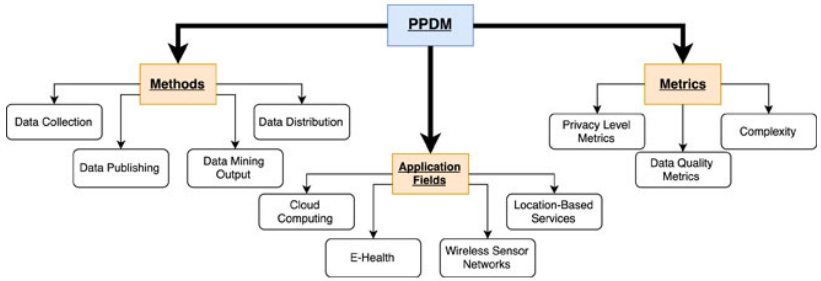
Privacy-Preserving Data Mining: Methods, Metrics, and Applications

Publisher: IEEE

2 Author(s) Ricardo Mendes ; João P. Vilela All Authors

A Framework for Categorizing and Applying Privacy-Preservation Techniques in Big Data Mining
Computer
Published: 2016



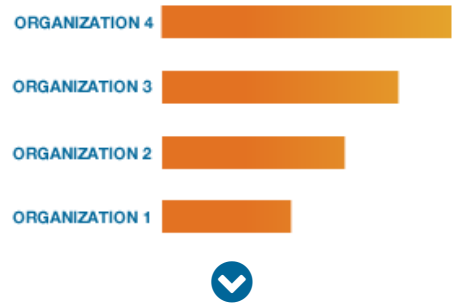
The graphical abstract presents a diagram of the topics covered in this survey, namely, Privacy Preserving Data Mining (PPDM) methods, metrics and applications. PPDM methods are categorised according to the data lifecycle phase at which they are applied, whereas metrics are classified as privacy level metrics, data quality metrics and complexity metrics. Finally, PPDM applications in relevant fields are presented. View less

Abstract: The collection and analysis of data are continuously growing due to the pervasiveness of computing devices. The analysis of such information is fostering businesses and c... **View more**

Metadata

Abstract:
The collection and analysis of data are continuously growing due to the pervasiveness of computing devices. The analysis of such information is fostering businesses and contributing beneficially to the society in many different fields. However, this storage and flow of possibly sensitive data poses serious privacy concerns. Methods that allow the knowledge extraction from data, while preserving privacy, are known as privacy-preserving data mining (PPDM) techniques. This paper surveys the most relevant PPDM techniques from the literature and the metrics used to evaluate such techniques and

Top Organizations with Patents on Technologies Mentioned in This Article



More Like This

Footnotes

presents typical applications of PPDM methods in relevant fields. Furthermore, the current challenges and open issues in PPDM are discussed.

Contents

Published in: IEEE Access (Volume: 5)

Page(s): 10562 - 10582

INSPEC Accession Number: 16950340

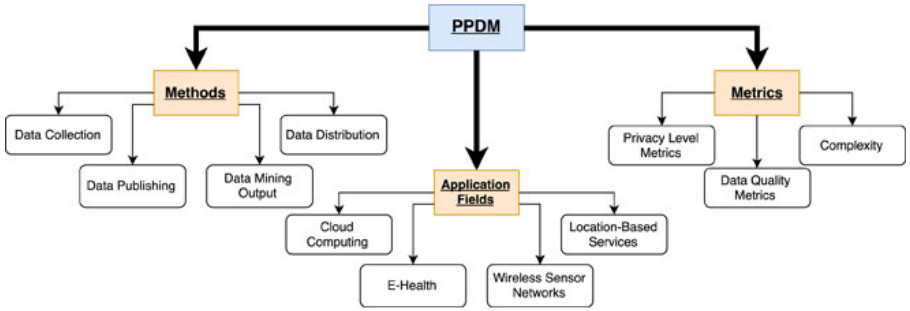
Date of Publication: 16 June 2017

DOI: 10.1109/ACCESS.2017.2706947

Electronic ISSN: 2169-3536

Publisher: IEEE

Funding Agency:



The graphical abstract presents a diagram of the topics covered in this survey, namely, Privacy Preserving Data Mining (PPDM) methods, metrics and applications. PPDM methods are categorised according to the data lifecycle phase at which they are applied, whereas metrics are classified as privacy level metrics, data quality metrics and complexity metrics. Finally, PPDM applications in relevant fields are presented. View less

Hide Full Abstract

SECTION I.
Introduction

Introduction

Contents

In the current information age, ubiquitous and pervasive computing is continually generating large amounts of information. The analysis of this data has shown to be beneficial to a myriad of services such as health care, banking, cyber security, commerce, transportation, and many others [1]. However, much of the collected information may be sensitive private data, which raises privacy concerns.

Although everyone has a concept of privacy, there is no universally accepted standard definition [2]. Privacy has been recognized as a right in the *Universal Declaration of Human Rights* [3] in 1948, however to a limited scope: the right to privacy at home, with family, and in correspondence. The difficulty in defining privacy comes as a consequence of the broadness of areas to which privacy applies [4], [5]. The scope of privacy can be divided into four categories [6]: *information*, which concerns the handling and collection of personal data; *bodily*, which relates to physical harms from invasive procedures; *communications*, which refers to any form of communication; *territorial*, which concerns the invasion of physical boundaries. This work will focus on the information category, which encompasses systems that collect, analyse and publish data.

In the information scope, Westin [7] defined privacy as "*the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others*", or in other words, as the right to control the handling of one's information. Bertino *et al.* [8] gave a similar definition, in terms of the control of the data, but explicitly incorporate the risks of privacy violation. These authors define privacy as "*the right of an individual to be secure from unauthorised disclosure of information about oneself that is contained in an electronic repository*". Other definitions were

proposed based on similar ideas of control and security [2]. Thus, one can conclude that the main idea of information privacy is to have control over the collection and handling of one's personal data.

Some benefits of the information technologies are only possible through the collection and analysis of (sometimes sensitive) data. However, this may result in unwanted privacy violations. To protect from information leakage, privacy preservation methods have been developed to protect owner's exposure, by modifying the original data [9], [10]. However, transforming the data may also reduce its utility, resulting in inaccurate or even infeasible extraction of knowledge through data mining. This is the paradigm known as Privacy-Preserving Data Mining (PPDM). PPDM methodologies are designed to guarantee a certain level of privacy, while maximising the utility of the data, such that data mining can still be performed on the transformed data efficiently. PPDM encompasses all techniques that can be used to extract knowledge from data while preserving privacy. This may consist on using data transformation techniques, such as the ones in Table 1, as primitives for adjusting the privacy-utility tradeoff of more evolved data mining techniques, such as the privacy models of Table 2 and the more classical data mining techniques of Table 3. PPDM also accounts for the distributed privacy techniques of Table 4 that are employed for mining global insights from distributed data without disclosure of local information. Due to the variety of proposed techniques, several metrics to evaluate the privacy level and the data quality/utility of the different techniques have been proposed [8], [11]–[12][13].

TABLE 1 Summary of the Privacy-Preserving Techniques at Data Collection (Section III-A)

Scenario: at data collection, an untrustworthy collector adversary may gather and improperly use private sensitive data from individuals. Randomization is employed to transform the original data so as to prevent privacy disclosure. The original data is not further used, nor stored.			
Randomization Method	**Description**	**Advantages & Disadvantages**	**Applications**
Additive Noise [26]	Data is randomised by adding noise with a known statistical distribution.	[+] Performs independently for each captured value (suitable for data collection). [+] Preserves statistical properties after reconstruction of the original distribution. [-] Limits data utility to the use of aggregate distributions	[27], [28]

		[-] Masking extreme values (such as outliers) require great quantities of noise, severely degrading data utility. [-] Noise reduction techniques can be used to accurately estimate the original individual values.	
Multiplicative Noise [29]	Data is randomised by multiplying noise with a known statistical distribution.	[+] More effective than additive noise at preserving privacy, since the reconstruction of the original individual values is more difficult. [+] Performs independently for each captured value (suitable for data collection). [+] Preserves statistical properties after reconstruction of the original distribution. [-] Limits data utility to the use of aggregate distributions. [-] Masking extreme values (such as outliers) require great quantities of noise, severely degrading data utility.	[26]–[30]

TABLE 2 Summary of Privacy-Preserving Techniques at Data Publishing (Section III-B) in Terms of the Employed Sanitisation Methods

Scenario: entities may seek to publish their data publicly for further research, and/or analysis. However, malicious individuals/adversaries may attempt to de-anonymize or target record owners for malevolent purposes. Privacy models are implemented to the datasets prior to the release, as to effectively anonymize records.

