules are considered as sensitive and to protect such knowledge, a sanitization procedure needs to be enforced. We can partition existing approaches into two classes: suppression-based [22, 87, 24] and reconstruction-based schemes [70].

Finally, Abul et al. in [2] addressed the problem of hiding sensitive trajectory patterns from a database of moving objects. A similar technique is used in [3], where authors addressed first the problem of hiding patterns that are a simple sequence of symbols and then they extend the proposed framework to the case of sequential patterns according to the classical definition [9].

2.2.4 Distributed Privacy-Preserving Data Mining

In most distributed frameworks the participants would like to cooperate in order to compute global data mining models and aggregate results. Unfortunately, often they do not fully trust each other and would like to avoid the distribution of their data sets.

For addressing this problem, many distributed privacy-preserving data mining methods have been developed: in some of them the data sets are horizontally partitioned while in other they are vertically partitioned. In the first case, the individual records are distributed across multiple parties and each of them has the same set of attributes. In the second case, each party can have different attributes of the same records. Thus, the question addressed in this cases is how to compute the results without sharing the data in such a way that nothing is disclosed except the final result of the data mining result.

This problem is also addressed in cryptography in the field of *secure multi-party computation*. In general, the methods developed in this context allow to compute functions over inputs provided by multiple parties without sharing the inputs.

As an example, consider a function f of n arguments and n different parties. If each party has one of the n arguments it is necessary a *protocol* that allows to exchange information and to compute the function $f(x_1, \ldots, x_n)$, without compromising privacy. A set of methods are discussed in [34], specifically the authors describe how to transform data mining problems into secure multi-party computation problems. Clifton et al. in [25] present some methods for privacy-preserving computations that can be used to support important data mining tasks. These methods include the secure sum, the secure set union, the secure size of set intersection and the scalar product and can be used as data mining primitives for secure multi-party computation in case of horizontally and vertically partitioned datasets.

2.3 Statistical Disclosure Control

The aim of Statistical Disclosure Control (SDC) is to protect statistical data. In particular, it seeks to modify the data in such a way that they can be published and mined without compromising the privacy of individuals or entities occurring in the database. In other words, SDC seeks to provide safe techniques against linking attacks. Moreover, after the data protection, data analyses have to be possible and the results obtained should be the same or similar to the ones that would be obtained analyzing the data before the protection.

The youngest sub-discipline of SDC is the microdata protection. It aims at protecting static individual data, also called *microdata*. In this section we provide a survey of SDC methods for microdata, that are the most common data used for data mining.

A microdata set X can be viewed as a table or a file with n records. Each record related to a respondent contains m values associated to m attributes. The attributes can be classified in the following categories: *Identifiers*, *Quasi-identifiers*, *Confidential attributes* and *Non-confidential attributes*.

As stated above, the purpose of SDC is to prevent that confidential information can be linked to specific respondents, thus we will assume all the identifiers have been removed from the original microdata sets to be protected.

In the literature, several microdata disclosure protection methods have been proposed. Microdata protection methods can classified as follows: *masking techniques* and *synthetic data generation techniques*.

Masking techniques, usually, generate a modified version of the original microdata set, which are still suitable for statistical analysis although the respondents' privacy is guaranteed and can be divided in two sub-categories [89]: Non-perturbative and Perturbative. Synthetic data generation techniques, instead, produce new data that replace the original data and preserve their key statistical properties. The released synthetic data are not referred to any respondent. Hence, the release of this data cannot lead to re-identification. The techniques can be of two kinds: *fully synthetic techniques* and *partially synthetic techniques*.

2.3.1 Non-perturbative Masking Techniques

Non-perturbative techniques do not modify the original dataset; rather, these methods produce protected dataset by using suppressions or reductions of details in the original dataset. Some of these methods are suitable only for categorical data while other are suitable for both continuous and categorical data.

Non-perturbative methods include:

Sampling: this technique allow us to publish a sample of the original microdata [89]. Thus, the protected microdata contains only the data about a part of the whole population. This kind of methods are not suitable for continuous data.

Generalization: this method provides protected microdata by replacing the values of a given attribute by using more general values [80] defined in a *generalization hierarchy*.

Global Recoding: this method reduces the details in the microdata by substituting the value of some attributes with other values [31, 32]. For a continuous attribute, the method divides in disjoint intervals the domain of that attribute. Then it associates a label to each interval and finally, replaces the real attribute value with the label associated with the corresponding interval. For a categorical attribute, the method combines several categories in order to form new and less specific categories and then the new value is computed.

***Local Suppression*:** this method [80] suppresses the value of some individual or sensitive attributes, by replacing them with a missing value. In this way the possibility of analysis is limited.

2.3.2 Perturbative Masking Techniques

Perturbative techniques alter the microdata set before the publication for preserving statistical confidentiality. The statistics computed on the dataset protected by perturbation do not differ significantly from the ones computed on the original microdata set. In general, a perturbative approach modifies the microdata set by introducing new

combinations of values and making unique combinations of values in the original microdata set. In the following, we describe the main approaches belonging to this group of techniques:

Random Noise: these methods perturb microdata set by adding random noise following a given distribution [75]. Two kinds of additive noise exist in literature: *uncorrelated* and *correlated*. Notice that additive noise is usually not suitable to protect categorical data. As stated in Section 2.2.1 the Randomization techniques introduced by the data mining community come from the methods traditionally used in statistical disclose control described now.

Data Swapping: the basic idea is to switch a subset of attributes between selected pairs of records in the original database [39]. In this way, the data confidentiality is not compromised and the lower order frequency counts or marginals are preserved.

Rank Swapping: the idea is to rank the values of an attribute according to their ascending order [31]. Then, each value is swapped with another value guaranteeing that the swapped records are within a specified rank-distance of one another.

Resampling: this technique [31, 29] replaces the values of a sensitive continuous attribute with the average value computed over a given number of samples of the original population in the microdata set.

Rounding: this method replaces original values of attributes with rounded values. In order to replace the value of an attribute the technique defines a *rounding set*, that for example contains the multiples of a given base value. Then, it selects rounded values in this set.

RAM (Post RAndomized Method): this technique [56, 32] allows to perturb categorical value for one or more attributes by using a probabilistic mechanism, namely a Markov matrix.

Micro-Aggregation: this technique, described in [31], groups individual record into aggregates of dimension k. Next, given a group, its average value is computed and then it is published instead of individual values.

2.3.3 Synthetic Techniques

Two kind of synthetic techniques exist in literature: *fully synthetic* and *partially synthetic*. Fully synthetic techniques generate a set of data that is completely new. This means that the released data are referred to any respondent. Hence, no respondent can be re-identified. Different techniques exist that can be applied only on categorical or continuous data, or on both of them. Some methods belonging to this category are: *Cholesky decomposition* [65], *Bootstrap* [38], *Multiple imputation* [79], *Latin Hypercube Sampling* [41].

Partially synthetic techniques produce a dataset, where the original data and synthetic data are mixed. In literature, several techniques belonging to this category have

been proposed such as: *Hybrid masking* [27], *Information Preserving Statistical Obfuscation* [21], *Multiply Imputed Partially Synthetic Data* [42], and *Blank and Impute technique* [75].

2.4 Anonymity in Complex Data

Many research efforts have focused on privacy-preserving data mining and data publishing. Most of them, however, address the anonymity problems in the context of general tabular data, while relatively little work has addressed more complex forms of data in specific domains, although this kind of data is growing rapidly: examples include social networking data, spatio-temporal data, query log data, and more. The analysis of these data is very interesting as they are semantically rich: such richness makes such data also very difficult to anonymize, because the extra semantics may offer unexpected means to the attacker to link data to background knowledge. Traditional techniques used for tabular data sets cannot be directly applied, so typically the standard approaches must be adjusted appropriately. Privacy issues, privacy models and anonymization methods both for relational data and for complex data are widely discussed in [44]. A survey of techniques for anonymity of query log data is presented in [26]. In this work the author seeks to assess some anonymity techniques against three criteria: a) how well the technique protects privacy, b) how well the technique preserves the utility of the query logs, and c) how well the technique might be implemented as a user control. In [96] Zhou et al. propose a brief systematic review of the existing anonymity techniques for privacy preserving publishing of social network data. Another interesting work is presented in [63], where Malin introduces a computational method for the anonymization of a collection of person-specific DNA database sequences. The analysis of person-specific DNA sequences is important but poses serious challenges to the protection of the identities to which such sequences correspond.

Since one of most important and sensitive types of data used in LIFT project is spatio-temporal data, in this section we focus our discussion on this kind of data showing that in the last years some reasonable results have been obtained by solutions that consider the particular nature of these data. The increasing availability of spatio-temporal data is due to the diffusion of mobile devices (e.g., mobile phones, RFID devices and GPS devices) and of new applications, where the discovery of consumable, concise, and applicable knowledge is the key step. Clearly, in these applications privacy is a concern, since a pattern can reveal the behavior of group of few individuals compromising their privacy. Spatio-temporal data sets present a new challenge for the privacy-preserving data mining community because of their spatial and temporal characteristics. An interesting investigation on the various scientific and technological issues and open problems about this research field is presented in [46].

Standard approaches developed for tabular data do not work for spatio-temporal data sets. For example, randomization techniques, discussed above, which modify a dataset to guarantee respondents' privacy while preserving data utility for analyses, are not applicable on spatio-temporal data, due to their particular nature. Therefore, alternative solutions have been suggested: some of them belong to the category of

confusion-based algorithm others belong to the category of approaches of *k-anonymity for location position collection*. All these techniques try to guarantee location privacy for trajectories.

The approaches in [48, 54, 55, 35] belong to the first category and provide confusion/obfuscation algorithm to prevent an attacker from tracking a complete user trajectory. The main idea is to modify true trajectories or generate fake trajectories in order to confuse the attacker. In [18, 17, 47, 20] authors presented techniques belonging to the second category. The main aim of these techniques is to preserve the anonymity of a user obscuring his route. They use the notion of k-anonymity adapted for the spatio-temporal context.

k-anonymity is the most popular method for the anonymization of spatio-temporal data. It is often used both in the works on privacy issues in location-based services (LBSs) [19, 64] and in the works of anonymity of trajectories [1, 71, 93]. In the work presented in [1], the authors study the problem of privacy-preserving publishing of moving object database. They propose the notion of (k, δ)-anonymity for moving objects databases, where δ represents the possible location imprecision. In particular, this is a novel concept of k-anonymity based on co-localization that exploits the inherent uncertainty of the moving objects whereabouts. In this work authors also propose an approach, called *Never Walk Alone*, for obtaining a (k, δ)-anonymous moving objects database. The method is based on trajectory clustering and spatial translation. In [71] Nergiz et al. address privacy issues regarding the identification of individuals in static trajectory datasets. They provide privacy protection by: (1) first enforcing k-anonymity, meaning every released information refers to at least k users/trajectories, (2) then reconstructing randomly a representation of the original dataset from the anonymization. Yarovoy et al. in [93] study problem of k-anonymization of moving object databases for the purpose of their publication. They observe the fact that different objects in this context may have different quasi-identifiers ans so, anonymization groups associated with different objects may not be disjoint. Therefore, a novel notion of k-anonymity based on spatial generalization is provided. In this work, authors propose two approaches that generate anonymity groups satisfying the novel notion of k-anonymity. These approaches are called *Extreme Union* and *Symmetric Anonymization*.

Lastly, we mention the very recent work [85], where Terrovitis and Mamoulis. This work is based on the assumption that different attackers know different and disjoint portions of the trajectories and the data publisher knows the attacker knowledge. So, the proposed solution is to suppress all the dangerous observations in the database.

The common result obtained by the above research works on the problem of the privacy-preserving publication of complex data is that finding an acceptable trade-off between data privacy on one side and data utility on the other side is hard and that no general method exists, capable of both dealing with "generic personal data" and preserving "generic analytical results". Usually, the proposed approaches guarantee the privacy requirements but hardly generate anonymous datasets with acceptable data quality: the data transformation obstructs the knowledge discovery opportunities of data mining technologies. This problem is due to the fact that the anonymization frameworks are designed without any assumption about the target analytical questions that are to be answered with the data. This point is fundamental because taking into ac-

count the possible target analysis to be applied to the transformed data means designing a transformation process capable to preserve some data properties that are necessary to preserve the results obtained by specific analytical and/or mining tasks. To this scope, [67] propose the *privacy by design* paradigm that promises a quality leap in the conflict between data protection and data utility.

2.5 Privacy by Design

This section aims to describe the *Privacy by Design* paradigm introduced in [67]. This paradigm reflects our general idea to develop technological frameworks to counter the threats of undesirable, unlawful effects of privacy violation, without obstructing the knowledge discovery opportunities of data mining technologies. The main idea is to inscribe privacy protection into the knowledge discovery technology by design, so that the analysis incorporates the relevant privacy requirements from the very start. Here, we evoke the concept of *Privacy by Design* coined in the '90s by Ann Cavoukian, the Information and Privacy Commissioner of Ontario, Canada. In brief, *Privacy by Design* refers to the philosophy and approach of embedding privacy into the design, operation and management of information processing technologies and systems. This paradigm promises a quality leap in the conflict between data protection and data utility. Here, the articulation of the general "by design" principle in the domain of knowledge discovery is that higher protection and quality can be better achieved in a goal-oriented approach. In such an approach, the knowledge discovery process (including the data collection itself) is designed with assumptions about:

(a) the (sensitive) personal data that are the subject of the analysis;

(b) the attack model, i.e., the knowledge and purpose of a malicious party that has an interest in discovering the sensitive data of certain individuals;

(c) the target analytical questions that are to be answered with the data.

These assumptions are fundamental for the design of a privacy-preserving framework for various reasons. First of all, the techniques for privacy preservation strongly depend on the nature of the data that we want to protect. For example, many proposed methods are suitable for continuous variables but not for categorical variables (or the other way round), while other techniques employed to anonymize sequential data such as clinical data or tabular data are not appropriate for moving object datasets. Clearly, different forms of data have different properties that must be considered during the transformation process.