Privacy Model	Sanatization Methods	Description	Advantages & Disadvantages	Applications and Domains
k-anonymity [38], [39]	Generalization, Suppression	Anonymity is guaranteed by the existance of at least other k-1 undistinguishable (w.r.t. the QID) records for each record in a database. This group of k undistinguishable records is referred to as equivalence class.	[+] Simplicity of definition. [+] Great amount of existing algorithms [-] Assumes that each record represents a unique individual. If this is not the case, an equivalence class with k records does not necessarily link to k different individuals. [-] Sensitive attributes are not taken into consideration for the anonymization, which can disclose information, specially if all records in a class have the same value for the sensitive attribute.	Wireless Sensor Networks: [40] Location-based services: [41], [42] Cloud: [43] E-health: [44]
l-diversity [45]	Generalization, Suppression	Expands the k-anonymity model by requiring every equivalence class to have at least l "well-represented" value for the sensitive attributes.	[+] The diversity of sensitive attribute values is taken into consideration for the anonymization. [-] Does not take into consideration the distribution of the sensitive values, which can lead to privacy breaches when the sensitive values are distributed in a skewed away (generally true).	E-health: [44], [46] Location-based services: [42], [47], [48]
t-closeness [49]	Generalization, Suppression	Solves the l-diversity problem of skewed sensitive values distribution by requiring that the distribution of the sensitive values in each equivalence class to be "close" to the corresponding distribution in the original table, where close means upper bounded by a threshold t.	[+] Takes into consideration the distribution of the sensitive values when forming the equivalence classes. [-] The information about the correlation between quasi-identifier attributes and sensitive attributes is lost as t decreases (as privacy increases).	Location-based services: [50]
Personalized privacy [51]	Generalization	Achieved by creating a taxonomy tree using generalisation, and by allowing the record owners to define a guarding node. Owners' privacy is breached if an attacker is allowed to infer any sensitive value from the subtree of the guarding node with a probability (breach probability) greater than a certain threshold.	[+] Owner's can define their privacy level. [+] Preserves maximum utility while respecting personal privacy preferences. [-] Hard to implement in practice.	Social networks: [52] Location-based Services: [53], [54]
ϵ-differential privacy [27]	Perturbation (Randomization)	Ensures that a single record does not considerably affect (adjustable through the value ϵ) the outcome of the analysis of the dataset. In this sense, a person's privacy will not be affected by participating in the data collection since it will not make significant difference in the final outcome.	[+] Provides a formal privacy guarantee and a solid privacy loss metric. [+] Ensures that the participation of a single individual does not lead to a privacy breach greater than the obtained from the non-participation of the same individual. [-] There is no experimental guide on setting ϵ as it strongly depends on the dataset. [-] Privacy guarantees can require heavy data perturbation for numerical data, leading to non-useful output.	E-health: [55]–[57] Smart meters: [58] Location-based services: [59]

TABLE 3 Summary of Privacy-Preserving Techniques at Data Mining Output Privacy (Section III-C)

Scenario: the output of data mining techniques (e.g. classifiers) are often made accessible through applications or interfaces. Malicious users can query these systems to infer sensitive information about the underlying data. In these cases, either the data or the application are altered to prevent disclosure.

Technique	Description	Advantages & Disadvantages	Applications and Domains
Association Rule Hiding [72], [73]	Data is perturbed in order to avoid mining sensitive rules (association rule mining). Optimally, all non-sensitive rules are mined, while no sensitive rule is discovered.	[+] Numerous algorithms have been proposed. [-] The problem is NP-hard, requiring heuristic solutions that tend to also hide non-sensitive rules (i.e. loss of information).	Cloud: [74] Social networks: [75]
Downgrading Classifier Effectiveness [9], [76]	Data is sanitised to reduce classifiers' accuracy, and consequently the possibility of inferring sensitive data. Since some rule based classifiers use association rule mining methods as subroutines, association rule hiding methods are also applied to downgrade the effectiveness of a classifier.	Similar to association rule hiding, with the increased complexity of taking into account possible inferences from other public databases.	Cloud: [77], [78]
Query Auditing and Inference Control [79], [80]	Techniques to prevent information disclosure from sequences of aggregate data queries. In query inference control, either the original data, or the output of the query is perturbed. In online query auditing, one or more queries from a sequence are denied, whereas in offline query auditing, past query sequences are analysed to evaluate if the output breached privacy.	[+] Extensively researched in the context of statistical database security. [-] Denying/blocking certain queries can also reveal information.	E-health: [81] Cloud: [82]

TABLE 4 Summary of Distributed Privacy Techniques (Section III-D)

Scenario: multiple entities may seek to mine global insights in the form of aggregate statistics, over the conjunction of all partitioned data, without revealing local information to the other entities (possibly adversaries). Semi-honest and malicious adversaries participating in distributed computation may attempt to learn more from the shared information and, in such case, deviate from the protocols. In this scenario, secure multiparty protocols are implemented to prevent disclosure of local datasets.

Protocol	Description	Usage Context	Applications and Domains
1 out of 2 oblivious-transfer [87], [88]	A secure protocol between two parties, in where 1 message out of 2 messages is received and decrypted by the *receiver*, and the *sender* which inputs the pair of messages is oblivious to which message was decrypted. The input messages are encrypted with the public keys sent by the receiver, and only one of the private keys for the decryption is held by this party. This protocol has been generalised to the case of k out of N participants.	Secure and private exchange of information.	E-health: [93]–[95] Cloud: [96]
Homomorphic encryption [89], [90]	Allows for algebraic computations on ciphertext in a way that the deciphered result matches the result of the algebraic operation with the plaintext that originated the ciphertext.	To secure data while stored, or in transaction while allowing for computation over encrypted data.	E-health: [93], [97], [98] Cloud: [25], [99] Wireless Sensor Networks: [100]
Secure sum [101]	Obtains the sum of the inputs from each site, without revealing such inputs to the other participant entities.	Operations that are used as building blocks for distributed data mining techniques. These primitives limit the amount of information that is released to other participating entities.	Cloud: [102], [103]
Secure set union [101]	A technique to share itemsets and other structures between participant entities in order to create unions of sets, without revealing the owners of each set.		Cloud: [104], [105]
Secure size of intersection [101]	The objective of this protocol is to compute the size of the intersection of the local datasets, without revealing the data.		Cloud: [106]
Scalar product [101]	Computes the scalar product between two parties without revealing the input vector to the other entity. Secure scalar product is of crucial importance since many data mining problems can be reduced to the computation of the scalar product.		Cloud: [25], [107], [108]
Set Intersection [109]	Computes the intersection of two sets, one from each participating party, without revealing any other information.		Cloud: [110], [111]

PPDM has drawn extensive attention amongst researchers in recent years, resulting in numerous techniques for privacy under different assumptions and conditions. Several works have focused on metrics to evaluate and compare such techniques in terms of the achieved privacy level, data utility and complexity. Consequently, PPDM has been effectively applied in numerous fields of scientific interest. The vast majority of PPDM surveys focus on the techniques [10], [14], [15], and others on the metrics to evaluate such techniques [8], [12], [13]. Some only briefly discuss the evaluation parameters and the trade off between privacy and utility [16], [17], whereas others summarily describe some of the existing metrics [4], [11]. The survey in [11] does combine techniques, metrics and applications, but focuses on data mining, thus lacking many PPDM techniques, metrics, and other application fields, and [9] has applications in various areas but lacks metrics. This paper covers a literature gap by presenting an up-to-date and thorough review on existing PPDM techniques and metrics, followed by applications in fields of relevance.

The remainder of this survey is organised as follows. Section II introduces the problem of data mining and presents some of the most common approaches to extract knowledge from data. Readers already familiarised with the basic concepts of data mining can skip to section III, where several PPDM methods are described according to the data lifecycle phase at which they are applied. Section IV presents metrics to evaluate such algorithms. Some applications of the PPDM algorithms in areas of interest are presented in section V, with emphasis on the assumptions and context (identified by a scenario description) at which privacy can be breached. Section VI discusses some learned lessons about PPDM and presents open issues for further research on PPDM. Section VII concludes this paper.

SECTION II.
Classical Data Mining Techniques

Information systems are continuously collecting great amounts of data. Services can greatly benefit from the *knowledge extraction* of this available information [1]. The terms knowledge discovery from data

(KDD) and data mining are often ambiguous [18]. KDD typically refers to the process composed of the following sequence of steps: data cleaning; data integration; data selection; data transformation; data mining; pattern evaluation; and knowledge presentation. In this section, a brief review on the classical paradigms of the data mining step will be presented, to provide the reader enough understanding for the remainder of this paper.

Data mining is the process of extracting patterns (knowledge) from big

data sets, that can then be represented and interpreted. In [19], *pattern* is defined as an expression to describe a subset of data (itemset), or a model applicable to a subset. Since data mining methods involve pattern discovery and extraction, *pattern recognition* techniques are often used.[1] Moreover, pattern recognition and *machine learning* may be seen as "two facets of the same field" [20], that had origins in different areas. Therefore, throughout this work, data mining will refer to the process of extracting patterns, and pattern recognition and machine learning will be used interchangeably to denote data mining paradigms.

The main objective of data mining is to form descriptive or predictive models from data [19]. Descriptive models attempt to turn patterns into human-readable descriptions of the data, whereas predictive models are used to predict unknown/future data. The models are formed using machine learning techniques, that can be categorised as *supervised* and *unsupervised* [18]. Supervised learning techniques are methods in which the training set (the dataset used to form the model) is already labelled. That is, the training set has both the input data and the respective desired output, leading the machine to learn how to distinguish data, and thus, forming the model. In contrast, unsupervised techniques attempt to find relations in the data from unlabelled sets, or simply, no training set is used.

Association rule mining, **classification** and **clustering** are three of the most common approaches in machine learning, where the first two are supervised learning techniques, and the latter is an unsupervised learning mechanism. The following subsections will briefly detail each of these approaches. Readers can refer to [21], [22], and [18] for a comprehensive study on these subjects.

A. Association Rule Mining

Association rule mining algorithms are designed to find relevant

relationships between the variables of a dataset. These associations are then expressed by rules in the form: *if (condition); then (result)*. Association rules have a probability of occurrence, that is, if *condition* is met, then there is a certain probability of occurring *result*. Using the notation from [18], association rules can be formalized as follows. Let $\mathcal{I} = I_1, I_2, \ldots, I_m$ be a set of binary attributes called items, and D a database of transactions, where each transaction T is a nonempty itemset such that $T \subseteq \mathcal{I}$. Let $A \subset \mathcal{I}$ and $B \subset \mathcal{I}$ be subsets of $\mathcal{I}$. Then, an association rule is an implication $A \Rightarrow B$, with $A \neq \emptyset$, $B \neq \emptyset$.

Not all rules are interesting to mine, in fact, association rule mining algorithms only mine *strong* rules. Strong rules satisfy a minimum *support* threshold and a minimum *confidence* threshold. The support of a rule is the probability (percentage) of transactions in D that contain $A \cup B$, or mathematically:

$$support(A \Rightarrow B) = P(A \cup B)$$

View Source

Intuitively, this metric reflects the usefulness of the rule $A \Rightarrow B$. Confidence measures how often the rule is true in D . It is measured by the following equation:

$$confidence(A \Rightarrow B) = P(B|A)$$

View Source

Using the support and confidence metrics, two-steps are required to mine association rules [18]:

1. Find all itemsets in D with a support greater or equal to a minimum support threshold (frequent itemsets);

2. Generate the strong association rules from the frequent itemset.

B. Classification

Classification is a supervised learning problem whose objective is to create a model, in this specific case, called a classifier, that can identify the *class label* of unknown data [18]. In other words, a classifier is created from a training set – a set whose output (the class label) is known –, and it is then used to classify unknown data, into one of the existing classes. Thus, classification is a two-step approach problem: the *training phase* (or learning step) and the *classification phase*. More formally, one seeks to define a function $f(\cdot)$ that outputs a class label y for a given attribute vector $X = (x_1, x_2, \ldots, x_n)$ as input, where $x_i, \forall i \in 1, 2, \ldots, n$ represents a value for attribute A_i . That is:

$$y = f(X)$$

View Source

In this situation, $f(X)$ maps a tuple of attribute values to the respective class label. This mapping function can be represented by mathematical formulae, classification rules, or decision trees [18].

Having the mapping function $f(X)$, one can classify any attribute vector X in the classification phase. To evaluate the classifier, an already classified input is considered and its accuracy is calculated as the percentage of correct classifications obtained. However, the training set cannot be used, since it would result in an optimistic estimation of the accuracy [18], and therefore, *test sets* are used instead. In practice, the training set is randomly divided into a smaller (than the original) training set and a test set.

C. Clustering

Clustering, or cluster analysis, is a process of grouping sets of objects (observations) in groups (clusters), in a way that objects from a cluster have more *similarities* than objects from different clusters [18]. Each cluster may be considered as a class with no label, and thus, clustering is sometimes referred to as *automatic classification*, i.e. classification that does not require a training set, but learns from observations. Since cluster analysis is an unsupervised learning paradigm, it may reveal interesting unknown relations in the data.

Algorithms for clustering differ significantly due to unstandardised notion of cluster and the similarity metric [23]. A categorisation that encompasses the most important clustering methods is given in [18], based on the following properties :

- Partitioning criteria: conceptually, clusters may be formed either hierarchically (more general clusters contain other more specific clusters), or all clusters are in the same level;
- Separation: clusters may be overlapping or non-overlapping. In the overlapping case, objects may belong to multiple clusters, whereas in the non-overlapping, clusters are mutually exclusive;
- Similarity measure: the metric for the similarity between objects may be distance-based or connectivity-based;
- Clustering space: clusters may be searched within the entire data space, which can be computationally inefficient for large data, or within data subspaces (*subspace clustering*), where dimensionality may be reduced by suppressing irrelevant attributes.

Due to these (and other) properties, numerous algorithms have been

proposed for a myriad of applications [18], [22].

Being a fast-expanding field, data mining presents some challenges such as scalability, efficiency, effectiveness and social impacts. The concern in collecting and using sensible data that may compromise privacy is one of those impacts and one that is being extensively researched [1]. The following section will describe how privacy and data mining are related, and review some of the most important methods to protect and preserve privacy.

SECTION III.
Privacy and Data Mining

Data collection and data mining techniques are applied to several application domains. Some of these domains require handling, and often publishing sensitive personal data (e.g. medical records in health care services), which raises the concern about the disclosure of private information [1].

Privacy-Preserving Data Mining (PPDM) techniques have been developed to allow for the extraction of knowledge from large datasets while preventing the disclosure of sensitive information. The vast majority of the PPDM techniques modify or even remove some of the original data in order to preserve privacy [9]. This data quality degradation is known as the natural trade-off between the privacy level and the data quality, which is formally known as utility. PPDM methods are designed to guarantee a certain level of privacy while maximising the utility of the data to allow for effective data mining. Throughout this

work, sanitised or transformed data will refer to the data that resulted from a privacy-preserving technique.

Contents

Several different taxonomies for PPDM methods have been proposed [9]–[10][11], [14], [17], [24]. In this survey a classification based on the data lifecycle phase at which the privacy-preservation is ensured will be considered [10], namely at: data collection, data publishing, data distribution and at the output of the data mining.

The following subsections will describe each of the phases at which privacy is ensured by attesting how privacy may be lost and by describing some of the most applied privacy-preserving techniques. Tables 1, 2, 3 and 4 summarise the privacy preserving methods presented at each corresponding subsection, and enumerate some of the advantages, disadvantages and applications and domains of such techniques. A description of the scenario is also given to contextualise the adversarial assumptions and the nature of the privacy-preserving methods. Note that Table 1 does not present application domains, since these randomisation techniques are mainly used as primitives for more complex privacy preservation techniques, such as the ones presented in the remaining tables. In fact, Tables 2 and 3 correspond to more evolved privacy-preserving data mining techniques that usually rely on data transformation techniques to adjust the privacy-utility tradeoff, without requiring modification to the data mining algorithms. On the other hand, the distributed privacy techniques of Table 4 are usually building blocks for distributed computations that preserve privacy and must, therefore, be integrated into the data mining techniques (as seen in [25]), therefore requiring modifications.

A. Data Collection Privacy

To ensure privacy at data collection time, the sensory device transforms the raw data by randomising the captured values, before sending to the

collector. The assumption is that the entity collecting the data is not to be trusted. Therefore, and to prevent privacy disclosure, the original values are never stored, and used only in the transformation process. Consequently, **randomisation** must be performed individually for each captured value.

Most common randomisation methods modify the data by adding noise with a known statistical distribution, so that when data mining algorithms are used, the original data distribution may be reconstructed, but not the original (individual) values. Thus, the randomisation process in data mining encompasses the following steps: randomisation at data collection, distribution reconstruction (subtracting the noise distribution from the first step) and data mining on the reconstructed data [10].

The simplest randomisation approach may be formally described as follows. Let X be the original data distribution, Y , a publicly known noise distribution independent of X , and Z the result of the randomisation of X with Y . That is:

$$Z = X + Y \tag{1}$$

View Source

The collector estimates the distribution Z from the received samples $z_1, z_2, \ldots, z_n$, with n the number of samples. Then, with the noise distribution Y (Y has to be provided with the data), X may be reconstructed using:

$$X = Z - Y \tag{2}$$

View Source

Equation 1 corresponds to the randomisation process at data collection,

while equation 2 corresponds to the reconstruction of the original distribution by the collector entity. Note, however, that the reconstruction of X using equation 2 depends on the estimation of the distribution Z . If Y has a large variance and the number of samples (n) of Z is small, then Z (and consequently X) cannot be estimated precisely [10]. A better reconstruction approach using the Bayes formula may be implemented.

Additive noise is not the only type of randomisation that can be used at collection time. In fact, the authors of [31] show experimentally how ineffective this technique may be at preserving privacy. More effective (against privacy disclosure) techniques, that apply multiplicative noise to randomise the data also exist [29], [30].

Since the original data is modified into perturbed data, these methods require specific data mining algorithms that can leverage knowledge discovery from distributions of data, and not from individual entries. This may lead to a greater loss of utility than other privacy-preserving methods. Nevertheless, some data mining methods such as clustering and classification may require only access to the data distribution and will thus, work well with the randomisation [10].

Data modification may be applied at other phases than at data collection, and other methods besides additive and multiplicative noise do exist. In fact, randomisation is considered to be a subset of the perturbation operations[2] (see Section III-B). However, at collection time the assumption is that the collector is not trusted. Therefore, the original data must not be store, nor buffered, after the transformation. Thus, each value has to be randomised individually, that is, without considering other past collected values.

B. Data Publishing Privacy

Entities may wish to release data collections either publicly or to third parties for data analysis without disclosing the ownership of the sensitive data. In this situation, preservation of privacy may be achieved by anonymizing the records before publishing. PPDM at data publishing is also known as Privacy Preserving Data Publishing (PPDP).

It has been shown that exclusively removing attributes that explicitly identify users (known as *explicit identifiers*) is not an effective measure [32]. Users may still be identified by pseudo or quasi-identifiers (QIDs) and by sensitive attributes. A QID is a non-sensitive attribute (or a set of attributes) that do not explicitly identify a user, but can be combined with data from other public sources to de-anonymize the owner of a record,what is known as *linkage* attacks [33]. Sensitive attributes are person-specific private attributes that should not be publicly disclosed, and that may be also linked to identify individuals (e.g. diseases in medical records).

Sweeney in 2000 presented a report [34] on an analysis done over the 1990 U.S. Census to identify different combinations of attributes (QIDs) that would uniquely identify a person in the U.S. He found out that 87% of the population was identifiable by using the QID set {5-digit ZIP, gender, date of birth}. This study was then repeated with the 2000 U.S. Census by Golle [35], where the percentage of de-anonymized records using the same QID dropped to 63% of the population. In 2002, Sweeney identified the governor of Massachusetts from an anonymous voter list with the same QID set [36]. By linking these values to an accessible medical anonymized dataset from the Group Insurance Commission (GIC), the author was also able to obtain the governors' medical records. This simple example shows how QIDs are a potential thread to de-anonymize identities on datasets where only explicit identifiers are removed.