Second, a valid framework for privacy protection has to define the background knowledge of the adversary, that strongly depends on the context and on the kind of data. So, an attack model, based on the background knowledge of the attacker, has to be formalized and a specific countermeasure associated to that attack model has to be defined in terms of the properties of the data to be protected. The definition of a suitable attack model is very important in this context. Different assumptions on the background knowledge of an attacker entail different defense strategies. Indeed, it is

clear that when the assumption on the background knowledge changes the transformation approach to be adopted also changes significantly. Consider, for example, that an attacker gains the access to a spatio-temporal dataset and that he/she knows some spatio-temporal points belonging to some trajectory of an individual. Two cases are possible: (i) the attacker knows the exact points or (ii) the attacker knows these points with a given uncertainty threshold. The attacker can try to re-identify the respondent by using his/her knowledge and by observing the protected database. Specifically, he/she should generate all the possible candidate trajectories by using the background knowledge as constraints. Clearly, the defense strategy that it is necessary to use in the case (ii) might be unsuitable for the case (i), because the assumption (ii) is weaker than the assumption (i). This does not mean that assumption (ii) is not valid, as it can be adequate for particular situations where (i) is unrealistically strong. In general, it is natural for different situations to require different privacy requirements and that one person can have different privacy expectations than another. For example the perception of the privacy for a famous actor is surely different from that of a common citizen, since most of the information about the actor's life is already made public because of he nature of the job. Clearly, the assumption that the background knowledge of an adversary depends on the context allows to realize frameworks that guarantee *reasonable* levels of privacy according to the privacy expectation.

Finally, a privacy-preserving strategy should find an acceptable trade-off between data privacy on one side and data utility on the other side. In order to reach this goal it is fundamental to take into account during the design of the framework the analytical questions that are to be answered with the transformed data. This means designing a transformation process capable to preserve some data properties that are necessary to preserve the results obtained by specific analytical and/or mining tasks.

Under the above assumptions, it is conceivable to design a privacy-preserving analytical process able to:

1. transform the source data into an anonymous or obfuscated version with a quantifiable privacy guarantee - i.e., the probability that the malicious attack fails (measured, e.g., as the probability of re-identification);

2. guarantee that the target analytical questions can be answered correctly, within a quantifiable approximation that specifies the data utility, using the transformed data instead of the original ones.

2.6 Privacy by Design for Data Publishing

The *privacy by design* paradigm has been used for the design of privacy-preserving frameworks for data publishing obtaining good results in terms of privacy protection and data utility. In that context, the aim is to publish databases providing privacy guarantees and assuring that the data can be used for some specific analysis. Clearly, as described in the previous section, for the design of a valid privacy-preserving framework it is important to take into account the kind of data to be transformed and the type of attack to be countered.

2.6.1 Privacy Models for Trajectory and Sequential Data

In this section we present some privacy models proposed in literature for the publishing of trajectory data and sequence data. So, we describe some attacks that an intruder who gains access to a published database of sequences or trajectories can conduct in order to make inferences, also on the basis of the background knowledge that (s)he possesses. We generically refer to this agent as an attacker.

Sequence Linking Attack

The attack model we describe in this section is presented in [76, 77, 67].

Before describing this attack model we introduce some useful notation. Let $\mathcal{I} = \{i_1, i_2, \ldots, i_n\}$ denote a set of items (e.g., events, actions, spatial locations or regions). Here, we consider the case of sequence databases of the form $\mathcal{D} = \{S_1, S_2, \ldots, S_N\}$, where each sequence $S = i_1 i_2 \ldots i_h$ ($i_j \in \mathcal{I}$) is an ordered list of single items; an item can occur multiple times in a sequence.

So, given a published sequence database $\mathcal{D}$ an intruder who gains access to it can conduct attacks in order to make inferences, also on the basis of the background knowledge that (s)he possesses. In particular, we refer to the *linking attack model*, i.e., the ability to link the released data to other external information, which enables the re-identification of (some of) the respondents associated with the data. In relational data, linking is made possible by *quasi-identifiers*, i.e., attributes that, in combination, can uniquely identify individuals, such as birth date and gender. The remaining attributes represent the private respondent's information, that may be violated by the linking attack. In privacy-preserving data publishing techniques, such as k-anonymity, the precise goal is to find countermeasures to this attack, and to release person-specific data in such a way that the ability to link to other information using the quasi-identifier(s) is limited.

In the case of sequential (person-specific) data, where each record is a temporal sequence of events which occurred to a specific person, the above dichotomy of attributes into quasi-identifiers (QI) and private information (PI) does not hold any longer: here, a (sub)sequence of events can play both the role of QI and the role of PI. To see this point, consider the case where sequences represent trajectories, i.e., lists of locations visited by an individual in the given order: the attacker may know a sequence of locations visited by a specific person P: e.g., by shadowing P for some time, the attacker may learn that P was in the shopping mall, then in the park, and then at the train station, represented by the sequence $\langle mall, park, station \rangle$. The attacker could employ this sequence to retrieve the complete trajectory of the P in the released dataset: this attempt would succeed, provided that the attacker knows that P's sequence is actually present in the dataset, if the known sequence $\langle mall, park, station \rangle$ is compatible with (i.e., is a subsequence of) just one sequence in the dataset. In this example of a linking attack in the sequence domain, the subsequence known by the attacker serves as QI, while the entire sequence is the PI that is disclosed after the re-identification of the respondent. Clearly, as the example suggests, it is rather difficult to distinguish QI and PI: in principle, any specific location can be the theater of a shadowing actions by a spy, and therefore any possible sequence (of locations, in this example) can be used as a QI,

i.e., as a means for re-identification. Put another way, distinguishing between QI and PI among the elements of a sequence, being them locations or events, means putting artificial limits on the attacker's background knowledge; on the contrary, in privacy and security research it is necessary to have assumptions on the attacker's knowledge that are as liberal as possible, in order to achieve maximal protection.

As a consequence of this discussion, we make the conservative assumption that *any sequence that can be linked to a small number of individuals is a potentially dangerous QI and a potentially sensitive PI*; then, we study an anonymity model that tries to achieve the maximal protection possible under this challenging assumption. The crucial point in defining the sequence linking attack lies exactly in the definition of QI and PI, which is formalized by the concept of *harmful sequence*, parametric with respect to an anonymity threshold k.

Definition 2.6.1 (k-Harmful Sequence)**.** *Given a sequence dataset $\mathcal{D}$ and an anonymity threshold k, a sequence T is* k-harmful *(in $\mathcal{D}$) iff $0 < supp_{\mathcal{D}}(T) < k$.*

In other words, a sequence is k-harmful if it is a subsequence of a number of sequences in $\mathcal{D}$ smaller than k and greater than 0. Essentially, harmful sequences are potentially dangerous QIs because they occur only a few times in the dataset (but at least once): thus, a harmful sequence can be used to select a few specific complete sequences in the dataset. Moreover, each harmful sequence reveals information pertaining to a small (but not empty) set of persons, hence information that is private in the sense that it reveals a specific, unusual behavior, which potentially violates the right to privacy of a few individuals that follow a path off the crowd (perhaps revealing personal preferences, habits, etc.) Conversely, non-harmful sequences are not considered dangerous, neither as QI nor as PI: a non-harmful sequence either does not occur in the dataset (and therefore does not help the attacker) or occurs so many times that (i) it is not useful as a QI, as it is compatible with too many subjects, and (ii) it is not useful as PI, as it reveals a sequential behavior common to many people. We now formalize the privacy model. So, first of all we introduce our assumptions about the additional knowledge used by the adversary for the attack.

Definition 2.6.2 (Adversary Knowledge)**.** *The attacker has access to the anonymized dataset $\mathcal{D}^*$ and knows: (i) the details of the scheme used to anonymize the data, (ii) the fact that respondent U is present in $\mathcal{D}$, and (iii) a (QI) sequence T relative to U.*

Then, we formalize the sequence linking attack, based on the above definition.

Definition 2.6.3 (Sequence Linking Attack)**.** *Given a published sequence dataset $\mathcal{D}$ where each sequence is uniquely associated with a de-identified respondent, the attacker tries to identify the sequence in $\mathcal{D}$ associated with a given respondent U, based on the additional knowledge introduced in Definition 2.6.2. We denote by $prob_{\mathcal{D}}(T)$ the probability that the sequence linking attack with a QI sequence T succeeds (over $\mathcal{D}$).*

From a data protection perspective, we aim at controlling the probability $prob_{\mathcal{D}}(T)$, for any possible QI sequence T. The linking attack can be performed by using either a harmful or a non-harmful sequence. Clearly, harmful sequences are dangerous because the attacker has a high probability of uniquely identifying the entire sequence of a respondent.

Trajectory Linking Attack

The attack model we describe in this section is presented in [68, 67].

A moving object dataset is a collection of spatio-temporal sequences $\mathcal{D} = \{T_1, T_2, \ldots, T_N\}$; each element $T_i \in \mathcal{D}$ is a sequence of spatial points with a timestamp element which we call *trajectory* in the remaining part of the chapter. In the following we introduce the formal definition of *trajectory* and *sub-trajectory*.

Definition 2.6.4 (Trajectory)**.** *A Trajectory or spatio-temporal sequence is a sequence of triples* $T =< x_1, y_1, t_1 >, \ldots, < x_n, y_n, t_n >$*, where* t_i $(i = 1 \ldots n)$ *denotes a timestamp such that* $\forall_{1<i<n}\ t_i < t_{i+1}$ *and* (x_i, y_i) *are points in* $\mathbf{R}^2$*.*

Intuitively, each triple $< x_i, y_i, t_i >$ indicates that the object is in the position (x_i, y_i) at time t_i.

Definition 2.6.5 (Sub-Trajectory)**.** *Let* $T =< x_1, y_1, t_1 >, \ldots, < x_n, y_n, t_n >$ *be a trajectory. A trajectory* $S =< x'_1, y'_1, t'_1 >, \ldots, < x'_m, y'_m, t'_m >$ *is a sub-trajectory of* T *or is contained in* T *(*$S \preceq T$*) if there exist integers* $1 \leq i_1 < \ldots < i_m \leq n$ *such that* $\forall 1 \leq j \leq m < x'_j, y'_j, t'_j >=< x_{i_j}, y_{i_j}, t_{i_j} >$*.*

We refer to the number of trajectories in $\mathcal{D}$ containing a sub-trajectory S as *support of* S and denote it by $supp_{\mathcal{D}}(S)$, more formally $supp_{\mathcal{D}}(S) = |\{T \in \mathcal{D} | S \preceq T\}|$.

The dataset owner applies an anonymization function to transform $\mathcal{D}$ into $\mathcal{D}^*$, the anonymized dataset.

Our anonymization scheme is based on:

(a) generating a partition in areas of the territory covered by the trajectories;

(b) applying a function for the spatial generalization to all the trajectories in order to transform them into sequences of spatial points that are centroids of specific areas;

(c) transforming the generalized trajectories to guarantee privacy.

We use g to denote the function that applies the spatial generalization to a trajectory. Given a trajectory $T \in \mathcal{D}$, this function generates the generalized trajectory $g(T)$, i.e. the centroid sequence of areas crossed by T.

Definition 2.6.6 (Generalized Trajectory)**.** *Let* $T =< x_1, y_1, t_1 >, \ldots, < x_n, y_n, t_n >$ *a trajectory. A generalized version of* T *is a sequence of pairs* $T_g =< x_{c_1}, y_{c_1} > , \ldots, < x_{c_m}, y_{c_m} >$ *with* $m <= n$ *where each* x_{c_i}, y_{c_i} *is the centroid of an area crossed by* T*.*

Note that, the function $g(.)$ drops the time component from the trajectory that becomes a sequence of generalized spatial points (centroids), where the order of the points in the sequence corresponds to the temporal order in which the points are visited: the point in position i is visited before the point in position $i + 1$.

Definition 2.6.7 (Generalized Sub-Trajectory)**.** *Let* $T_g =< x_1, y_1 >, \ldots, < x_n, y_n >$ *be a generalized trajectory. A generalized trajectory* $S_g =< x'_1, y'_1 >, \ldots, < x'_m, y'_m >$ *is a* generalized sub-trajectory *of* T_g *or* is contained *in* T_g *if there exist integers* $1 < i_1 < \ldots < i_m < n$ *such that* $\forall 1 \leq j \leq m < x'_j, y'_j >=< x_{i_j}, y_{i_j} >$.

We refer to the number of generalized trajectories in a dataset $\mathcal{D}^{\mathcal{G}}$ containing a sub-trajectory S_g as *support of* S_g and denote it by $supp_{\mathcal{D}^{\mathcal{G}}}(S_g)$, where $supp_{\mathcal{D}^{\mathcal{G}}}(S_g) = \left|\{T_g \in \mathcal{D}^{\mathcal{G}} | S_g \preceq T_g\}\right|$.

An attacker may know a sub-trajectory of the trajectory of some specific person, and could use this information to retrieve the complete trajectory of the same person in the released dataset. Thus, we assume the following adversary knowledge.

Definition 2.6.8 (Adversary Knowledge)**.** *The attacker has access to the anonymized dataset* $\mathcal{D}^*$ *and knows: (a) the details of the scheme used to anonymize the data, (b) the fact that a given user* U *is in the dataset* $\mathcal{D}$ *and (c) a sub-trajectory* S *relative to* U.

The ability to link the published data to external information, which enables various respondents associated with the data to be re-identified is known as a *trajectory linking attack model*.

The movement data have a sequential nature and are a particular case of sequence data discussed in Section 2.6.1. As already discussed in the previous section, in the case of data with sequential nature without any kind of additional semantic information on the data it is hard to make a clear distinction between quasi-identifiers (QI) and private information (PI). Thus, as in the case of general sequence data, in the case of spatio-temporal data a sub-trajectory can play both the role of QI and PI. In a linking attack conducted by a sub-trajectory known by the attacker the entire trajectory is the PI that is disclosed after the re-identification of the respondent, while the sub-trajectory serves as QI.