Aggarwal [10] states that the majority of anonymization algorithms focus on QIDs, disregarding sensitive attributes as it is wrongly assumed that without these, there is no risk of linkage attack with public information. In fact, the author claims that is fair to assume that an adversary has background information about its target [10] and thus concludes, that algorithms that do take into account sensitive attributes provide better privacy protection.

The anonymization of records in a database may be achieved by implementing different *privacy models*. Privacy models attempt to preserve records' owner identity by applying one, or a combination of the following data sanitising operations:

- Generalization: replacement of a value for a more general one (parent). Numerical data may specified by intervals (e.g. an age of 53 may be specified as an interval in the form of [50,55]), whereas categorical attributes require the definition of a hierarchy. A good example of a hierarchy could be the generalisation of the values engineer and artist from a occupation attribute to professional. Another possibility would be to have the parent value of student to represent all types of student in the same occupation attribute;

- Suppression: removal of some attribute values to prevent information disclosure. This operation can also be performed column wise in a data-set (removes all values of an attribute) or row wise (removes an entry).

- Anatomization [37]: de-associates QIDs and sensitive attributes in two separate tables making it more difficult to link QIDs to sensitive attributes. In this case, values remain unchanged;

- Perturbation: replacement of the original data for synthetic values with identical statistical information. The randomisation methods described in subsection III-A (additive and multiplicative noise) are examples of data perturbation. Data swapping and synthetic data generation are also perturbation techniques. In data swapping, sensitive attributes exchange between different entries of the dataset in order to prevent the linkage of records to identities, whereas in synthetic data generation, a statistical model is formed with the original data, and then synthetic values are obtained from the model.

This list is not an extensive enumeration of the existing operations. These are however, the most commonly used, and are sufficient to allow the comprehension of the remainder of this work. Readers can refer to [33] for a more thorough list.

Based on these operations, a set of privacy models has been proposed as follows. One of the most known privacy models is the k **-anonymity** model, proposed by Samarati and Sweeny [38], [39]. This model's key concept is that of k -anonymity: if the identifiable attributes of any database record are undistinguishable from at least other $k-1$ records, then the dataset is said to be k -anonymous. In other words, with a k -anonymized dataset, an attacker could not identify the identity of a single record since other $k-1$ similar records exist. The set of k records is known as *equivalence class* [10]. Note that "identifiable attributes" in the aforementioned definition refers to QIDs.

In the k -anonymity model, the value k may be used as a measure of privacy: the higher the value of k , the harder it is to de-anonymize records. In theory, in an equivalence class, the probability of de-anonymizing a record is $1/k$. However, raising k will also reduce the utility of the data since higher generalisation will have to occur.

Different algorithms have been proposed to achieve k -anonymity, where the vast majority applies generalisation and suppression operations [10]. This privacy model was one of the first applied for group based anonymization and served as a development base for more complex models. Some of the advantages of the k -anonymity privacy model include the simplicity of definition and the great amount of existing algorithms. Nevertheless, this privacy model has two major problems. The first problem has to due with the consideration that each record represents a unique individual, or in other words, that each represented individual has one, and only one record. If this is not the case, an equivalence class with k records does not necessarily link to k different individuals. The second problem relates to the fact that sensitive attributes are not taken into consideration when forming the k -anonymized dataset. This may lead to equivalent classes where the values of some sensitive attributes are equal for all the k records and consequently, disclosure of private information of any individual belonging to such groups. Other consequence of not taking into account sensitive attributes when forming the classes is the possibility of de-anonymizing an entry (or at least narrow down the possibilities) by associating QIDs with some background knowledge over a sensitive attribute.

The aforementioned attribute disclosure problem may be solved by increasing the diversity of sensitive values within the equivalence classes, an approach taken in the l **-diversity** model [45]. This model expands the k -anonymity model by requiring every equivalence class to abide by the l -diversity principle. An l -diverse equivalence class is a set of entries such that at least l “well-represented” values exist for the sensitive attributes. A table is l -diverse if all existing equivalence classes are l -diverse.

The meaning of "well-represented" values is not a concrete definition. Instead, different instantiations of the l -principle exist, differing on this particular definition [33], [45]. One of the simplest instantiations considers that the sensitive attributes are "well-represented" if there are at least l distinct values in an equivalence class, what is known as *distinct l-diversity*. In these conditions, a l -diverse equivalence class has at least l records (since l distinct values are required), and satisfies k -anonymity with $k = l$. A stronger notion of l -diversity is the definition of *entropy* l -diverse, defined as follows. An equivalence class is entropy l -diverse if the entropy of its sensitive attribute value distribution is at least $log(l)$. That is:

$$-\sum_{s \in S} P(QID, s) \log(P(QID, s)) \geq \log(l)$$

View Source

where s is a possible value for the sensitive attribute S , and $P(QID, s)$ is the fraction of records in a QID equivalence group, that have the s value for the S attribute. Note that entropy l -diversity can also be extended to multiple sensitive attributes by anatomizing the data [44].

Similarly to the k -anonymity model, in both entropy and distinct l -diversity instantiations, l (or in the former case, $\log(l)$) acts as a measure of privacy. Increasing this value, increases the variability of the existing values of the sensitive attribute in each equivalence class, decreasing the possibility of sensitive attribute disclosure. However, stronger generalisations and higher number of suppressions have to occur on the raw data, thus leading to higher loss of utility.

Although the l -diversity model increases the diversity of sensitive values within equivalence classes, it does not take into consideration the distribution of such values. This may present privacy breaches when the

sensitive values are distributed in a skewed away, which is generally true. To better understand this breach, consider the example given in [33]: a patient table with skewed attribute distribution, where 95% of the entries have FLU and the remaining 5% have HIV in the sensitive attribute column. An adversary seeks to find a record or groups of records having HIV, and has knowledge of the original sensitive attribute distribution. When forming the l -diverse groups, the maximum entropy would be achieved with groups having 50% of FLU entries and 50% of HIV entries. However, such groups would disclose that any entry within the group has a 50% of probability of having the value HIV in the sensitive attribute. This attribute disclosure may be worsened (infer with higher probability of having HIV), if the adversary has some background knowledge over the target(s).

In order to prevent attribute disclosure from the distribution skewness (skewness attacks), Li *et al.* [49] presented the t **-closeness** privacy model. This model requires the distribution of the sensitive values in each equivalence class to be "close" to the corresponding distribution in the original table, where close is upper bounded by the threshold t . That is, the distance between the distribution of a sensitive attribute in the original table and the distribution of the same attribute in any equivalence class is less or equal to t (t -closeness principle). Formally, and using the notation found in [49], this principle may be written as follows. Let $Q = (q_1, q_2, \ldots, q_m)$ be the distribution of the values for the sensitive attribute in the original table and $P = (p_1, p_2, \ldots, p_m)$ be the distribution of the same attribute in an equivalence class. This class satisfies t -closeness if the following inequation is true:

$$Dist(P, Q) \leq t$$

View Source

The t -closeness principle also has various instantiations depending on

The t-closeness principle also has various instantiations depending on the distance function that is used to measure the closeness [10], [49]. The three most common functions are the variational distance, the Kullback-Leibler (KL) distance and the Earth Mover's distance (EMD).

The three aforementioned privacy models preserve privacy by applying global/equitable measures to all records/identities. Xiao and Tao [51] presented the concept of **personalized privacy**, where the privacy level is defined by record owners. The purpose of this method is to preserve the maximum utility while respecting personal privacy preferences. Personalized privacy is achieved by creating a taxonomy tree using generalisation, and by allowing the record owners to define a guarding node. Owners' privacy is breach if an attacker is allowed to infer any sensitive value from the subtree of the guarding node with a probability (breach probability) greater than a certain threshold.

As an example of a personalised privacy model, consider a case where there is a sensitive attribute DISEASE. A record owner may be willing to disclose that he is ILL (and not NOT ILL), but protect which type, or which ill he contracts, i.e. ILL is his guarding node in the taxonomy tree. Other user may not mind to share that besides being ILL, it has a TERMINAL DISEASE, without specifying which specific disease. Finally, other record owner could allow to share the specific disease (e.g. LUNG CANCER). In this example, the LUNG CANCER value belongs to the taxonomy subtree of TERMINAL DISEASE, which is the guardian node of the second described owner, and TERMINAL DISEASE is in the subtree of ILL.

Personalized privacy has the advantage of letting record owner's define their privacy measure. However, this may be hard to implement in practice for two main reasons [33]: approaching record owners may not always be a viable/practical option; and, since record owners have no access to the distribution of sensible values, the tendency will be to over

protect the data, by selecting more general guarding nodes.

All previous privacy models try to either protect record owners' identity, or to protect the inference of sensitive values from anonymized records (or groups of records). Nevertheless, they do not measure how the presence (or absence) of a record impacts owner's privacy. Consider this hypothetical example: a statistical analysis over an anonymized database revealed that female smokers over 60 years old and weighting over 85kg, have a 50% chance of having cancer. A person belonging to this specific population will suffer from attribute disclosure, even if the dataset does not contain its record. From another point of view, if this person were on the database and the same conclusion would be reached, then there would be no further disclosure from the participation of this individual on such database, that is, no information would be *leaked*.

Dwork [27] presented the notion of differential privacy to measure the difference on individual privacy disclosure between the presence and the absence of the individual's record. In his work, the author proposed the ϵ **-differential privacy** model that ensures that a single record does not considerably affect the outcome of the analysis over the dataset. In this sense, a person's privacy will not be affected by participating in the data collection since it will not make much difference in the final outcome.

The ϵ -differential privacy model may be formalised as follows. Let $K(\cdot)$ be a randomised function, and D_1 and D_2 two databases differing at most on one record, then:

$$\ln\left(\frac{Pr\left[K\left(D_1\right)\in S\right]}{Pr\left[K\left(D_2\right)\in S\right]}\right)\leq\epsilon\quad\forall S\subseteq Range(K) \tag{3}$$

View Source

where ϵ is set a priori by the publishing entity, and S is a subset of

where ϵ is set a priori by the publishing entity, and S is a subset of $Range(K)$, with $Range(K)$ the set of all possible outputs of K. Note that equation 3 may be extended to group privacy, by having on the right side of the equation $c.\epsilon$, with c a small integer that corresponds to the number of records in the group [27].

Despite being a strong and formal privacy concept, differential privacy has some limitations [55], such as setting the appropriate value of ϵ. However, differential privacy is fairly recent and thus, more research is currently on-going [60].

Group anonymization privacy models (e.g. k-anonymity) and differential privacy are considered to be two of the major research branches in privacy [4]. In fact, several variants were proposed in the literature as to tackle some of the handicaps of the base models. Since these variants are considered extensions to the privacy models described through this section, a simple enumeration of these techniques is given below without any particular order. Interested readers can refer to [33] and [61] for detailed descriptions on some of the referred group anonymization privacy models.

- k-anonymity variants: k^m-anonymization [62], (α ,k)-anonymity [63], p-sensitive k-anonymity [64], (k, e)-anonymity [65], MultiR (MultiRelational) k-anonymity [66] and (X, Y)-anonymity [67];
- l-diversity variants: (τ ,l)-diversity [68] and (c, l)-diversity [45];
- t-closeness variants: closeness [69];
- ϵ-differential privacy variants: differential identifiability [70] and membership privacy [71].

In summary, the preservation of privacy at data publishing is achieved by applying privacy models that alter the original table in order to prevent

information disclosure. Each model has advantages and disadvantages in protecting from different types of inferences (e.g. identity, attribute). In contrast with privacy-preserving methods at collection time, privacy models can achieve a better control over the privacy level due to the publisher's access to the full data (recall the trade-off between privacy and utility). Other privacy models have been proposed in the literature [33], however, their underlying principles are the same as in the seminal contributions presented in this section.

C. Data Mining Output Privacy

The outputs of the data mining algorithms may be extremely revealing, even without explicit access to the original dataset. An adversary may query such applications and infer sensitive information about the underlying data. Below, a description of the most common techniques to preserve privacy to the output of the data mining is presented.

- **Association Rule Hiding** - In association rule data mining, some rules may explicitly disclose private information about an individual or a set of individuals. Association rule hiding is a privacy-preserving technique whose objective is to mine all non-sensitive rules, while no sensitive rule is discovered. Non-optimal solutions perturb data entries in a way that sensitive rules are hidden (e.g. suppression of the rule's generating item-sets), but may incorrectly hide a significant number of non-sensitive rules in the process. Nevertheless, different approaches, including exact solutions (all sensitive rules are hidden and no non-sensitive is hidden), have been proposed [10], [72]. The concept of association rule hiding was first introduced by Atallah *et al.* in [73].

- **Downgrading Classifier Effectiveness** - Classifier applications may also leak information to adversary users. A good example are the membership inference attacks, in which an adversary

determines if a record is in the training dataset (original data) [83], [84]. To preserve privacy in classifier applications, techniques for downgrading the accuracy of the classifier are often used [9], [76]. Since some rule based classifiers use association rule mining methods as subroutines, association rule hiding methods are also applied to downgrade the effectiveness of a classifier [9].

- **Query Auditing and Inference Control** - Sometimes entities may provide access to the original dataset, allowing exclusively statistical queries to the data. More specifically, users can only query aggregate data from the dataset, and not individual or group records. Nevertheless, some queries (or sequences of queries) may still reveal private information [9], [10]. Query auditing has two main approaches: *Query Inference Control*, where either the original data, or the output of the query are perturbed; and *Query Auditing*, where one or more queries are denied from a sequence of queries. Query auditing problems may be further classified into *offline* and *online* versions. In the former version, queries are known a priori, and the answers to such inquiry were already given to the user. The objective in this case is to evaluate if the query response(s) breached privacy. In the online version, queries arrive in an unknown sequence, and the privacy measures take action at the time of the queries. Query auditing and inference control

  techniques have been studied extensively in the context statistical database security. Classical approaches may be found in [79] and [80].

Note that in all four methods described above, the developed application is affected, since either the utility of the data used to build the application is lower than the original value, the application itself is downgraded, or the access to the data is restricted. Thus, the trade-off between privacy and utility is present.

D. Distributed Privacy

There are situations where multiple entities seek to mine global insights in the form of aggregate statistics, over the conjunction of all partitioned data, without revealing local information to the other entities (possibly adversaries).

A generalisation of this problem is the well studied secure multiparty computation (SMC) problem from the cryptography field [85]. In SMC, the objective is to jointly compute a function from the private input of each party, without revealing such input to the other parties. That is, at the end of the computation, all parties learn exclusively the output. This problem is solved using secure data transfer protocols that also apply to the privacy-preserving distributed computation [86].

In SMC, the assumption that adversaries respect the protocol at all times, is not often true [86]. The level of security of a protocol depends on the type of adversarial behaviour considered. Two main types of adversaries are defined in the literature: *semi-honest* adversaries and *malicious* adversaries. In the semi-honest behaviour, also called honest-but-curious model, adversaries abide by the protocol specifications, but may attempt to learn more from the received information. In the malicious behaviour model, adversaries deviate from the protocol and may even collude with other corrupted parties. Semi-honest scenarios are considered to be a good model of the real entities behaviour [86].

In a distributed scenario, a dataset may be partitioned either *horizontally* or *vertically*. In the horizontal case, each entity contains different records with the same set of attributes, and the objective is to mine global insights about the data. In vertically partitioned datasets, entities contain records with different attributes pertaining to the same identity. The junction of the dataset in this latter partition type allows to infer

knowledge that could not be obtained from the individual datasets. An example of an horizontally partitioned datasets is a clinic chain, where each site has different costumers, and the attributes associated with each costumer are common to all sites (such as type of disease and client's QID). For vertical partitioned datasets, stores with complementary items may be sequentially visited by the same clients, thus creating patterns that would not exist in each store's database. Distributed privacy-preserving algorithms exist for both types of partitioning.

In the remainder of this section, a description of two types of distributed privacy-preserving data mining protocols is presented. The first type, is a set of secure protocols that prevent information disclosure from the communication and/or computation between entities. For this set, the **oblivious transfer protocol** and the **homomorphic encryption** are described. The second type, considers a set of primitive operations that are often used in many data mining algorithms, and are thus suitable for distributed privacy. The described operations are the **secure sum**, the **secure set union**, the **secure size of intersection**, the **scalar product** and the **set intersection**. This second type of protocols may also use encryption techniques, such as the oblivious transfer protocol, to prevent information disclosure between entities.

The oblivious transfer protocol is a basic building block of most SMC techniques, and is by definition, a two-party protocol (between two entities). In PPDM, the 1 out of 2 oblivious-transfer protocol [87] is often implemented. In this approach, a *sender* inputs a pair (x_0, x_1) and learns nothing (has no output), while the *receiver* inputs a single bit $\sigma \in \{0, 1\}$ and learns x_σ . That is, the receiver learns one out of the possible two inputs/messages given by the sender, and the sender learns nothing.

The 1 out of 2 oblivious-transfer protocol procedure starts with the creation of two public encrypted keys by the receiver: P_σ with private key[3] K_σ and $P_{1-\sigma}$ with an unknown private key. The receiver proceeds to send P_σ and $P_{1-\sigma}$ to the sender. The sender encrypts x_0 with P_0 and x_1 with P_1 , and sends these encrypted messages back to the receiver. The receiver, knowing only how to decrypt P_σ (using K_σ), obtains only $x_\sigma, \sigma \in \{0, 1\}$.

The aforementioned description of the 1 out of 2 oblivious-transfer protocol works only for the semi-honest adversarial behaviour, since it is assumed that the receiver only knows how to decrypt one of the messages (only knows K_σ). However, oblivious transfer protocols exist for the malicious behaviour model [86], [88]. Furthermore, this protocol can be used over horizontal and vertical partitioned datasets.

Other technique from the SMC field that is raising attention amongst researchers is the **homomorphic encryption**. The concept of homomorphic encryption was firstly introduced by Rivest *et al.* [89], under de term *privacy homomorphism*. The objective was to be able to perform algebraic operations on encrypted text (ciphertext), in a way that the deciphered result would match the result of the operation with the plaintext that originated the ciphertext.

Earlier homomorphic cryptosystems were only able to perform specific algebraic operations and were thus considered partially homomorphic systems [90]. In contrast, fully homomorphic encryption supports any arbitrary function over the ciphertext. The first fully homomorphic system was proposed by Gentry [90], in 2009. Since then, there has been a development of other and sometimes more efficient solutions [91]. However, the efficiency of fully homomorphic systems is still insufficient for real-time applications [92].

Fully homomorphic encryption sees applications in most privacy-preserving cloud applications. For instance, queries can be made in an encrypted way, and the result is only decrypted when reaching the inquirer. This process not only protects the data in transmission, since it is encrypted but also protects from disclosure of information from the inquirer to the entity providing access to the search application through an encrypted query. Searching through encrypted files that are stored in the cloud is another possibility in full homomorphic systems. More examples of applications may be found in [90] and [91].

Clifton *et al.* [101] presented a set of secure multiparty computations to preserve privacy in distributed scenarios. Such techniques are often used as primitives to the data mining methods, and therefore, provide a useful approach to build distributed privacy-preserving data mining algorithms. These techniques include: secure sum, secure set union, secure size of set intersection and scalar product, and are referred in the literature as protocols. Below, the general idea of each method is described.