Here, we consider the following attack:

Definition 2.6.9 (Attack Model)**.** *Given the anonymized dataset* $\mathcal{D}^*$ *and a sub-trajectory* S *relative to a user* U*, the attacker:* (i) *generates the partition of the territory starting from the trajectories in* $\mathcal{D}^*$*;* (ii) *computes* $g(S)$ *generating the sequence of centroids of the areas containing the points of* S*;* (iii) *constructs a set of candidate trajectories in* $\mathcal{D}^*$ *containing the generalized sub-trajectory* $g(S)$ *and tries to identify the whole trajectory relative to* U.
The probability of identifying the whole trajectory by a sub-trajectory S *is denoted by* $prob(S)$.

From the point of view of data protection, minimizing the probabilities of re-identification is desirable. Intuitively, the set of candidate trajectories corresponding to a given sub-trajectory S should be as large as possible. A good solution would be to minimize the probabilities of re-identification and maximize data utility by minimizing the transformation of the original data. We propose a k-anonymity setting as a compromise. The general idea is to control the probability of the re-identification of any trajectory to below the threshold $\frac{1}{k}$ chosen by the data owner. Thus, our goal is to find an anonymous version of the original dataset $\mathcal{D}$, such that, on the one hand, it is still

useful for analysis, when published, and on the other, a suitable version of k-anonymity is satisfied.

The crucial point of our attack model is that it can be performed by using any sub-trajectory in $\mathcal{D}$: a sub-trajectory occurring only a few times in the dataset (but at least once) enables a few specific complete trajectories to be selected, and thus the probability that the sequence linking attack succeeds is very high. On the other hand, a sub-trajectory occurring so many times that it is compatible with too many subjects reduces the probability of a successful attack.

Attack by Background Network on Trajectory Data

The attack model we describe in this section is presented in [4].

We focus on trajectories of objects moving over a background (road) network, which is modeled as a directed graph.

Definition 2.6.10 (Background Network)**.** *The background road network is a directed labeled graph $BN = (V, E, l)$, where V is a set of vertices, each vertex $v_i = (x_i, y_i)$ is a point in R^2; $E \subseteq V \times V$ is a set of edges, where each edge (v_i, v_j) is the straight line going from vertex v_i to vertex v_j; and $l : E \rightarrow R$ is a labeling function that assigns to an edge a label representing the minimum time necessary to cover the edge (i.e., its length over the maximum speed allowed on it).*

The dataset owner applies an anonymization function to transform $\mathcal{D}$ into $\mathcal{D}^*$, the anonymized dataset. This anonymization function aims at solving the *Trajectory Pattern Hiding Problem* that is formalized as follows. Given a set of sensitive trajectory patterns $P_h = \{P_1, \ldots, P_n\}$ that must be hidden from a database $\mathcal{D}$ consistent with BN. Given a disclosure threshold ψ, the Trajectory Pattern Hiding Problem requires to transform $\mathcal{D}$ in a database $\mathcal{D}^*$ such that:

1) $\mathcal{D}^*$ is still consistent with BN;

2) $\forall\ P_i \in P_h,\ sup_{[\mathcal{D}^*,\tau]}(P_i) \leq \psi$;

3) the difference between $\mathcal{D}$ and $\mathcal{D}^*$ is minimized.

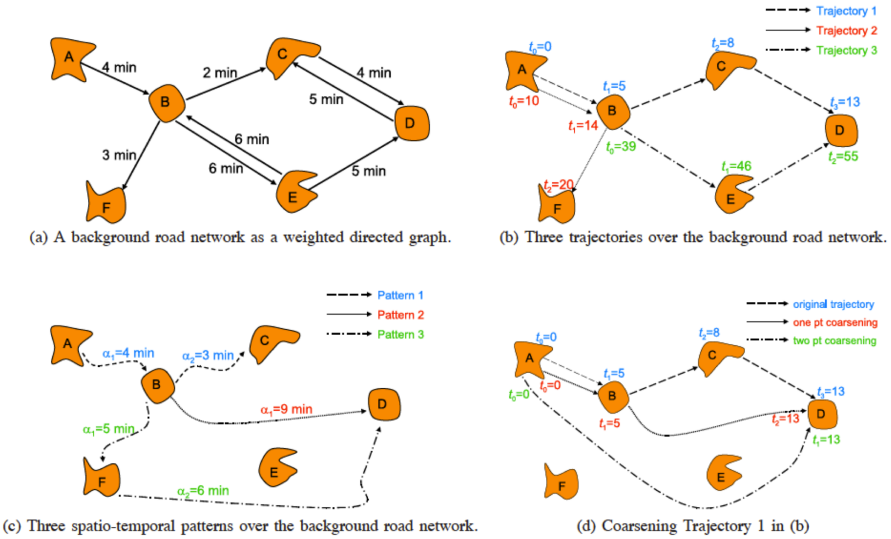
The problem requires to sanitize the input database $\mathcal{D}$ in such a way that a set of sensitive patterns P is hidden while the most of the information in $\mathcal{D}$ is maintained. The resulting database $\mathcal{D}^*$, that is the released one, must be consistent with the background road network.

The first requirement of this problem asks to avoid creating unreal trajectories in the sanitization process, since the road network BN is a publicly available knowledge and thus unreal trajectories could be easily identified. The second requirement asks all sensitive patterns to be hidden in $\mathcal{D}^*$, i.e., they must have a support not more than the given disclosure threshold ψ. Finally, the third requirement asks to keep $\mathcal{D}^*$ as similar as possible to $\mathcal{D}$. This is a very general definition which does not say how the sanitization is actually performed.

Consider a temporal sequence of vertexes representing the trajectory T, and suppose that it is sanitized by suppressing the point (v_i, t_i) from the subsequence

$$(v_{i-1}, t_{i-1}), (v_i, t_i), (v_{i+1}, t_{i+1}).$$

If there exists only one path from v_{i-1} to v_{i+1} time-consistent with BN, then the attacker can easily infer the suppressed point. This kind of inference channels can help the attacker reconstructing (even only partially) the original data, and this in turn can cause some of the sensitive patterns $P \in P_h$ to be disclosed. We name this kind of inference *Attack by Background Knowledge* or *Attack by Lack of Alternative Paths*. Instead, when there are many alternative paths from v_{i1} to v_{i+1} time consistent with BN, then the task of reconstructing the missing part and discovering the hidden pattern is not trivial. Obviously, the larger is the number of possible alternative paths, the more secure is the provided sanitization. This leads to the definition of a interesting property that our sanitized data should exhibit. The attack by background network is obvious in Figure 2.2(d). Suppose only Pattern 1 (in Figure 2.2(c)) is sensitive and Trajectory 1 needs to be sanitized. The one point coarsening seem to remove the sensitive knowledge from the trajectory, and thus its disclosure is safe. However, an attacker knowing the background road network in Figure 2.2(a) can easily deduce that it is impossible to get from point B to point $\mathcal{D}$ in 8 minutes taking E as a midpoint. So, the attacker is hundred percent sure that the trajectory followed C as the midpoint, thus revealing the sensitive knowledge by reconstruction. Note that even the two point coarsening is a pseudo-hiding in this case. But publishing only the first two (out of four) spatio-temporal points, another coarsening, does not disclose the sensitive knowledge.



(a) A background road network as a weighted directed graph.

(b) Three trajectories over the background road network.

(c) Three spatio-temporal patterns over the background road network.

(d) Coarsening Trajectory 1 in (b)

Figure 2.2: Spatio-temporal trajectories and patterns

2.6.2 Privacy Model for Semantich-rich Sequence Data

The progress on device technology, data analysis and mining are creating entirely new forms of data, that are always more complex and richer of semantic information, i.e.,

additional information describing specific data properties. Naturally, this information can be used during the mining process to extract a more interesting and richer knowledge. *Semantic-rich sequence data* are data with a sequential nature for which specific descriptions of the items and of the events are available. A concrete example of this kind of data is represented by the so-called *semantic trajectories*, a new form of mobility data with far richer semantic information attached to the traces of personal mobility. In other words, we are rapidly moving from raw trajectories, i.e., sequences of time-stamped generic points sampled during the movement of a sensed device, to what sequences of stops and moves of a person during her/his movements, where each location of stop can be attached to some semantics, or purpose - either by explicit sensing or by inference.

We argue that these new data with semantic information attached poses even greater privacy threats w.r.t. simple sequence data. We used the privacy by design paradigm to design a privacy model to face this challenging problem [69].

The first problem introduced by this semantic information is that an attacker can use it to infer further private personal information about a user. As an example consider the semantic trajectory data, from the fact that a person has stopped in a certain sensitive location, e.g., an oncology clinic, an attacker can derive private personal information of the health of such person. So, in this context, an item of a sequence is *sensitive* if it allows to infer personal sensitive information of an individual.

In [69] we essentially devises a privacy model for semantic trajectories, with reference to a background knowledge defining which are the *sensitive* and *non-sensitive* places in a specific application. The background knowledge is represented through a specific taxonomy, describing sensitive and non-sensitive places at different levels of abstraction (e.g., a tourist landmark, a museum, the Louvre museum; a health-related service, a hospital, the Children's Hospital).

An intruder who gains access to dataset of semantic trajectories $\mathcal{ST}^*$ may possess some background knowledge allowing to conduct attacks making inferences on the dataset.

Definition 2.6.11 (Adversary Knowledge)**.** *The attacker has access to the generalized dataset $\mathcal{ST}^*$ and knows: (a) the algorithm used to anonymize the data, (b) the privacy place taxonomy P-Tax , (c) that a given user is in the dataset and (d) a quasi-identifier place sequence S_Q visited by the given user.*

What is the information that has to remain private? In our model, we keep private all the sensitive places visited by a given user. Therefore, the attack model considers the ability to link the released data to other external information enabling to infer visited sensitive places.

Definition 2.6.12 (Attack Model)**.** *The attacker, given a published semantic trajectory dataset $\mathcal{ST}^*$ where each trajectory is uniquely associated to a de-identified respondent, tries to identify the semantic trajectory in $\mathcal{ST}^*$ associated to a given respondent U, based on the additional knowledge introduced in Definition 2.6.11. The attacker, given the quasi-identifier sequence S_Q constructs a set of candidate semantic trajectories in $\mathcal{ST}^*$ containing S_Q and tries to infer the sensitive leaf places related to U. We denote by $Prob(S_Q, S)$ the probability that, given a quasi-identifier place sequence*

S_Q related to a user U, the attacker infers his/her set of sensitive places S which are the leaves of the taxonomy $PTax$.

From a data protection perspective, we aim at controlling the probability $Prob(S_Q, S)$. To prevent the attack defined above we propose to release a *c-safe* dataset.

Definition 2.6.13 (C-Safety)**.** *The dataset $\mathcal{ST}$ is defined c-safe with respect to the place set Q if for every quasi-identifier place sequence S_Q, we have that for each set of sensitive places S the $Prob(S_Q, S) \leq c$ with $c \in [0, 1]$.*

Chapter 3

Privacy by Design for LIFT-based Systems

In this chapter we define the general attack model that we consider in the LIFT-based systems and we describe how we can apply the *privacy by design* paradigm for obtaining privacy guarantees in this context.

3.1 Privacy Requirements for LIFT-based Systems

We consider a distributed-computing environment, composed of a collection of n remote sites (nodes) and a designated coordinator site. Streams of data arrive continuously at remote sites, while the coordinator site is responsible for processing a global function through local computations in the nodes. Each node can then be assigned a safe zone for its local data-stream values that can offer guarantees for the value of the global function over the entire collection of nodes.

As discussed in Section 2.5 in order to apply the *privacy by design* paradigm and thus, designing a valid privacy-preserving frameworks it is important to take into account the kind of data to be transformed and the type of attack to be countered. To this scope we have to answer the following important questions:

(1) Who may be an attacker in this context?

(2) Which data does the attacker access?

(3) Which background knowledge does the attacker possess to infer new and sensitive information?

(4) May the communicated data streams violate the user privacy?

3.1.1 Attacker

We refer to any possible agent who gains access to the data that a node communicates to the coordinator and who can conduct an attack in order to make inferences also on

the basis of the background knowledge that he possesses as *attacker*. In addition to a third party we consider the coordinator and in some cases the local nodes as an untrusted party and thus an attacker. The coordinator corresponds to the *data recipient* in privacy-preserving data publishing context. Indeed, in that context it is typical to consider the data recipient untrusted because even if it is a trustworthy entity, however, it is difficult to guarantee that all staff in this entity is trustworthy as well. This assumption makes the solutions based on encryption and cryptographic approaches, in which only authorized and trustworthy recipients are given the private key for accessing the cleartext, useless. While the coordinator is untrusted in all settings, we will consider nodes as trusted in some settings. If the data collection takes place at the user himself (e.g. movements are monitored via his mobile phone) we consider the node as trusted. If the data is collected outside of the user (e.g. by a Bluetooth antenna) we consider the node as untrusted. In order to ensure privacy in both settings we require an *open privacy* policy at the nodes. Open privacy means that algorithms and software components are available to the public so that their behavior can be verified. In the first case this is necessary because the user is not the author of the application that is provided for his mobile device. In the second case this is necessary because the nodes record possibly privacy sensitive information of which a secure processing has to be ensured.

We do not consider attacks from intruders that access the data during the communications by sniffing because these can be avoided by cryptography techniques.

The data streams communicated to the coordinator could provide sensitive information to an attacker who by using external information could learn more information about the individual and private sphere of a specific user. A privacy-preserving framework should avoid that an attacker gaining access to the data can enrich his knowledge violating the individual privacy of a person. Clearly, the sensitivity of the communicated information depends on the application. In particular, it is clear that in different applications the type of data that each node has to communicate to the coordinator could have very different characteristics and it is strongly related to the kind of local/global function that has to be computed in the system.