The **secure sum** protocol allows to obtain the sum of the inputs from each site, without revealing such inputs to the other entities. The implementation starts by designating one of the sites as the *master* site, where the computation starts and ends. The master site generates a random value R uniformly distributed in $[0, n]$, with n the upper bound of the final value, and then passes $(R + v_1)$ mod n , with v_1 the local input, to the next site. Each participating site then adds their local value to the received value and send the result of the mod n operation to the next site. Since the received values are uniformly distributed in the interval of $[0, n]$, sites learn nothing about other local values. In the end, the master site receives the last result and retrieves the true result (sum of the v_i values) by subtracting R. This value is then passed onto the other parties. The secure sum protocol requires a trusted master site, and to prevent disclosure, sites must not collude. Nevertheless, some

adaptations have been proposed to protect disclosure from such limitations [101], [102].

Contents

The **secure set union** is an important protocol for pattern mining [10]. The idea is to share rules, itemsets and other structures between sites, in order to create unions of sets, without revealing the owners of each set. One possible implementation of this protocol [101] uses commutative encryption,[4] where each site encrypts both its sets and received encrypted sets from other parties. Then, as the information is passed, the decryption takes place at each of the sites, by a different (scrambled) order than the encryption order. Since the decryption order is arbitrary, ownership anonymity is preserved.

Another protocol that uses commutative encryption to anonymize ownership of the items is the **secure size of set intersection**. The objective of this protocol is to compute the size of the intersection of the local datasets. The general idea is as follows. Each entity uses commutative encryption to encrypt local items and then passes them to another entity. When a set of items is received by one of the parties, encryption takes place for each of the received items, followed by an arbitrary order permutation, and finally passed onto another entity. When all items have been encrypted by every entity, the number of values that are equal across all the encrypted itemsets is the size of the intersection. Note that this technique does not require decryption and due to the use of commutative encryption,[4] the order of encryption is not important.

The last secure protocol presented in [101] is the **scalar product** between two parties. Formally, the problem can be defined as follows. Given two parties P_1 and P_2 , where P_1 has a vector $\vec{X} = x_1, \ldots, x_n$ and P_2 has a vector $\vec{Y} = y_1, \ldots, y_n$ of the same size of $\vec{X}$, the objective is to compute the scalar product $\vec{X} \cdot \vec{Y} = \sum_{i=1}^{n} x_i * y_i$, such that neither P_1

learns $\vec{Y}$, nor P_2 learns $\vec{X}$. Similarly to the secure sum, the secure scalar product may be achieved by adding randomness to an input vector, and the final output is retrieved by cancelling out the randomness [101], [112]. Some approaches also use homomorphic encryption, or the oblivious transfer protocol, to prevent data disclosure [112], [113].

Another important secure protocol to ensure distributed privacy and security is the set intersection, or private matching [109]. In this protocol, the intersection of two sets, each provided by one of the two participating parties P_1 and P_2 , is computed without revealing any other information. Formally, let $X = \{x_1, \ldots, x_n\}$ be the set of P_1 and $Y = \{y_1, \ldots, y_k\}$ be the set of P_2 , the objective is to compute $I = X \cap Y$, while revealing only I to each party. One efficient solution proposed in [109] uses a partially homomorphic encryption scheme, and is implemented as follows. P_1 defines a polynomial P whose roots are the elements in X , that is, $P(y) = (x_1 - y)(x_2 - y)\ldots(x_n - y) = \sum_{u=0}^{n} \alpha_u y^u$. This party then sends homomorphic encryptions of the coefficients to P_2 . With the encrypted coefficients, P_2 can compute for each $y_i \in Y$, $E(r_i \cdot P(y_i) + y_i)$ by multiplying $P(y_i)$ by a random number r_i (different for each i) and adding its input y_i , where $E(\cdot)$ represents the homomorphic encryption. P_2 then sends these k results to P_1 . For each $y_i \in X \cap Y$, $E(r_i \cdot P(y_i) + y_i) = E(y_i)$, since $P(y_i) = 0$ and thus, P_2 will know that y_i is in the intersection. P_1 can decrypt the received results and check if the results are either on their set, and consequently in the intersection, or are simply random values (recall the addition of r_i).

While the aforementioned implementation of the set intersection protocol involves only two parties, the multiparty case has also been studied [114]. Furthermore, the semi-honest behaviour model was assumed, however, a modification to provide security against malicious parties was also proposed in the original paper [109].

Contents

SECTION IV.
PPDM and Privacy Metrics

Since privacy has no single standard definition, quantifying privacy is quite challenging. Nevertheless, in the context of PPDMs, some metrics have been proposed. Unfortunately, no single metric is enough, since multiple parameters may be evaluated [8], [11], [86]. The existing metrics may be classified into three main categories, differing on what aspect of the PPDM is being measured: **privacy level** metrics measure how secure is the data from a disclosure point of view, **data quality** metrics quantify the loss of information/utility and **complexity** metrics, which measure efficiency and scalability of the different techniques.

Privacy level and data quality metrics can be further categorised into two subsets [8]: *data metrics* and *result metrics*. Data metrics evaluate the privacy level/data quality by appraising the transformed data that resulted from applying a privacy-preserving method (e.g. randomisation or a privacy model). Result metrics make a similar evaluation, but the assessment is done to the results of the data mining (e.g. classifiers) that were developed with the transformed data.

The following subsections present a survey on PPDM metrics concerning privacy level, data quality and complexity. Table 5 summarises the privacy level and data quality metrics described in this section, sub-categorised as data or result metrics.

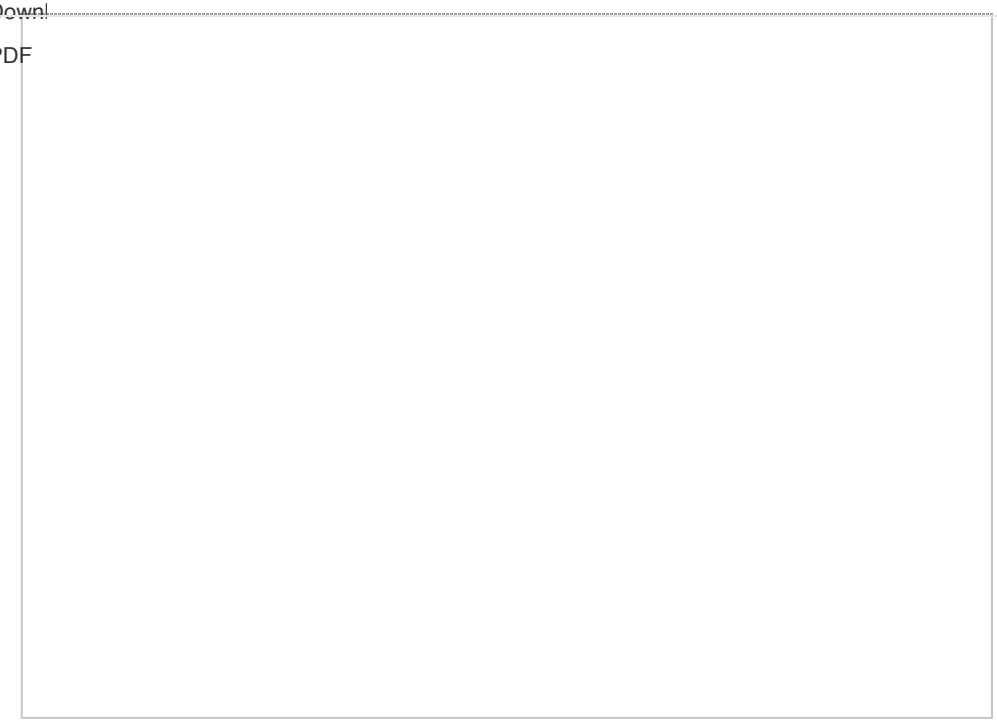
TABLE 5 Privacy Level and Data Quality Metrics, Further Categorised as Data

Metrics, if the Evaluation is Made Based on the Transformed Data, or as Result Metrics, if the Evaluation is Made Based on the Results of the Data Mining Technique (e.g., the Produced Classifiers) on the Transformed Data

Contents

Downl

PDF



A. Privacy Level

As previously mentioned, the primal objective of PPDM methods is to preserve a certain level of privacy, while maximizing the utility of the

data. The level of privacy metrics give a sense of how secure is the data from possible privacy breaches. Recall from the aforementioned discussion that privacy level metrics can be categorised into data privacy metrics and result privacy metrics. In this context, data privacy metrics measure how the original sensitive information may be inferred from the transformed data that resulted from applying a privacy-preserving method, while result privacy metrics measure how the results of the data mining can disclose information about the original data.

One of the first proposed metrics to measure data privacy is the **confidence level** [26]. This metric is used in additive-noise-based randomisation techniques, and measures how well the original values may be estimated from the randomised data. If an original value may be estimated to lie in an interval $[x_1, x_2]$ with $c\%$ confidence, then the interval $(x_2 - x_1)$ is the amount of privacy at $c\%$ confidence. The problem with this metric is that it does not take into account the distribution of the original data, therefore making it possible to localise the original distribution in a smaller interval than $[x_1, x_2]$, with the same $c\%$ confidence.

To address the issue of not taking into account the distribution of the original data, the **average conditional entropy** metric [117] is proposed based on the concept of information entropy. Given two random variables X and Z ,[5] the average conditional privacy of X , given Z is $H(X|Z) = 2^{h(X|Z)}$, where $h(X|Z)$ is the conditional differential entropy of X , defined as:

$$h(X|Z) = -\int_{\Omega_{X,Z}} f_{X,Z}(x,z) \log_2 f_{X|Z=z}(x)\, dxdz$$

View Source

where $f_X(\cdot)$ and $f_Z(\cdot)$ are the density functions of X and Z , respectively.