3.1.2 Attack Model

We consider attackers with the ability to link data to external information, which enables various respondents associated with the data to be re-identified. This attack model is known as a *linking attack model*. In relational data, linking is made possible by *quasi-identifiers*, i.e., attributes that, in combination, can uniquely identify individuals, such as birth date and gender. The remaining attributes represent the private respondent's information, that may be violated by the linking attack. In some complex context such as trajectory data, the above dichotomy of attributes into quasi-identifiers (QI) and private information (PI) can be impossible and in that cases the information known by the attacker is considered as QI while the new information that he may infer represents the PI to be protected. This is properly our case: we assume that the attacker knows some information about a specific user and can use this information as quasi-identifier. We further assume that the attacker knows that the user is one of the nodes in the system, and seeks to identify the user's communications and so the sensitive information about him.

Definition 3.1.1 (Attack Model)**.** *Given the background knowledge $\mathcal{BN}$ relative to a given user X, the attacker who gains access to one or several data streams communicated by different users can construct a set of candidates who are compatible with $\mathcal{BN}$ and tries to identify the user X in this set of candidates and to infer new sensitive information about him.*
The probability of identifying the user and inferring new information is denoted by $prob(\mathcal{BN})$.

The success of this attack depends on the background knowledge that the intruder possesses as he uses this information for the re-identification. As explained in Section 2.5 the background knowledge strongly depends on the context and on the kind of data to be protected. Therefore, to provide a formal definition of the adversary knowledge and to define how the attacker can use it to conduct a linking attack we need to know the specific application where we want to provide privacy guarantees. This means that we need to know:

(i) the form of data that each node has to communicate to the coordinator

(ii) the kind of function that has to be computed in the system.

Without this information we cannot assume anything about the knowledge adversary and as a consequence we cannot define the form of attack that an intruder can conduct in the LIFT-based system. For example, if our framework has to process global functions for data mobility analysis and the nodes has to communicate information about user positions the background knowledge could be composed of a set of positions visited by a specific person. In contrast, if our framework has process functions about the analysis of query logs and the nodes communicate information about user queries then the background knowledge could be composed of a set of queries related to a specific user. It is evident that changing the type of data can change the type of computed function and as a consequence the adversary to conduct the attacker and to generate the set of candidates has to execute a different computation.

If we have a formal definition of both background knowledge $\mathcal{BN}$ and attack model it is possible to design a suitable privacy-preserving framework to counter that kind of attack. In Chapter 4 we will show some application scenarios where we modeled the $\mathcal{BN}$ and customized the attack defined above.

3.2 Properties of a Countermeasure

In this section we provide some general properties of a suitable countermeasure. A privacy-preserving approach has to keep under control the natural trade-off between privacy protection, data quality and quality of system performance.

Therefore, a valid privacy-preserving method has to provide:

1. a *quantifiable privacy guarantee* - i.e., the probability that the malicious attack fails (measured, e.g., as the probability of re-identification);

2. a *quantifiable data utility* - i.e., the target analytical questions have to be answered correctly, within a *quantifiable approximation* that specifies the data utility, using the transformed data instead of the original ones;

3. a *performance guarantee*: the privacy-preserving technique could affect the performance of the overall systems for example in terms of number of communications and as a consequence it has to keep this degradation of the performance as low as possible.

For guaranteeing the privacy protection it is necessary to transform the data to be communicated to the coordinator and this transformation will introduce some noise which could affect the result of the global computation. From the point of view of data protection, a data transformation that minimizes the probability of re-identification $prob(\mathcal{BN})$ is desirable. Intuitively, the set of candidates corresponding to a given background knowledge $\mathcal{BN}$ should be as large as possible. Clearly, a solution of this type will lead to a high data quality loss and as a consequence an error in the computation result. Therefore the desirable solution is a transformation able to minimize the probabilities of re-identification and maximize data utility by minimizing the transformation of the original data. So, the aim is to guarantee the data protection while avoiding to completely destroy the analytical result; in other word the data transformation has to preserve the analytical result within a certain quantifiable approximation.

The application of the *privacy by design paradigm* in this context and so, making assumptions on the data to be communicated to the coordinator, on the kind of local/global function that has to be computed in the system and on the attack model to be countered, allows to obtain reasonable results in terms of privacy guarantees and data quality preservation.

Chapter 4

Privacy in Application Scenarios

4.1 Privacy in Distributed Density Map Computation

In this section we provide the study of privacy issues in a distributed framework where the coordinator has to evaluate the density of vehicles in correspondence of specific areas of a territory. We will show that if we know the specific application and so the function that the coordinator has to process then it is possible to formally define the background knowledge and the attack that an adversary can conduct to infer user sensitive information.

4.1.1 Density Monitoring Problem

Our application consists in evaluating the density of vehicles in correspondence of a given set $\mathcal{RP}$ of n_{RP} points in space, called *reference points*. In particular, density is estimated through a kernel-based approach, i.e., the density in a point is computed by counting all vehicles in space, yet weighted according to their distance from the point. The architecture of the framework is as follows. In the system we have a coordinator C and multiple remote nodes. Each node is a sensor that represents a user moving (vehicle) in the considered space and computes for each observation its *local kernel function* w.r.t. the set $\mathcal{RP}$ of reference points. The coordinator computes the density map as an aggregation of the all local functions computed by the nodes. The *global density map* is computed when the coordinator receives a query requiring this analysis.

Definition 4.1.1 (DMP: Density Monitoring Problem)**.** *Given a set $\mathcal{RP} = \{RP_1, \ldots, RP_{n_{RP}}\}$ of n_{RP} reference points, a set $\mathcal{V} = \{V_1, \ldots, Vn_V\}$ of vehicles and a kernel function $K(.)$, the density monitoring problem consists in computing, at each time instant, the function $f_{DMP} : \mathcal{V} \rightarrow R^{n_{RP}}$, defined as $f_{DMP}(V) = [K_1, \ldots, K_{n_{RP}}]^T$, where:*

$$\forall 1 \leq i \leq n_{RP}.\ K_i = \frac{1}{n_V} \sum_{j=1}^{n_v} K(V_j^{xy} - RP_i^{xy})$$

Here, $V_j^{xy} \in R^2$ and $RP_i^{xy} \in R^2$ represent, respectively, the actual position of vehicle V_j and the position of reference point RP_i. While the kernel function $K(.)$

could be for example a *triangular* function or *Gaussian* function. Most standard kernel functions are radial functions, i.e., their value only depend on the distance from the origin (in our case that translates to "distance between node and RP"). Moreover, usually their value drops monotonically as such distance grows. The standard example is the Gaussian kernel. Figure 4.1 shows a simple example of the DMP for a single reference point and six vehicles. Therefore, in order to enable the coordinator C to compute each K_i, a node j could send one of three possible types of information:

1) its position V_j^{xy}

2) its distance from the RP_i, i.e, $d_{ij} = V_j^{xy} - RP_i^{xy}$

3) the contribution $K(V_j^{xy} - RP_i^{xy})$

In the first case C before computing the global function K has to compute for each node the distance and the kernel function $K(V_j^{xy} - RP_i^{xy})$ for each RP_i. In the second case, C has to compute only $K(V_j^{xy} - RP_i^{xy})$. Finally in the last case C has to compute only the global function. For the final result the three solutions are equivalent. But in the first case C knows exactly the user position, instead in the last two cases C knows that the user j is in the area defined by circle with radius equal to the distance $d_{ij} = V_j^{xy} - RP_i^{xy}$ and with center the point RP_i^{xy}. Our choice is to use the third option. So, each node sends to the coordinator C the value of the kernel function.

In the naive setting each node j at a given instant t computes for each reference point RP_i the kernel function $K(V_j^{xy} - RP_i^{xy})$ and communicates the list of contributions K_j^t to the coordinator. The coordinator maintains a data structure containing the last information communicated by each node. This data structure is a matrix where the j-th row contains the last information K_j communicated by the node j.

When the coordinator receives a query he uses the information in the global data structure for the computation of the global density map. Clearly, in this setting each node must communicate an update for each observation. Whenever the number n_V of vehicles or their location update frequency (or both) reach high values, it is necessary to trade the exactness of the estimation defined above with a reduction of information exchange and processing. The loss of precision, in our context, is bounded by a parameter ϵ, that represents the deviation from the exact output for the DMP.

Definition 4.1.2 (ADMP: Approximate DMP)**.** *Given a DMP with reference points $\mathcal{RP} = \{RP_1, \ldots, RP_{n_{RP}}\}$, vehicle set $\mathcal{V} = \{V_1, \ldots, Vn_V\}$ and kernel function $K(.)$, and given an error tolerance parameter ϵ, the approximate density monitoring problem consists in computing, at each time instant, a function $f_{ADMP} : \mathcal{V} \to R^{n_{RP}}$, such that it always holds that $error(f_{ADMP}(V), f_{DMP}(V)) \leq \epsilon$. Possible definitions for the error function include the following:*

- *Average:* $error_{AVG}(K^A, K) = \frac{\sum_{i=1}^{n_{RP}} |K_i^A - K_i|}{n_{RP}}$
- *Worst-case:* $error_{worst}(K^A, K) = max_{i=1}^{n_{RP}} |K_i^A - K_i|$

where $K = f_{DMP}(V)$ and $K^A = f_{ADMP}(V)$. Equivalently, we can define them as $error_{AVG}(K^A, K) = \frac{1}{n_{RP}} ||K_i^A - K_i||_1$, and $error_{worst}(K^A, K) = ||K_i^A - K_i||_\infty$.

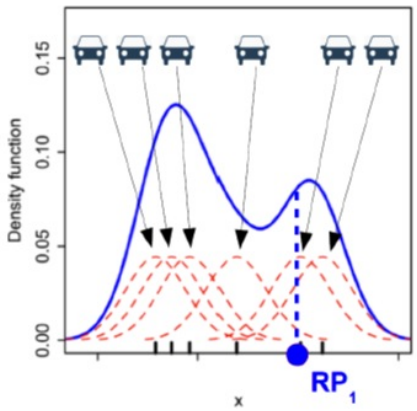


Figure 4.1: Example of vehicle density estimation for a reference point RP_1, on a single dimension, with a Gaussian kernel.

In order to reduce the amount of communications we introduce in the framework the use of a predictive model: the *user's mobility profiles* representing the user typical trips.

4.1.2 Mobility Profiles

In this section, we present the details of the definition of a *user's mobility profile*. The daily mobility of each user can be essentially summarized by a set of single trips that the user performs during the day. When trying to extract a *mobility profile* of users, our interest is in the trips that are part of their habits, therefore neglecting occasional variations that divert from their typical behavior. Therefore in order to identify the individual mobility profiles of users from their GPS traces, the following steps will be performed - see Figure 4.2:

1. divide the whole history of the user into trips (Figure 4.2(a))
2. group trips that are similar, discarding the outliers (Figure 4.2(b))
3. from each group, extract a set of representative trips, to be used as mobility profiles (Figure 4.2(c)).

Mobility Profile Definitions

Trips The history of a user is represented by the set of points in space and time recorded by their mobility device:

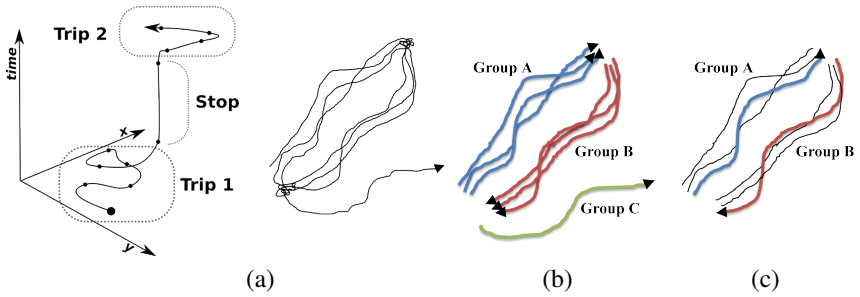


Figure 4.2: Mobility profile extraction process: (a) trip identification; (b) group detection/outlier removal; (c) selection of representative mobility profiles.

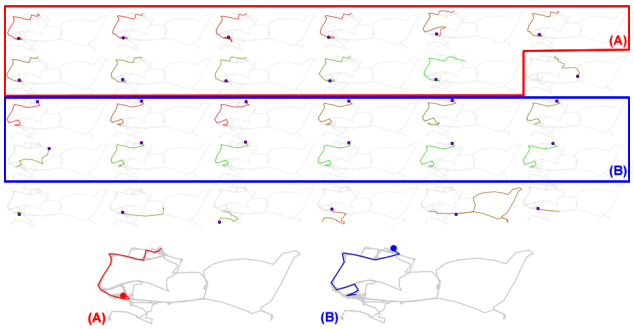


Figure 4.3: Trajectories of a user and the corresponding groups and routines extracted (A and B). Of the 30 trips, 11 are part of group A, and 12 of group B, while the remaining 7 are noise. The two routines are spatially similar, yet move in opposite directions (points represent the end of trips), i.e., south (A) vs. north (B).

Definition 4.1.3 (User history)**.** *The user history is defined as an ordered sequence of spatio-temporal points* $H = \langle p_1 \dots p_n \rangle$ *where* $p_i = (x, y, t)$ *and* x, y *are spatial coordinates and* t *is an absolute timepoint.*

This continuous stream of information contains different trips made by the user, therefore in order to distinguish between them we need to detect when a user stops for a while in a place. This point in the stream will correspond to the end of a trip and the beginning of the next one. We adopt a heuristic-based approach [91] for the detection of the stops. Thus we look for points that change only in time; i.e. they keep the same spatial position for a certain amount of time quantified by the temporal threshold $th^{stop}_{temporal}$. Specularly, a spatial threshold $th^{stop}_{spatial}$ is used to remove both the noise introduced by the imprecision of the device and the small movements that are of no interest for a particular analysis.

We indicate with $\overline{S} = \langle S_1 \dots S_t \rangle$ the set of all stops over H. Once we have found the stops in the users history we can identify the trips:

Definition 4.1.4 (Trip)**.** *A trip is defined as a subsequence* T *of the user's history* H *between two consecutive stops in the ordered set* $\overline{S}$ *or between a stop and the first/last point of* H *(i.e.,* p_1 *or* p_n*).*

The set of extracted trips $\bar{T} = \langle T_1 \dots T_c \rangle$ in Fig. 4.2(a), are the basic steps to create the user mobility profile. Notice that the thresholds $th^{stop}_{spatial}$ and $th^{stop}_{temporal}$ are the knobs for expressing specific analytical requirements.

Trip Groups Our objective is to use the set of trips of an individual user to find his routine behaviors. We do this by grouping together similar trips based on concepts of spatial distance and temporal alignment, with corresponding thresholds for both the spatial and temporal components of the trips. In order to be defined as *routine*, a behavior needs to be supported by a significant number of similar trips. The above ideas are formalized as follows:

Definition 4.1.5 (Trip Group)**.** *Given a set of trips* $\bar{T}$*, spatial and temporal thresholds* $th^{group}_{spatial}$ *and* $th^{group}_{temporal}$*, a spatial distance function* $\delta : \bar{T}^2 \rightarrow \mathcal{R}$ *and a temporal alignment constraint* $\alpha : \bar{T}^2 \times \mathcal{R} \rightarrow \mathcal{B}$ *between pairs of trips, and a minimum support threshold* $th^{group}_{support}$*, a* trip group *for* $\bar{T}$ *is defined as a subset of trips* $g \subseteq \bar{T}$ *such that:*

1. $\forall t_1, t_2 \in g . \delta(t_1, t_2) \leq th^{group}_{spatial} \wedge \alpha(t_1, t_2, th^{group}_{temporal})$*;*
2. $|g| \geq th^{group}_{support}$.

Condition 1 requires that the trips in a group are approximately co-located, both in space and time, while condition 2 requires that the group is sufficiently large. Again, the thresholds are the knobs that the analyst will progressively tune the extraction process with.

Mobility Profile Each group obtained in the previous step represents the typical mobility habit of a user, i.e., one of his routine movements. Here we summarize the whole group by choosing the central element of such a group:

Definition 4.1.6 (Routine)**.** *Given a trip group g and the distance function δ used to compute it, its* routine *is defined as the medoid of the set, i.e.:*

$$routine(g, \delta) = \arg\min_{t \in g} \sum_{t' \in g \setminus \{t\}} \delta(t, t')$$

Notice that the temporal alignment is always satisfied over each pair of trips in a group, therefore the alignment relation α does not appear in the definition. Now we are ready to define the users mobility profile.

Definition 4.1.7 (Mobility Profile)**.** *Given a set of trip groups G of a user and the distance function δ used to compute them, the user's* mobility profile *is defined as his corresponding set of routines:*

$$profile(G, \delta) = \{routine(g, \delta) \mid g \in G\}$$

Mobility Profile Construction The definitions provided in the previous section were kept generic w.r.t. the distance function δ. Different choices can satisfy different needs, possibly both conceptually (which criteria define a good group/routine assignment) and pragmatically (for instance, simpler criteria might be preferred for the sake of scalability). Obviously, the results obtained by different instantiations can vary greatly. Hence the crucial point is the selection of groups of trajectories. Our proposal is to use a clustering method to carry out this task. We choose the clustering algorithm for trajectories proposed in [11], consisting of two steps. First, a density-based clustering is performed, thus removing noisy elements and producing dense – yet, possibly extensive – clusters. Secondly, each cluster is split through a bisection k-medoid procedure. Such method splits the dataset into two parts through k-medoid (a variant of k-means) with $k = 2$, then the same splitting process is recursively applied to each sub-group. Recursion stops when each resulting sub-cluster is compact enough to fit within a distance threshold of its medoid, by removing sub-clusters that are too small. The bisection k-medoid procedure guarantees that requirements 1 and 2 of Definition 4.1.5 are satisfied. The clustering method adopted is parametric w.r.t. a repertoire of similarity functions, that includes: *Ends* and *Starts* functions, comparing trajectories by considering only their last (respectively, first) points; *Route similarity*, comparing the paths followed by trajectories from a purely spatial viewpoint (time is not considered); *Synchronized route similarity*, similar to Route similarity but considering also time.

4.1.3 Approximate Density Map Computation

As explained in Section 4.1.1, in the naive setting we can have a lot of communications which can be reduced by the use of the user's mobility profiles. In this setting we can identify three main phases of the whole process: *Setup*, *Monitoring* and *Querying*.

Setup. In this phase the coordinator sends to the nodes the parameters for the computation of their profiles (i.e. the set of thresholds, spatial distance measure and temporal alignment relation) and the position of all the reference points. The nodes after the computation send back their mobility profiles to the coordinator.

Monitoring. In this phase each node periodically sends the information related to his position and this information will be used from the coordinator for the computation. During this phase each node uses his mobility profiles to reduce the number of communications. Specifically, the user's mobility profiles are adopted as predictive model. The predictive model is used as a dynamic constraint adopting the safe zone approach.

Why do nodes use the mobility profiles to reduce the communications? The idea is that each node should communicate his kernel function only when it is too far from its mobility profile. Since typically the movements of a user are compatible with his profile then this should reduce the number of communications. The use of profiles introduces in the system an error that in the definition of the problem (Definition 4.1.2) we call ϵ. In general, we have a global error ϵ that is admissible and this error is the composition of the error that each node can introduce. In Definition 4.1.2 we consider two possible error definitions: the worst case and the average case. In the worst case, each reference point must be estimated with the maximum error ϵ. Such error, then, can be "distributed" among the nodes in several ways. The basic solution is to allow each node an error of $\frac{\epsilon}{n_V}$, where n_V is the total number of nodes in the system. However, any other partitioning of ϵ in n_V parts is could be a good solution, provided that their sum is less than ϵ. In the average case, essentially we have an overall tolerable error equal to $\epsilon \times n_{RP}$ (where n_{RP} is the total number of reference points), and it can be distributed among the reference point in any way. The straightforward way is to do it uniformly, i.e., ϵ for each reference point, making it equivalent to the worst case. However, if a reference point is rather far from the traffic, it might generate lower values, and therefore be affected by lower errors, so part of its ϵ might be saved for less lucky reference points.

When and how does user node introduce an error? When a user j reaches a new position V_j^{xy}, for each reference point he computes $k(V_j^{xy} - RP_i^{xy})$ and $k(P_j^{xy} - RP_i^{xy})$, i.e, the kernel function considering his real position and the kernel function considering the position in the profile. If the difference is less than the admissible local error ϵ' then the user does not have to communicate anything because the coordinator already knows the value $k(P_j^{xy} - RP_i^{xy})$ otherwise he communicates the new value, i.e, $k(V_j^{xy} - RP_i^{xy})$.

Querying. In this phase the coordinator receives a query requiring the density map computation. So, it uses the last communication received from each node and/or the mobility profiles for the computation.

4.1.4 Attack Model

In this section we discuss about the privacy breaches and possible attack models in this specific application. We will show as, knowing the application where introducing privacy constraints., it is possible to formally define the attack models by customizing the general attack model we defined for the LIFT-based systems (Section 3.1.2). Specifically, we can define a precise adversary knowledge that an attacker may use in a

specific attack in order to infer sensitive information and we describe how the attacker could conduct the attack. Finally, For each attack model, we will provide some ideas about the general properties of a reasonable countermeasure against the considered attack and we will indicate some possible direction that could be investigated.

Privacy Issues and User's Mobility Profiles

The communication of a user's mobility profile can violate the user privacy because it reveals common and typical trips of the user. The coordinator for the computation does not have the necessity to know the user related to a specific profile. So, the first step of a privacy-preserving technique could be to de-identify the profiles, i.e., by removing the direct identifiers of the users. But it has been shown that the privacy protection cannot be accomplished by simple de-identification. Indeed, if an attacker knows some places commonly visited by a specific user he can use this information to re-identify the user in the collection of de-identified user profiles and to discover his whole typical trip.

Example 4.1.1. *Consider a framework with N users who compute their profiles and send them to the coordinator that is untrusted. The set of profiles received are de-identified so that the coordinator does not know the user who corresponds to a specific profile. Now, assume the coordinator knows that the user X commonly visits the points p_1 and p_5. May he use this information to infer the whole user profile? The answer is yes! Any attacker with this information can select all the profiles containing both p_1 and p_5. If the number of profiles compatible with this information is small the attacker has a high probability to link the user X to his real mobility profile.*

In the following we formalize the adversary knowledge.

Definition 4.1.8 (Adversary Knowledge)**.** *The attacker has access to the set of user profiles $\mathcal{P}$ and knows: (a) the details of the scheme used to anonymize the profiles, (b) the fact that a given user X is one of the n_V nodes of the framework and (c) a sub-sequence of approximate position S of the user X.*

The ability to link data to external information, which enables various respondents associated with the data to be re-identified is known as a *linking attack model*.

As explained in the previous chapter, in relational data the linking is made possible by *quasi-identifiers*, i.e., attributes that, in combination, can uniquely identify individuals, such as birth date and gender. The remaining attributes represent the private respondent's information, that may be violated by the linking attack.

Without any kind of additional semantic information on the data it is hard to make a clear distinction between quasi-identifiers (QI) and private information (PI) in the context of spatio-temporal data. Thus, in this case a sub-sequence of positions can play both the role of QI and PI. In a linking attack conducted by a sub-sequence of approximate positions known by the attacker are considered as QI while the entire mobility profile is the PI that is disclosed after the re-identification of the respondent, while the sub-sequence serves as QI.

Here, we consider the following attack:

Definition 4.1.9 (Profile-based Attack)**.** *Given the set of user mobility profiles $\mathcal{P}$ and the adversary knowledge introduced in Definition 4.1.8, the attacker: (1) constructs a set of candidate profiles in $\mathcal{P}$ compatible with the sub-sequence S and (2) tries to identify the whole mobility profile relative to X.*
The probability of identifying the whole profile by a sub-sequence S is denoted by $prob(S)$.

Countermeasure From the point of view of data protection, minimizing the probabilities of re-identification is desirable. Intuitively, the set of candidate profiles corresponding to a given sub-sequence S should be as large as possible. A good solution would be to minimize the probabilities of re-identification and maximize data utility by minimizing the transformation of the original data.

A possible compromise could be the k-anonymity setting.

The general idea is to control the probability of the re-identification of any profile to below the threshold $\frac{1}{k}$ chosen as parameter of the system. In order to do this, we should guarantee that the set of profiles that the coordinator receives are an anonymous version of the originals, such that, on the one hand, it is still useful for predicting the movement of users.

We can consider two possible solutions:

- each node sends its mobility profile to a trusted party that is an *anonymizer*. It transforms the set of profiles in such a way to obtain a k-anonymous version and sends the transformed set to the coordinator and back to the nodes.
- all the nodes cooperate to compute the k-anonymous version of the profiles without using any trusted party. In this case it is possible to apply a technique based on secure multiparty computation following the basic idea of the frameworks that compute a distributed spatio-temporal clustering [50, 49]. The main problems of this last solution are: (1) each node should have a lot of computation power and (2) this setting introduces communications among the nodes that normally are not considered in LIFT-based systems.

In general, the anonymization step here does not affect the final computation of the density map. This means that the anonymization approach does not introduce any error to the global density map computation. The only effect could be to increase the number of communications of a node. Indeed, the anonymization step, given a set of mobility profiles $\mathcal{P}$ will transform it to $\mathcal{P}'$. Clearly, the predictive models $\mathcal{P}'$ are less precise than $\mathcal{P}$ and will describe the typical mobility behavior of a user with some approximation, therefore it could be happen more often that the user will be far from his profile and, as explained in Section 4.1.3, this situation will bring to an increase of the number of communication. In other words, the countermeasure in this case has to keep under control:

- the *privacy protection* against the profile-based attack by maximizing the probability of re-identification of a user;
- the quality of the service in terms of the overall performance of the LIFT-based system, i.e., it should minimize the increase of the communications.

Privacy Issues and Updates Communications

At a given instant a user computes for each reference point RP_i the kernel function $K(V_j^{xy} - RP_i^{xy})$ and sends these values to the coordinator if he is too far from his profile. We call the information communicated in this case *update*. The communication of an update could violate the user privacy because it reveals with some little approximation his position. Indeed, most standard kernel functions are radial therefore the coordinator given a contribution As described above, at a given instant a user computes for each reference point RP_i the kernel function $K(V_j^{xy} - RP_i^{xy})$ can compute the area defined by the circle with center RP_i^{xy} and radius $V_j^{xy} - RP_i^{xy}$. But if more RP_i are involved the coordinator can intersect the various circles and estimate accurately the position of the node.

As in the case of the profiles, the coordinator for computing the density map does not have the necessity to know the user related to the update received. So, the de-identification of the updates can be applied without generating any problem for the global computation. Clearly, this does not solve the privacy risks. The attacker who knows that the user visited one or more areas can conduct an attack that allows to infer other places visited by the user. Clearly, the areas known by the attacker can: (a) belong to the user profile; (b) not belong to the user profile. In the first case the coordinator will not receive any communication about this areas because thanks to the use of the profiles as a predictive model no update is necessary, therefore the knowledge of the attacker cannot help him to infer other information about the user. In the case (b), when the user visits one or more areas known by the attacker (coordinator) then he will communicate the corresponding updates. As a consequence the attacker knowing that a specific user X visited that areas can infer a series of location visited by X.

In the following we formalize the adversary knowledge.

Definition 4.1.10 (Adversary Knowledge)**.** *The attacker has access to the set of updates from the users $\mathcal{U}$ and knows: (a) the details of the scheme used to anonymize the updates, (b) the fact that a given user X is one of the n_V nodes of the framework, (c) a set of approximate positions S of the user X and (d) the list of reference points $\mathcal{RP}$.*

Here, we consider the following attack:

Definition 4.1.11 (Distance-based Attack)**.** *Given the set of user updates and the adversary knowledge in Definition 4.1.10 the attacker: (1) constructs a set of candidate updates $\{U_i\} \subseteq \mathcal{U}$ each one compatible with the approximate positions S and (2) tries to identify the whole set of areas visited by the user X.*
The probability of identifying the whole set of places visited by X is denoted by $prob(S)$.