In multiplicative noise randomisation, privacy may be measured using the variance between the original and the perturbed data [28]. Let x be a single original attribute value, and z the respective distorted value, $\frac{Var(x-z)}{Var(x)}$ expresses how closely one can estimate the original values, using the perturbed data.

In the data publishing privacy subsection (subsection III-B), the k -anonymity, the l -diversity, the t -closeness, and the ϵ -differential privacy models were presented. Each of these models has a certain control over the privacy level, since variables k , l , t and ϵ are defined a priori and thus, act as privacy metrics, for a prescribed level of security. However, these metrics are specific to such techniques.

Result privacy metrics, as opposed to data privacy metrics, are metrics that measure if sensitive data values may be inferred from the produced data mining outputs (a classifier, for example). These metrics are more application specific than the previously described. In fact, Fung *et al.* [33] defined these metrics as “special purpose metrics”.

One important result privacy metric is the **hidden failure** (HF), used to measure the balance between privacy and knowledge discovery [8]. The hidden failure may be defined as the ratio between the sensitive patterns that were hidden with the privacy-preserving method, and the sensitive patterns found in the original data [115]. More formally:

$$HF = \frac{\#R_P(D')}{\#R_P(D)}$$

View Source

where HF is the hidden failure, D' and D are the sanitised dataset and the original dataset, respectively, and $\#R_P(\cdot)$ is the number of sensitive patterns. If $HF = 0$, all sensitive patterns are successfully hidden, however, it is possible that more non-sensitive information will be lost in the way. This metric may be used in any pattern recognition data mining technique (e.g. classifier or an association rule algorithm). Note that this metric does not measure the amount of information lost. For that, data quality metrics (presented in the following subsection) are used instead.

B. Data Quality

Privacy-preserving techniques often degrade the quality of the data. Data quality metrics (also called functionality loss metrics [11]) attempt to quantify this loss of utility. Generally, the measurements are made by comparing the results of a function over the original data, and over the privacy-preserved transformed data.

When evaluating data quality, three important parameters are often measured [12]: the *accuracy*, which measures how close is the transformed data from the original data, the *completeness*, which evaluates the loss of individual data in the sanitised dataset, and *consistency*, which quantifies the loss of correlation in the sanitised data. Furthermore, and similarly to the privacy level metrics, data quality measurements may be made from a data quality point of view, or from the quality of the results of a data mining application. Several metrics have been defined for both points of view, and for each of the parameters described above. In this subsection, a description of some of the most commonly used metrics will be given.

Fletcher and Islam [13] surveyed a series of metrics used to measure information loss from the data quality perspective, for generalisation and suppression operations, and for equivalence classes algorithms (such as the k -anonymity). For the generalisation and suppression techniques, the authors described the **Minimal Distortion** (MD) (first proposed as generalisation height [116]), the **Loss Metric** (LM) [118] and the **Information Loss** (ILoss) metric [51]. The MD metric is a simple counter that increments every time a value is generalised to the parent value. The higher the MD value, the more generalised is the data, and consequently, more information was lost. The LM and ILoss metrics measures the average information loss over all records, by taking into account the total number of original leaf nodes in the taxonomy tree. The ILoss differs from the LM metric by applying different weights to

different attributes, for the average. The weight may be used to differentiate higher discriminating generalisations [45]. For the equivalence class algorithms, the **Discernibility Metric** (DM) [120] was described. This metric measures how many records are identical to a given record, due to the generalisations. The higher the value, the more information that is lost. For example, in the k -anonymity, at least $k-1$ other records are identical to any given record, thus the discernibility value would be at least $k-1$ for any record. Increasing k , will increase generalisation and suppression, and consequently the discernibility value. For this reason, this metric is considered to be the opposite concept of the k -anonymity.

In [117], a metric to measure the accuracy of any reconstruction algorithm (such as in randomisation) is defined. The authors measure the information loss by comparing the reconstructed distribution and the original distribution. Let $f_X(x)$ be the original density function and $\hat{f_X}(x)$ the reconstructed density function. Then, the information loss is defined as:

$$I(f_X(x), \hat{f_X}(x)) = \frac{1}{2} E\left[\int_{\Omega_X} \left| f_X(x) - \hat{f_X}(x)) \right| \, dx\right]$$

View Source

where the expected value corresponds to the L_1 distance between the original distribution $f_X(x)$ and the reconstructed estimation $\hat{f_X}(x)$. Ideally, the information loss should be $I(f_X(x), \hat{f_X}(x)) = 0$, which states that $f_X(x) = \hat{f_X}(x)$, that is, the reconstruction was perfect, and therefore, no information was lost.

The metrics for evaluating the quality of the results are specific to the data mining technique that is used. These metrics are often based on the comparison between the results of the data mining with the perturbed

data and with the original data.

Two interesting metrics to measure data quality loss from the results of pattern recognition algorithms are the **Misses Cost** (MC) and the **Artifactual Patterns** (AP), presented in [115]. The MC measures the number of patterns that were incorrectly hidden. That is non-sensitive patterns that were lost in the process of privacy preservation (recall the aforementioned discussion on association rule hiding). This metric is defined as follows. Let D be the original database and D' the sanitised database. The misses cost is given by:

$$MC = \frac{\#\sim R_P(D) - \#\sim R_P(D')}{\#\sim R_P(D)}$$

View Source

where $\#\sim R_P(X)$ denotes the number of non-restrictive patterns discovered from database X . Ideally, an $MC = 0\%$ is desired, which means that all non-sensitive patterns are present in the transformed database. The AP metric measures artifact patterns, i.e. the number of patterns that did not exist in D , but were created in the process that led to D' . The following equation defines the AP metric.

$$AP = \frac{|P'| - |P \cap P'|}{|P'|}$$

View Source

where P and P' are the set of all patterns in D and D' , respectively, and $|\cdot|$ represents the cardinality. In the best case scenario, AP should be equal to 0, indicating that no artificial pattern was introduced in the sanitisation process.

For clustering techniques, the **Misclassification Error** (M_E) metric proposed in [119] measures the percentage of data points that "are not well classified in the distorted database". That is, the number of points that were not grouped within the same cluster with the original data and with the sanitised data. The misclassification is defined by the following equation:

$$M_E = \frac{1}{N} \times \sum_{i=1}^{k} (|Cluster_i(D)| - |Cluster_i(D')|)$$

View Source

with N the number of total points in the database, k the number of clusters, and $|Cluster_i(X)|$ the number of legitimate data points of the *ith* cluster in database X .

Additional metrics to evaluate the quality of results for classification and clustering are described in [13]. These metrics include commonly used quantitative approaches to measure the quality of data mining results, such as the Rand index [121] and the F-measure [122]. Finally note that cryptographic techniques implemented in distributed privacy preserve data quality since no sanitisation is applied to the data.

C. Complexity

The complexity of PPDM techniques mostly concern the efficiency and the scalability of the implemented algorithm [8]. These metrics are common to all algorithms, and therefore, only a brief discussion will be presented in this subsection.

To measure the efficiency, one can use metrics for the usage of certain resources, such as time and space. Time may be measured by the CPU time or by the computational cost. Space metrics quantify the amount of

memory required to execute the algorithm. In distributed computation, it may also be interesting to measure the communication cost, based either on the time, or the number of exchanged messages, and the bandwidth consumption. Both time and space are usually measured as a function of the input.

Scalability refers to how well will a technique perform under increasing data. This is an extremely important aspect of any data mining technique since databases are ever increasing. In distributed computation, increasing the inputs may severely increase the amount of communications. Therefore, PPDM algorithms must be designed in a scalable way. Scalability may be evaluated empirically by subjecting the system to different loads [123]. For example, to test if a PPDM algorithm is scalable, one can make several experiments with increasing input data, and measure the loss of efficiency. The loss of efficiency over experiments can then be used to measure scalability, since a more scalable system will present lower efficiency losses when under the same “pressure” as a less scalable system.

SECTION V.

PPDM Applications

In the previous two sections, a description of different privacy-preserving techniques, as well as a set of metrics to measure the privacy level, data quality and complexity was given. This section describes some existing PPDM applications, focusing on the employed privacy-preserving techniques and on the metrics used to measure the preservation of privacy.

The following subsections group the PPDM applications in the following fields: cloud computing, e-health, wireless sensor networks (WSN), and location-based services (LBS). Furthermore, in the e-health subsection, an emphasis on genome sequencing will be given due to the rising privacy research interest in the area, and in the LBS subsection, typical applications such as vehicular communications and mobile device location privacy will be described. Note that this section does not extensively surveys existing PPDM applications. Nonetheless, it is sufficient to illustrate some of the described privacy-preserving methods described in this work, and relate the applicability with the assumptions and privacy requirements of the applications. For comprehensive reviews on privacy in genome sequencing, WSN and location privacy readers can refer to [124]–[125][126], respectively.

A. Cloud PPDM

The U.S. National Institute of Standards and Technology (NIST) defined cloud computing [127] as "*a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction*." In other words, cloud is a distributed infrastructure with great storage and computation capabilities that is accessible through the network, anytime and anywhere. Therefore, applications (or services) that collect, store and analyse large data quantities often require the cloud. However, entities need to either trust cloud providers with data,[6] or to apply techniques that protect data while stored and/or during distributed computation. The cloud may be also used to publish data, and in this case, query auditing and inference control may be required. Consequently, cloud-based services are one of the primary focus of privacy-preserving techniques [128].