The point (1) of the above definition means that the adversary given an update, i.e., a list of $K(V_{xy} - RP_i)$ computes the real position of the user (with an approximation) and selects all the updates related to position similar to S. When the number of updates selected is low the probability of re-identification becomes high.

Countermeasure In order to counter the distance-based attack we will investigate solutions using location perturbation, k-anonymity, differential privacy or particular

combinations of them. The literature on privacy in location base services provides many possibilities to be investigated.

Clearly, the countermeasure has to have specific properties that are important for allowing a correct running of the system. The most important property is that the privacy-preserving method has to keep under control:

- the *privacy protection* and thus this method has to maximize the re-identification probability of a user that provide a way to measure the quantity of privacy that is guaranteed;
- the *data utility*, that in this specific context means to guarantee the minimum transformation to be applied to the update because it could generate other errors in the computation of the global density map.

4.2 Privacy for Measuring Customer-Location Interactions

4.2.1 Application Description

The goal of the application is to provide companies with up-to-date measures of customer-location interactions. Such measures are, for example, the total number of customers per week or the average frequency by which customers visit a shop. An interaction denotes hereby simply the visit of a person to a specified location, e.g. a supermarket or cinema.

In our application we assume that each person carries a mobile device which is able to determine the position of a user and is thus able to record the history of a user's movements. Given the set of trajectories of all users and a location database (e.g. points of interest) the number of visits of each person and location can be inferred. Formally a visit is defined as follows [57]:

Definition 4.2.1 (Visit)**.** *Given a geographic coordinate space $\mathcal{S}_\mathcal{C}$, a temporal coordinate space $\mathcal{T}_\mathcal{C}$, a location $l \subseteq \mathcal{S}_\mathcal{C}$, $l \neq \emptyset$, a mobile entity e along with the entity's trajectory function $tr : \mathcal{T}_\mathcal{C} \rightarrow \{ \{s \mid s \in \mathcal{S}_\mathcal{C}\}, \emptyset \}$ and a time interval $\varepsilon > 0$, a visit is the tuple (l, e, t_1, t_2) with $t_1, t_2 \in \mathcal{T}_\mathcal{C}$, $t_1 < t_2$ for which the following holds*

1. *the intersection of l and $tr(t)$ is non-empty for all $t \in [t_1, t_2]$, i.e. $l \cap tr(t) \neq \emptyset \ \forall t \in [t_1, t_2]$,*
2. *the time span $[t_1, t_2]$ is maximal, i.e. there exists no time interval $[t_1^*, t_2^*] \supseteq [t_1, t_2]$ so that $l \cap tr(t) \neq \emptyset \ \forall t \in [t_1^*, t_2^*]$,*
3. *the time interval of intersection is greater or equal to ε, i.e. $t_2 - t_1 \geq \varepsilon$.*

The definition requires a minimum visit duration which may be specified according to application requirements. In addition, a visit always spans the maximum time period that a person spends at the same location.

Example The owner of a supermarket chain would like to know on a monthly basis how many potential customers each of his supermarkets attracts, how often people go shopping at any of his supermarkets and which percentage of the people living nearby his supermarkets uses his shopping facilities.

These three questions can be answered by the visit potential measures *gross visits*, *average visits* and *entity coverage* as defined by [57]. Gross visits specify the total number of visits between a given set of mobile entities and geographic locations. Average visits specify the average number of visits per entity and entity coverage measures the percentage of entities which visit at least on location of the location set. Depending on the specification of the location set, visit potential measures for a single supermarket or all supermarkets of the chain can be calculated.

More formally visit potential measures are defined as follows [57].

Definition 4.2.2 (Gross visits)**.** *Given a location set L, a set of mobile entities E and the number of visits $NV(t, l, e)$ between each entity $e \in E$ and location $l \in L$ until time t. Gross visits are defined as the number of total visits until time t:*

$$grVs(t, L, E) = \sum_{l \in L} \sum_{e \in E} NV(t, l, e).$$

Definition 4.2.3 (Average visits)**.** *Given a location set L, a set of mobile entities E and the number of visits $NV(t, l, e)$ between each entity $e \in E$ and location $l \in L$ until time t. Average visits are the average number of visited locations per mobile entity until time t:*

$$avgVs(t, L, E) = \frac{\sum_{l \in L} \sum_{e \in E} NV(t, l, e)}{|E|}.$$

Definition 4.2.4 (Entity coverage)**.** *Given a location set L, a set of mobile entities E and the number of visits $NV(t, l, e)$ between each entity $e \in E$ and location $l \in L$ until time t. Entity coverage is defined as the proportion of mobile entities which visit at least one location of the location set until time t:*

$$eCov(t, L, E) = \frac{|\{e \in E \mid NV(t, L, e) \geq 1\}|}{|E|}.$$

4.2.2 Privacy Model

Naive Sceanrio

A naive solution to the above described scenario is to transmit the trajectory data of each entity along with an entity identifier to the coordinator (either concurrently or at given points in time). The coordinator stores the data and is thus in possession of a trajectory database with recent as well as historic movement data. In addition, the coordinator possesses sociodemographic information about each individual. Given a specific location and entity set the coordinator is able to calculate any required visit potential measure for any specified period of time. More formally, the data stored at the coordinator or local node has the following format:

Definition 4.2.5 (Trajectory database at coordinator)**.** *The trajectory database at the coordinator consists of a set of tuples of the form* (id, x, y, t) *which denote the identifier of an entity* (id) *along with its position* (x, y) *at time instant* t*.*

Definition 4.2.6 (Sociodemographic database at coordinator)**.** *The sociodemographic database at the coordinator consists of a set of tuples of the form* $(id, a_1, a_2, ..., a_n)$ *which denote the identifier of an entity* (id) *along with* n *sociodemographic characteristics* $a_1, ..., a_n$*.*

The specific type of sociodemographic information stored at the coordinator is application dependent. It may consist, for example, of gender, age, place of living etc. However, we assume that obvious personal identifiers as, for example the name or address of a person, are not contained in the data set.

Definition 4.2.7 (Trajectory database at local node)**.** *The trajectory database at a local node is a set of tuples of the form* (x, y, t) *which denote the entity's position* (x, y) *at time instant* t*.*

The described scenario offers only very weak privacy protection as is shown in the following attack models. Note that we assume that an attacker attacks either data stored at the coordinator or at a mobile node. We do not consider attacks during data transmission as they can be avoided by using cryptographic techniques.

In the first attack scenario we consider an attack on data stored at the coordinator, which is equivalent to an untrusted coordinator.

Definition 4.2.8 (Adversary Knowledge)**.** *The attacker has access to the trajectory and sociodemographic database. He knows that the identifiers in both databases correspond to each other. Further, the attacker knows parts of the movement history of a user and / or parts of a user's sociodemographic data and the details of the privacy-preserving approach.*

Definition 4.2.9 (Attack Model - Linking Attack)**.** *Given the knowledge in Definition 4.2.8, the attacker extracts all trajectories that contain the known movement sequences of a specific user* X*. The attacker also extracts all persons that match the known sociodemography. He then combines the resultant data records based on the identifier and tries to identify the movement history and/or further sociodemographic characteristics of the user* X*.*

In the second attack scenario, an attacker intrudes a mobile device and retrieves data stored on the device.

Definition 4.2.10 (Adversary Knowledge - Device Attack)**.** *The attacker is a third party and has access to the mobile device of a user and knows at which location the trajectory data is stored.*

Definition 4.2.11 (Attack Model- Device Attack)**.** *The attacker extracts the stored trajectory data that contain the movement sequences of a user.*

Definition 4.2.12 (Countermeasure - Device Attack)**.** *The trajectory data will be encrypted before it is stored on the device.*

While an attack on the mobile device may be counteracted by encrypting the trajectory data, this is not possible at the coordinator as the coordinator has to evaluate the mobility data.

Aim of Privacy Model

Knowledge about visiting behavior is a rich source of information for private and public companies or institutions. However, all attempts to generate that information must protect the privacy of the individuals. This is especially true in times when companies can potentially misuse their market position to collect sensitive mobility data from their customers.

The general aim of privacy protection in our model is that an attacker cannot infer

- historic movement information (including trajectory as well as pattern information),
- the current position or
- sociodemographic variables (e.g. age, gender, place of living)

of any participating individual. Note that our assumed model is stricter than k-anonymity. While k-anonymity allows the publishing of trajectories of movement patterns if there are at least k other individuals with a similar movement, our model forbids the disclosure of any such information. Our reason for this requirement is that even though a common movement behavior may be revealed, it gives the adversary knowledge about the movement of the individual.

Idea for the LIFT-Approach

As the above scenario shows it is not a good idea to centralize all mobility data at the coordinator from a privacy perspective. First, the true ambitions of the coordinator with respect to the data may be unknown. Second, an attack by a third party would disclose a large amount of sensitive data. In addition, centralization poses a problem with respect to scalability. If the application is deployed nationwide, tens of millions of potential devices will sent frequent updates to the coordinator. This massive amount of traffic data would cause serious network problems as well as processing problems at the coordinator.

Thus, privacy as well as scalability require the performance of local inference. Processing data locally has the advantage that sensitive mobility data can be encapsulated at the nodes. Only aggregate statistics will be transmitted to the coordinator. In consequence, the aggregation has a positive effect both on privacy and communication. The cost of communication can be further improved by transmitting data only if a local change is likely to cause a global change, i.e. by applying the safe zone approach.

Figure 4.4 depicts the general approach. Instead of storing trajectory data at the local nodes, the nodes will directly evaluate the number of visits for given sets of locations. Only an identifier for the location set and the number of visits have to be stored (using encryption technology). This data is sent via a proxy to the communicator. The

proxy ensures that the coordinator cannot identify data from a node via its IP address. As the data is encrypted, the proxy itself cannot evaluate the data from the nodes. For each location set the coordinator maintains a distribution of k-visiting entities. This distribution states how many entities visits a given location set once, twice, ... or not at all. It can be shown that this distribution is sufficient to derive all required visit potential measures.

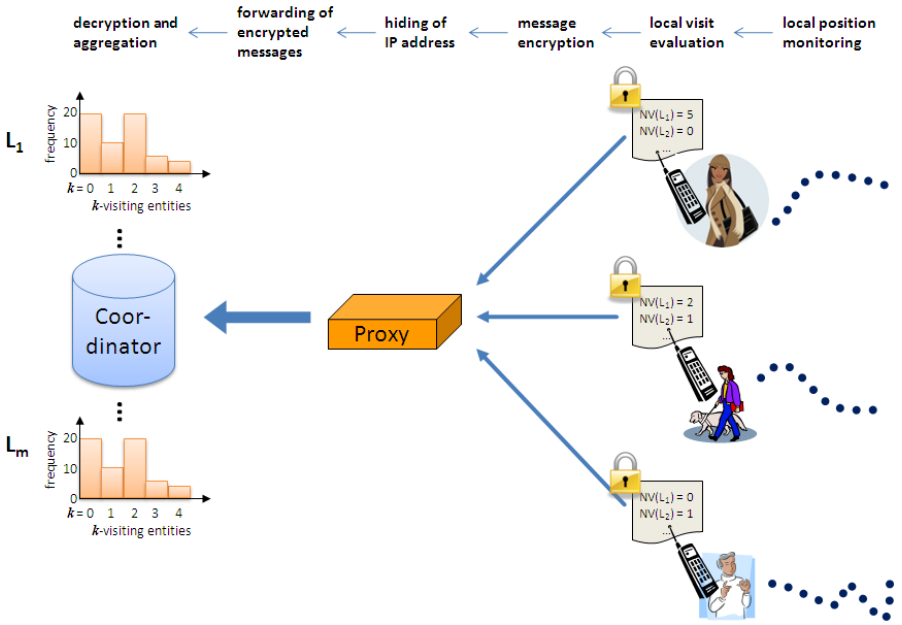


Figure 4.4: Privacy framework for customer-location interaction measurement

Clearly, without storing trajectory information, an attacker cannot access this information either at the coordinator or at a local node. If an attacker intrudes the coordinator, he will only obtain a statistic about the visit frequency of all entities for some (encoded) location set. The statistic may refer to a specific sociodemographic group, however, contains no personal information. Therefore, an intruder cannot infer information about a person's movements or current location.

Note that in this scenario we assume that the coordinator and the proxy do not cooperate.

4.3 Privacy for Counting Distinct Entities in a Region

4.3.1 Application Description

The aim of this application is to count the number of distinct persons in a given region. In our scenario these regions are typically large areas as, for example, a park or part

of city. The regions have in common that visitors can enter them through several entrances, so it is hard to maintain an overview of the total number of visitors. In addition, within the area people can move around freely to visit different attractions (e.g. stages, shows, shops) that are distributed over the area. As the movement inside the region is not controlled, crowds may form at attractions or at narrow passage ways which may potentially become dangerous. Typical events that match this description are open air concerts, sport events or youth meetings as the World Youth Day. Although events of this size, expecting several ten to hundred thousands of visitors, are carefully planned, the true number and behavior of visitors only appears at the event itself. For example, a simple change in weather conditions may lead to an increase of visitors (e.g. sunshine) or to a sudden leaving of persons (e.g. unexpected rain).

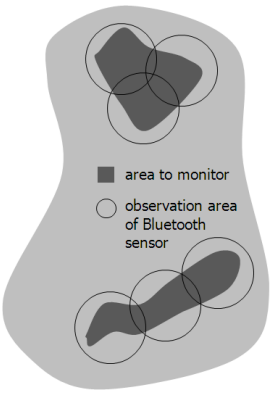


Figure 4.5: Areas to be monitored and their coverage by Bluetooth antennas

In our application we therefore want to monitor the number of people that are in one or more defined areas inside the event region. As monitoring device we decided to use Bluetooth sensors because they do not require to equip people with additional hardware devices. In addition, Bluetooth can be employed indoors as well as outdoors. Each of the areas to monitor will be covered by one or more Bluetooth antennas (see Figure 4.5). The monitoring areas of the antennas will typically overlap because weather conditions and obstacles may influence the signal strength. In addition, people will move between the monitoring areas of the antennas which will also lead to overlapping sensor readings. Thus, combining sensor readings of several antennas (local nodes) has to answer a *count distinct* query (global function).

More formally, for a single area that shall be monitored we use $i = 1..n$ Bluetooth antennas. The antennas scan the environment in previously defined intervals of time for Bluetooth-enabled devices and retrieve their media access control (MAC) addresses. At each local node we thus obtain a data stream of the following form.

Definition 4.3.1 (Bluetooth data stream)**.** *The Bluetooth data stream B_i at an antenna $i \in \{1, ..., n\}$ consists of a set of tuples of the form (id, t) where id is the MAC address of a scanned device at time moment t.*

For a a single scan we will denote the obtained data as follows.

Definition 4.3.2 (Bluetooth scan)**.** *The set of MAC address collected at antenna* $i \in \{1, ..., n\}$ *at a single scan at time* t *is defined as* $B_{i,t} = \{id_1, id_2, ..., id_m\}$.

Our aim is to determine for each area covered by Bluetooth antennas $i = 1..n$ the number of distinct Bluetooth-enabled devices at time moment t, i.e.

$$| B_t | = | \bigcup_{i=1}^{n} B_{i,t} |.$$

Due to the overlapping of sensor areas we know that the number of distinct people $| B_t |$ in the area is bounded by the sum of people registered at each Bluetooth antenna, i.e.

$$| B_t | = | \bigcup_{i=1}^{n} B_{i,t} | \leq \sum_{i=1}^{n} | B_{i,t} |.$$

However, our global value is not a linear combination of the local sensor readings.

4.3.2 Privacy Model

Naive Scenario

The naive solution for the scenario is that each antenna (local node) sends a copy of each scan to a central server (coordinator) which then performs a count distinct query.

Definition 4.3.3 (Bluetooth database)**.** *The Bluetooth database* B *is a collection of tuples of the form* (aid, id, t) *where* aid *is the identifier of a Bluetooth antenna and* id *is the MAC address of a scanned device at antenna* aid *at time moment* t.

In a setting with an untrusted coordinator this may lead to serious problems as the coordinator will be able to use the MAC address as quasi identifier.

Definition 4.3.4 (Adversary Knowledge - MAC address as quasi identifier)**.** *The attacker has access to the Bluetooth database and knows the location of the Bluetooth antennas. In addition, he knows the user who he wants to monitor and that he carries a device with enabled Bluetooth function.*

Definition 4.3.5 (Attack Model - MAC address as quasi identifier)**.** *Before or after the event the attacker obtains the MAC address of the person that he wants to monitor (e.g. by performing a Bluetooth scan when he is close to the person). He then retrieves all data tuples for the given device id from the Bluetooth database.*

Definition 4.3.6 (Counter Measure - MAC address as quasi identifier)**.** *Before the local nodes send their data to the coordinator they apply a hash function to the MAC addresses of the scanned devices. All antennas that monitor a part of a larger area use the same hash function. The coordinator does not know the parameters of the hash function.*

However, even though the MAC addresses may be hashed at the local nodes before transmission, the coordinator is still able to recognize movement histories from the total of all data streams. By ordering tuples with the same hashed id by their time stamp, the coordinator obtains a trajectory on the spatial and temporal resolution of the Bluetooth antennas. As the scanning area of Bluetooth antennas typically ranges between 20-100 m the coordinator can obtain a high resolution of a person's position. If the coordinator knows the antenna locations he can easily extract movement patterns of a person.

Definition 4.3.7 (Adversary Knowledge)**.** *The attacker is the coordinator itself and has access to the Bluetooth database. In addition, he knows the location of the Bluetooth antennas as well as parts of the movement history of a specific user X.*

Definition 4.3.8 (Attack Model - Linking Attack)**.** *Given the knowledge in Definition 4.3.7, the attacker orders all data tuples by the (hashed) device ids and timestamps. He then extracts all data tuples for a given device id that contain the known movement sequences and tries to identify the user X.*

Note that in this scenario we concentrate on attacks on the data that the coordinator posses. Of course, also local nodes can be attacked, however, in such cases it might be easier for an attacker to place a Bluetooth antenna in the area by himself. Nevertheless, the hashing of MAC addresses prevents an identification of a user in case the logging data of a local node should be attacked. In addition, we make the strict assumption that the local nodes are trusted and will not cooperate in our setting.

Aim of Privacy Model

In order to find public acceptance to apply Bluetooth techniques at large events, the privacy standards have to be high. Therefore the general aim of privacy protection in our model is that an attacker cannot infer

- historic movement information,
- the current position or
- the MAC address of a device

of a person.

Idea for the LIFT-Approach

Similar to the previous scenario, our approach is to evaluate the Bluetooth data locally and to transmit only aggregated data to the coordinator. However, as stated above this requires the assumption that the local nodes are trusted.

An advantage of the scenario is that our global function, counting the number of distinct items within a given time interval over distributed data streams, has already been treated in literature. Especially, we will exploit sketches to anonymize and compress the data. The Flajolet-Martin sketch (FM sketch) has been designed to count the number of distinct items in a data stream [45, 16, 40].

In its basic form the FM sketch hashes items (MAC addresses) into a $d \times w$ array using d different hash functions. For each row the probability of an item to be hashed in bucket $k \in \{1, ..., w\}$ is 2^{-k}. If at least one item has been hashed in a bucket the value of the bucket is 1 else it is 0. The number of distinct items can be obtained from the FM sketch by evaluating the position within each row where the transition between used (1) and unused (0) buckets occurs. Due to their structure several FM sketches can be combined by performing a bucket-wise OR operation on their values (assuming that the same hash functions are applied at each node).

This means that in our scenario only the sketches have to be maintained at the local nodes. Also the communication to the coordinator is reduced to transmitting the sketches. The communicator combines the sketches and applies the sketch estimation function to obtain the number of distinct items. Each local node processes the MAC addresses of each scan, however, discards the device identifiers after processing. Thus, no identifying data has to be stored. In addition, the hash functions have to be known only by the local nodes. This means that the coordinator cannot infer information about the absence or presence of a given MAC address by analyzing the sketches. Furthermore, the hash functions show a high degree of collisions so that even in the case that the coordinator obtains a hash function he is unlikely to trace a single device id.

Chapter 5

Conclusion and Roadmap

Privacy is an ever-growing concern in our society: the lack of reliable privacy safeguards in many current services and devices is the basis of a diffusion that is often more limited than expected. Unfortunately, it is increasingly hard to transform the data in a way that it protects sensitive information because of the complexity of the systems and of the data where privacy is a serious concern. In the last few years, several techniques for creating anonymous or obfuscated versions of data sets have been proposed, which essentially aim to find an acceptable trade-off between data privacy on the one hand and data utility on the other. These techniques are designed for guaranteeing the privacy protection during the data publishing phase. In LIFT-based systems the scenario is completely different we have a distributed architecture where local nodes send data to central system. In the application scenarios described above we saw that transmitting data from local nodes to a central system, besides a computational bottleneck, may also be a privacy bottleneck when the data contains personal, possibly sensitive, information about people. For example, when the local nodes consist of mobile phones or other personal portable devices the centralization can be a violation to data protection laws or individual rights (or expectations) of privacy. As a consequence, a framework aimed at minimizing the communication of data from local nodes to a central system opens promising scenarios for data protection and privacy safeguards.

In the following we draw a roadmap for the research towards privacy-preserving LIFT-based systems. We plan to define for different application scenarios the privacy model. This means, that for each scenario we will provide:

- a formal definition of the adversary knowledge and attack models as we showed for the scenarios in Chapter 4;
- the design, implementation and test of suitable privacy-preserving frameworks;
- the definition of measures able to evaluate the data utility preserved after the data transformation.

In literature, many interesting anonymity and privacy techniques have been proposed for data publishing context (see Chapter 2). They cannot be applied directly to

LIFT-like system because of the distributed nature of the architecture. But, we think that it is possible to consider the existing approaches as a good start point to be investigated. In other words, we believe that with adequate specific considerations and assumptions we will devise new privacy-preserving techniques inspired by the well-known models such as k-anonymity, l-diversity, randomization, and so on.

Clearly, each proposed privacy-preserving approach has to provide quantifiable privacy guarantees and has to assure a quantifiable data utility and a system performance guarantee. In order to evaluate the data utility preserved we need the definition of specific measures able to quantify the effects of the data transformation on the data and the results of the local/global computations in the system. These measures will enable a deeply analysis of the imprecision and the information loss introduced by the data transformation. Given different systems with different global functions to be computed/monitored we need different measurements with completely different properties. So, even in this case we can say that the methodology for the evaluation of the information loss introduced by the privacy-preserving technique depends on the application.

Bibliography

[1] O. Abul, F. Bonchi, and M. Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *Proceedings of the 24th International Conference on Data Engineering, ICDE 2008*, pages 376–385, 2008.

[2] Osman Abul, Maurizio Atzori, Francesco Bonchi, and Fosca Giannotti. Hiding sensitive trajectory patterns. In *Workshops Proceedings of the 7th IEEE International Conference on Data Mining (ICDM 2007)*, pages 693–698, 2007.

[3] Osman Abul, Maurizio Atzori, Francesco Bonchi, and Fosca Giannotti. Hiding sequences. In *Proceedings of the 23rd International Conference on Data Engineering Workshops, ICDE 2007*, pages 147–156, 2007.

[4] Osman Abul, Francesco Bonchi, and Fosca Giannotti. Hiding sequential and spatiotemporal patterns. *IEEE Trans. Knowl. Data Eng.*, 22(12):1709–1723, 2010.

[5] Nabil R. Adam and John C. Wortmann. Security-control methods for statistical databases: A comparative study. *ACM Computing Surveys*, 21(4):515–556, 1989.

[6] Gagan Aggarwal, Tomás Feder, Krishnaram Kenthapadi, Samir Khuller, Rina Panigrahy, Dilys Thomas, and An Zhu. Achieving anonymity via clustering. In *Proceedings of the Twenty-Fifth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 153–162. ACM, 2006.

[7] Gagan Aggarwal, Tomás Feder, Krishnaram Kenthapadi, Rajeev Motwani, Rina Panigrahy, Dilys Thomas, and An Zhu. Anonymizing tables. In *Proceedings of Database Theory - ICDT 2005, 10th International Conference*, volume 3363 of *LNCS*, pages 246–258, 2005.

[8] Dakshi Agrawal and Charu C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the Twentieth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. ACM, 2001.

[9] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proceedings of ICDE*, 1995.

[10] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 439–450, 2000.

[11] G. Andrienko, N. Andrienko, S. Rinzivillo, M. Nanni, D. Pedreschi, and F. Giannotti. *Interactive Visual Clustering of Large Collections of Trajectories*. VAST: Symposium on Visual Analytics Science and Technology, 2009.

[12] Mikhail J. Atallah, Ahmed K. Elmagarmid, M. Ibrahim, Elisa Bertino, and Vassilios Verykios. Disclosure limitation of sensitive rules. In *Proceedings of the 1999 Workshop on Knowledge and Data Engineering Exchange, KDEX '99*, page 45. IEEE Computer Society, 1999.

[13] Maurizio Atzori, Francesco Bonchi, Fosca Giannotti, and Dino Pedreschi. Blocking anonymity threats raised by frequent itemset mining. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM 2005)*, pages 561–564. IEEE Computer Society, 2005.

[14] Maurizio Atzori, Francesco Bonchi, Fosca Giannotti, and Dino Pedreschi. k-anonymous patterns. In *Proceedings of 9th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 10–21, 2005.

[15] Maurizio Atzori, Francesco Bonchi, Fosca Giannotti, and Dino Pedreschi. Anonymity preserving pattern discovery. *The International Journal on Very Large Data Bases (VLDB)*, 17(4):703–727, 2008.

[16] Z. Bar-Yossef, T. S. Jayram, R. Kumar, D. Sivakumar, and L. Trevisan. Counting distinct elements in a data stream. In *Proc. of the 6th International Workshop on Randomization and Approximation Techniques (RANDOM'02)*, pages 1–10. Springer-Verlag, 2002.

[17] Alastair R. Beresford and Frank Stajan. Mix zones: user privacy in location-aware services. In *Proceedings of 2nd IEEE Conference on Pervasive Computing and Communications Workshops (PerCom 2004 Workshops)*, pages 127–131. IEEE Computer Society, 2004.

[18] Alastair R. Beresford and Frank Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46–55, 2003.

[19] Claudio Bettini and Sergio Mascetti. Preserving k-anonymity in spatio-temporal datasets and location-based services. In *PRISE*, 2006.

[20] Claudio Bettini, Xiaoyang Sean Wang, and Sushil Jajodia. Protecting privacy against location-based personal identification. In *Proceedings of Secure Data Management, Second VLDB Workshop, SDM 2005*, volume 3674 of *LNCS*, pages 185–199. Springer, 2005.

[21] Jim Burridge, Luisa Franconi, Silvia Polettini, and Julian Stander. A methodological framework for statistical disclosure limitation of business microdata. Technical Report 1.1-D4, CASC Project, 2002.

[22] LiWu Chang and Ira S. Moskowitz. Parsimonious downgrading and decision trees applied to the inference problem. In *NSPW '98: Proceedings of the 1998 workshop on New security paradigms*, pages 82–89. ACM, 1998.

[23] Keke Chen and Ling Liu. Privacy preserving data classification with rotation perturbation. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM 2005)*, pages 589–592. IEEE Computer Society, 2005.

[24] Chris Clifton. Using sample size to limit exposure to data mining. *Journal of Computer Security*, 8(4):281–307, 2000.

[25] Chris Clifton, Murat Kantarcioglu, Jaideep Vaidya, Xiaodong Lin, and Michael Y. Zhu. Tools for privacy preserving data mining. *SIGKDD Explorations*, 4(2):28–34, 2002.

[26] Alissa Cooper. A survey of query log privacy-enhancing techniques from a policy perspective. *ACM Transactions on the Web (TWEB)*, 2(4):1–27, 2008.

[27] Ramesh A. Dandekar, Josep Domingo-Ferrer, and Francesc Sebé. Lhs-based hybrid microdata vs rank swapping and microaggregation for numeric microdata protection. In *Inference Control in Statistical Databases*, pages 153–162, 2002.

[28] Elena Dasseni, Vassilios S. Verykios, Ahmed K. Elmagarmid, and Elisa Bertino. Hiding association rules by using confidence and support. In *Information Hiding*, Lecture Notes in Computer Science, pages 369–383. Springer, 2001.

[29] Josep Domingo-Ferrer and Josep M. Mateo-Sanz. On resampling for statistical confidentiality in contingency tables. *Computers & Mathematics with Applications*, 38(11–12):13–32, 1999.

[30] Josep Domingo-Ferrer and Josep M. Mateo-Sanz. Practical data-oriented microaggregation for statistical disclosure control. *IEEE Transactions on Knowledge and Data Engineering*, 14(1):189–201, 2002.

[31] Josep Domingo-Ferrer and Vicenç Torra. A quantitative comparison of disclosure control methods for microdata. In J. Theeuwes L. Zayatz, P. Doyle and Amsterdam: North-Holland J. Lane, editors, *Confidentiality, Disclosure, and Data Access : Theory and Practical Applications for Statistical Agencies*, pages 111–134. Elsevier, 2001.

[32] Josep Domingo-Ferrer and Vicenç Torra. Distance-based and probabilistic record linkage for re-identification of records with categorical variables. *Butlletí de l'ACIA*, 28:243–250, 2002.

[33] Josep Domingo-Ferrer and Vicenç Torra. Ordinal, continuous and heterogeneous k-anonymity through microaggregation. *Data Mining and Knowledge Discovery*, 11(2):195–212, 2005.

[34] Wenliang Du and Mikhail J. Atallah. Secure multi-party computation problems and their applications: a review and open problems. In *Proceedings of the New Security Paradigms Workshop 2001*, pages 13–22, 2001.

[35] Matt Duckham and Lars Kulik. A formal model of obfuscation and negotiation for location privacy. In *Proceedings of Pervasive Computing, Third International Conference, PERVASIVE 2005*, pages 152–170, 2005.

[36] Cynthia Dwork. Differential privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *ICALP (2)*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2006.

[37] Alexandre V. Evfimievski, Ramakrishnan Srikant, Rakesh Agrawal, and Johannes Gehrke. Privacy preserving mining of association rules. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 217–228. ACM, 2002.

[38] Stephen E. Fienberg. A radical proposal for the provision of micro-data samples and the preservation of confidentiality. Technical Report 611, Carnegie Mellon University Department of Statistics, 1994.

[39] Stephen E. Fienberg and Julie McIntyre. Data swapping: Variations on a theme by dalenius and reiss. In *Privacy in Statistical Databases: CASC Project International Workshop, PSD 2004*, volume 3050 of *LNCS*, pages 14–29. Springer, 2004.

[40] P. Flajolet and G. N. Martin. Probabilistic counting algorithms for data base applications. *J. Comput. Syst. Sci.*, 31:182–209, 1985.

[41] Ale Florian. An efficient sampling scheme: Updated latin hypercube sampling. *Journal Probabilistic Engineering Mechanics*, 7(2):123–130, 1992.

[42] Luisa Franconi and Julian Stander. A model based method for disclosure limitation of business microdata. *Journal of the Royal Statistical Society D-Statistician*, 51:1–11, 2002.

[43] B. C. M. Fung, K. Wang, and P. S. Yu. Top-down specialization for information and privacy preservation. In *ICDE*, pages 205–216, 2005.

[44] Benjamin C. M. Fung, Ke Wang, Ada Wai-Chee Fu, and Philip S. Yu. *Introduction to Privacy-Preserving Data Publishing: Concepts and Techniques*. Data Mining and Knowledge Discovery. Chapman & Hall/CRC, August 2010.

[45] S. Ganguly, M. Garofalakis, and R. Rastogi. Processing set expressions over continuous update streams. In *Proc. of the 2003 ACM SIGMOD international conference on Management of data (SIGMOD'03)*, pages 265–276. ACM, 2003.

[46] Fosca Giannotti and Dino Pedreschi, editors. *Mobility, Data Mining and Privacy - Geographic Knowledge Discovery*. Springer, 2008.

[47] Marco Gruteser and Xuan Liu. Protecting privacy in continuous location-tracking applications. *IEEE Security & Privacy*, 2(2):28–34, 2004.

[48] B. Hoh and M. Gruteser. Protecting location privacy through path confusion. In *Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, pages 194–205. IEEE Computer Society, 2005.

[49] Ali Inan, Selim Volkan Kaya, Yücel Saygin, Erkay Savas, Ayça Azgin Hintoglu, and Albert Levi. Privacy preserving clustering on horizontally partitioned data. *Data Knowl. Eng.*, 63(3):646–666, 2007.

[50] Ali Inan and Yücel Saygin. Privacy preserving spatio-temporal clustering on horizontally partitioned data. In A. Min Tjoa and Juan Trujillo, editors, *DaWaK*, volume 4081 of *Lecture Notes in Computer Science*, pages 459–468. Springer, 2006.

[51] Vijay S. Iyengar. Transforming data to satisfy privacy constraints. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 279–288. ACM, 2002.

[52] W. Johnson and J. Lindenstrauss. Extensions of lipshitz mapping into hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

[53] Roberto J. Bayardo Jr. and Rakesh Agrawal. Data privacy through optimal k-anonymization. In *Proceedings of the 21st International Conference on Data Engineering, ICDE 2005*, pages 217–228, 2005.

[54] Hidetoshi Kido, Yutaka Yanagisawa, and Tetsuji Satoh. An anonymous communication technique using dummies for location-based services. In *International Conference on Pervasive Services*, pages 88–97. IEEE Computer Society, 2005.

[55] Hidetoshi Kido, Yutaka Yanagisawa, and Tetsuji Satoh. Protection of location privacy using dummies for location-based services. In *Proceedings of the 21st International Conference on Data Engineering, ICDE 2005*, page 1248, 2005.

[56] Peter Kooiman, Leon Willenborg, and José Gouweleeuw. Pram: A method for disclosure limitation of microdata. Research paper no. 9705, 1997.

[57] C. Körner, D. Hecker, M. May, and S. Wrobel. Visit potential: A common vocabulary for the analysis of entity-location interactions in mobility applications. In M. Painho, M. Y. Santos, and H. Pundt, editors, *Geospatial Thinking - Proc. of the 13th AGILE International Conference on Geographic Information Science (AGILE 2010)*, Lecture Notes in Geoinformation and Cartography, pages 79–95. Springer, 2010.

[58] Guanling Lee, Chien-Yu Chang, and Arbee L. P. Chen. Hiding sensitive patterns in association rules mining. In *Proceeding of the 28th International Computer Software and Applications Conference (COMPSAC 2004)*, pages 424–429. IEEE Computer Society, 2004.

[59] Kristen LeFevre, David J. DeWitt, and Raghu Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 49–60. ACM, 2005.

[60] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Proceedings of the 23rd International Conference on Data Engineering, ICDE 2007*, pages 106–115. IEEE, 2007.

[61] Kun Liu, Hillol Kargupta, and Jessica Ryan. Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):92–106, 2006.

[62] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *Proceedings of the 22nd International Conference on Data Engineering, ICDE 2006*, page 24. IEEE Computer Society, 2006.

[63] Bradley Malin. Protecting dna sequence anonymity with generalization lattices. *Methods of Information in Medicine*, 44(5):687–692, 2005.

[64] Sergio Mascetti, Claudio Bettini, Xiaoyang Sean Wang, and Sushil Jajodia. k-anonymity in databases with timestamped data. In *Proceedings of the 3th International Symposium on Temporal Representation and Reasoning (TIME 2006)*, pages 177–186, 2006.

[65] Josep Maria Mateo-Sanz, Antoni Martínez-Ballesté, and Josep Domingo-Ferrer. Fast generation of accurate synthetic microdata. In *Proceedings of Privacy in Statistical Databases: CASC Project International Workshop, PSD 2004*, pages 298–306. Springer, 2004.

[66] Adam Meyerson and Ryan Williams. On the complexity of optimal k-anonymity. In *Proceedings of the Twenty-third ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 223–228. ACM, 2004.

[67] Anna Monreale. *Privacy by Design*. PhD thesis, Department of Computer Science, University of Pisa, June, 2011.

[68] Anna Monreale, Gennady Andrienko, Natalia Andrienko, Fosca Giannotti, Dino Pedreschi, Salvatore Rinzivillo, and Stefan Wrobel. Movement data anonymity through generalization. *Transactions on Data Privacy*, 3:91–121, August 2010.

[69] Anna Monreale, Roberto Trasarti, Dino Pedreschi, Chiara Renso, and Vania Bogorny. C-safety: a framework for the anonymization of semantic trajectories. *Transactions on Data Privacy*, 4(2):73–101, 2011.

[70] Juggapong Natwichai, Xue Li, and Maria E. Orlowska. Hiding classification rules for data sharing with privacy preservation. In *Proceedings of the Data Warehousing and Knowledge Discovery, 7th International Conference, DaWaK 2005*, pages 468–477, 2005.

[71] Mehmet Ercan Nergiz, Maurizio Atzori, Yücel Saygin, and Baris Güç. Towards trajectory anonymization: a generalization-based approach. *Transactions on Data Privacy*, 2(1):47–75, 2009.

[72] Stanley R. M. Oliveira and Osmar R. Zaïane. Privacy preserving clustering by data transformation. In *Proceedings of the XVIII Simpósio Brasileiro de Bancos de Dados, SBBD*, pages 304–318, 2003.

[73] Stanley R. M. Oliveira and Osmar R. Zaïane. Protecting sensitive knowledge by data sanitization. In *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003)*, pages 613–616. IEEE Computer Society, 2003.

[74] Stanley R. M. Oliveira and Osmar R. Zaïane. Data perturbation by rotation for privacypreserving clustering. Technical Report TR04-17, Department of Computing Science, University of Alberta, Edmonton, Canada, 2004.

[75] Federal Committee on Statistical Methodology. Statistical policy working paper 22, may 1994. Report on Statistical Disclosure Limitation Methodology.

[76] R. G. Pensa, A. Monreale, F. Pinelli, and D. Pedreschi. Pattern-preserving k-anonymization of sequences and its application to mobility data mining. In *PiLBA*, 2008.

[77] Ruggero G. Pensa, Anna Monreale, Fabio Pinelli, and Dino Pedreschi. Anonymous sequences from trajectory data. In *Proceedings of the Seventeenth Italian Symposium on Advanced Database Systems, SEBD 2009*, pages 361–372, 2009.

[78] S. Rizvi and J. R. Haritsa. Maintaining data privacy in association rule mining. In *VLDB*, pages 682–693, 2002.

[79] Donald B. Rubin. Discussion of statistical disclosure limitation. *Journal of Official Statistics*, (9(2)):461–468, 1993.

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

[81] Yücel Saygin, Vassilios S. Verykios, and Chris Clifton. Using unknowns to prevent discovery of association rules. *SIGMOD Record*, 30(4):45–54, 2001.

[82] Xingzhi Sun and Philip S. Yu. A border-based approach for hiding sensitive frequent itemsets. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM 2005)*, pages 426–433. IEEE Computer Society, 2005.

[83] Latanya Sweeney. Uniqueness of simple demographics in the u.s. population. Technical report, Laboratory for International Data Privacy, Carnegie Mellon University, Pittsburgh, PA, 2000.

[84] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International journal of uncertainty, fuzziness, and knowledge-based systems*, 2002.

[85] Manolis Terrovitis and Nikos Mamoulis. Privacy preservation in the publication of trajectories. In *Proceedings of the 9th International Conference on Mobile Data Management (MDM 2008)*, pages 65–72, 2008.

[86] V. S. Verykios, A. K. Elmagarmid, E. Bertino, Y. Saygin, and E. Dasseni. Association rule hiding. *IEEE TKDE*, 16(4):434–447, 2004.

[87] Ke Wang, Benjamin C. M. Fung, and Philip S. Yu. Template-based privacy preservation in classification problems. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 466–473. IEEE Computer Society, 2005.

[88] Ke Wang, Philip S. Yu, and Sourav Chakraborty. Bottom-up generalization: A data mining solution to privacy protection. In *Proceedings of the 4th IEEE International Conference on Data Mining (ICDM 2004)*, pages 249–256. IEEE Computer Society, 2004.

[89] Leon Willenborg and Ton DeWaal. *Elements of Statistical Disclosure Control*. Springer-Verlang, 2001.

[90] William E. Winkler. Using simulated annealing for k-anonymity. Technical Report 7, US Census Bureau.

[91] Xiangye Xiao, Yu Zheng, Qiong Luo, and Xing Xie. Finding similar users using category-based location history. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2010.

[92] Xiaokui Xiao and Yufei Tao. Anatomy: Simple and effective privacy preservation. In *Proceedings of the 32nd International Conference on Very Large Data Bases*, pages 139–150. ACM, 2006.

[93] Roman Yarovoy, Francesco Bonchi, Laks V. S. Lakshmanan, and Wendy Hui Wang. Anonymizing moving objects: how to hide a mob in a crowd? In *Proceedings of the EDBT 2009, 12th International Conference on Extending Database Technology*, pages 72–83, 2009.

[94] Justin Z. Zhan, Stan Matwin, and LiWu Chang. Privacy-preserving collaborative association rule mining. In *Proceedings of the Data and Applications Security XIX, 19th Annual IFIP WG 11.3 Working Conference on Data and Applications Security*, pages 153–165, 2005.

[95] Peng Zhang, Yunhai Tong, Shiwei Tang, and Dongqing Yang. Privacy preserving naive bayes classification. In *Proceeding of the Advanced Data Mining and Applications, First International Conference, ADMA 2005*, volume 3584 of *LNCS*, pages 744–752. Springer, 2005.

[96] Bin Zhou, Jian Pei, and WoShun Luk. A brief survey on anonymization techniques for privacy preserving publishing of social network data. *SIGKDD Explorations*, 10(2):12–22, 2008.