In [25], a scheme for classification over horizontally partitioned data under a semi-honest behaviour is proposed. This scheme allows owners to store encrypted data in the cloud, thus preserving privacy of data in communications and while stored. Furthermore, queries to the cloud are allowed to obtain the classes of a given set of inputs over encrypted data without the need for intermediate decryption. Using homomorphic encryption, the data, the query and the result are encrypted, and only the "querist" can decrypt the result, thus protecting the user from information leakage even against the cloud provider. The authors formally prove the security of the scheme and evaluate the computational and communicational complexity through simulations. Another approach that uses homomorphic encryption for cloud computing is presented in [99], for storing and mining association rules in a vertically distributed environment with multiple servers. This approach achieves privacy if at least one out of n servers is honest, and similarly to [97], security is proven mathematically, based on cryptography. Additionally, the authors of the paper [99] also present a series of efficient secure building blocks, which are required for their solution.

Privacy in the cloud is not limited to the use of secure protocols. For instance. in [43], a technique to publish data to the cloud based on the concept of k -anonymity is presented. The authors describe a novel approach where equivalence classes have less than k records, but still ensure the k -anonymity principle by exploring overlaps in the definitions of the equivalence classes. That is, by creating classes with less than k records (a divisor of k records) such that each record could belong to multiple classes and, thus, provide k -anonymity. By having a lower number of records in each class, the number of required generalisations is lower, and thus, more utility is preserved. The authors show this result by measuring the information loss, and also show the good performance of the implementation.

B. E-Health PPDM

Health records are considered to be extremely private, as much of this data is considered sensitive. However, the increase in the amount of data, combined with the favourable properties of the cloud has led health services to store and exchange medical records through this infrastructure [129]. Thus, to protect from unwanted disclosures privacy-preserving approaches are considered.

In [129], a survey on the state-of-the-art privacy-preserving approaches employed in the e-Health clouds is given, where the authors divide PPDM techniques in either cryptographic and non-cryptographic. The cryptographic techniques are usually based on encryption, whereas non-cryptographic approaches are based on policies and/or some sort of restricted access. An example of a cryptographic technique is found in [97], where the authors propose a privacy-preserving medical text mining and image feature extraction scheme based on fully homomorphic data aggregation under semi-honest behaviour is presented. The authors formally prove that their encryption is secure from the data point of view and from the results point of view. They also evaluate the performance of the PPDM by measuring computation and communication costs over the amount of input data.

An emerging field in e-health that is raising a growing privacy interest is genome sequencing. Genome sequencing is the process of studying genetic information about an individual through the study of sequences of DNA (Deoxyribonucleic acid). Genomic data sees applications in [124] health care, forensics and even direct-to-consumer services. Due to the advances in genome sequencing technologies and the capabilities of the cloud for computation and communication of data, this area has experienced a recent boom in research, including in the privacy field.

Genetic data is highly identifiable and can be extremely sensitive and personal, revealing health conditions and individual traits [124]. Furthermore, this type of data also reveals information about blood relatives, thus involving not only a single individual [124]. It is, therefore, critical to prevent unwanted disclosure of this type of data, while preserving maximum utility.

For genome data publishing, Uhlerop *et al.* [56] proposed a solution for releasing aggregate data based on the ϵ -differential privacy. This approach was motivated by the work in [130], where an attack to accurately identify the presence of an individual from a DNA mixture of numerous individuals was introduced. Thus, in [56], additive noise is added to the statistical data to be released, in order to achieve ϵ -differential privacy. Simulations have shown that ϵ -differential privacy is achievable and good statistical utility was preserved. However, for big and sparse data, the release of simple summary statistics is problematic from both privacy and utility perspectives.

Recently, McLaren *et al.* [98] proposed a framework for privacy-preserving genetic storing and testing. The depicted scenario involved the patients (P), a certified institution (CI), which has access to unprotected raw genetic data and therefore must be a trusted entity, a storage and processing unit (SPU) and medical units (MU). Both the SPU and the MU

follow a semi-honest adversarial behaviour, i.e. they will follow the protocol, but may attempt to infer sensitive data about the patients. Essentially, the patient supplies the data to the CI, which stores such data encrypted in the SPU using a partially homomorphic encryption scheme. MUs can then use secure two-party protocols with the SPU to operate the data in encrypted form, to be decrypted only when the result is returned from the SPU to the MU. Their framework has proven to be efficient, although it was limited to some genetic tests. Fully homomorphic encryption is suggested has a future solution to this limitation, however,

the computational cost is currently prohibitive.

C. Wireless Sensor Networks PPDM

Wireless Sensor Networks (WSN), sometimes called Wireless Sensor and Actuator Networks (WSAN), are networks of sparsely distributed autonomous sensors (and actuators), that monitor (act upon changes in) the physical environment [131] (e.g. light, temperature). Each sensor/actuator is referred to as node in the WSN and data is exchanged wirelessly between these devices. Since nodes have low battery capacity, one of the most important challenges in WSN networks is the efficiency in communication and processing of data at each node [131]. Thus, techniques to aggregate data from multiple sensors are often used to reduce network traffic and hence, improve battery life and consequently the sensor's lifetime. Data generated in WSNs may be considered sensitive in many different applications. For instance, sensed humidity of a room may determine room occupancy and house electrical usage over time may be used to track household behaviour [100]. Due to the aggregation of data and the WSNs' topology, attackers may try to control one or a few nodes to obtain access to all information. In this case, even if the communications are encrypted, the compromised nodes have the ability to decrypt the information, giving the adversary full access [132]. Therefore, privacy-preservation techniques may be required.

In [100], an approach to leverage the advantage of data aggregation for efficiency and to preserve privacy on the collected data is proposed. In this work, users can only query aggregator nodes to obtain aggregated data. Aggregator nodes query a set of nodes for the sensed values and proceed to compute the aggregation results over the received data, which is then forwarded to the inquirer. However, users must be able to verify the integrity of the aggregated data, since malicious users may try to control aggregation nodes and send false data. The WSN owners, on the other hand, want to prevent disclosure of individual sensor data, thus

restricting query results to aggregated data. The challenge here is how to verify the integrity of aggregated data, without access to the original sensed data. To address this issue, a framework where the user has full access to encrypted sensor data, in addition to the access to the aggregated data is proposed. The user can verify the integrity of the aggregated data by making use of the encrypted sensed values, without decrypting such data. Four solutions were described, each providing a different privacy-functionality trade-off, where one of the solutions uses (partially) homomorphic encryption to achieve perfect privacy, that is, no individual sensed value is disclosed. The authors compare the four solutions in terms of the number of messages exchanged and the supported aggregation functions.

Another approach that makes use of the aggregation of data to preserve privacy is proposed in [40]. This approach is non-cryptographic and implements a similar concept to k -anonymity, referred to as *k-indistinguishable*, where instead of generalisations, synthetic data is added to camouflage the real values (obfuscation). Aside from using k to control the number of indistinguishable values, a discussion to decrease the probability of privacy breach under colluding nodes (combining information from multiple nodes) is given. The authors also compare the performance of their implementation against encryption approaches,

where the results show that this method is more time and power efficient than such approaches.

The above examples concern information leakage from within the WSN. However, large WSN may be queried by multiple entities (clients) that may not trust the network owners [133]. The network owners may infer clients' intentions through the respective queries and profiles. These queries may be specific to a given area, or a given event thus revealing the intention. As stated in [133], one solution would be to query all sensors in

the network and save only the readings of interest. However, this would result in a significant load on the network, specially in large networks.

To address this issue, Carbunar *et al.* [133] proposed two approaches differing on the type of network models: querying server(s) that belong to a single owner (organization) and querying servers (at least two) belonging to different organizations. In both scenarios, servers are considered to be semi-honest, i.e. they abide by the protocols, but attempt to learn more than allowed. In the single owner model, the idea is to create a trade-off between the area of sensors that is queried and the privacy that is achieved. If the client queries only the region of interest, then no privacy is achieved, but the cost is minimal, whereas if the query targets all the network, the cost is maximum, but the achieved privacy is also maximized. The solution is thus a function that transforms an original query sequence into a transformed sequence, in order to conceal the region(s) of interest. Two metrics were used to measure privacy: the *spatial privacy level* and the *temporal privacy level*. The spatial metric is the inverse of the probability of the server guessing the regions of interest from the transformed query. The temporal privacy level measures the distance between the distributions of frequency of access of the regions obtained with the transformed and original query. A higher distance value translates into a better obfuscation of the frequency of access to the regions of interest. In the multiple owners situation,

cryptography is used to assign a virtual region to each sensor, that is only recognized by both the client and the sensor. A queried server then broadcasts the encrypted query, which is dropped by sensors that do not belong to the target virtual region. Sensors from the queried region encrypt the sensed values and return the results, which can only be decrypted by the client. This solution is fully private, as long as the servers used to create the virtual regions do not collude.

D. Location-Based Services PPDM

Pervasive technologies such as the global positioning system (GPS) allow to obtain highly accurate location information. LBSs use this spatiotemporal data to provide users with useful contextualized services [134]. However, this same information can be used to track users and consequently discover for example, their workplace, their houses' location and the places that they visit [126]. Furthermore, this information can also be used to identify users, since routes and behaviours often have characteristic patterns [135]. Therefore, the possibility of location information leakage is a serious concern and a threat to one's privacy. This type of leakage occurs when attackers have access to the LBS data, or when LBS providers are not trustworthy. In computational location, privacy is achieved with anonymity, data obfuscation (perturbation), or through application-specific queries [126]. Below, some examples are described.

For location anonymity, users can be assigned IDs (pseudonyms) to prevent identity disclosure. However, these pseudonyms must be changed periodically so that users cannot be tracked over time and space, and consequently disclose identity. To prevent this type of disclosure, Beresford and Stajano [41] presented the concept of "mix zone". In this approach, IDs are changed every-time users enter in a mix zone. In this type of zone, at least $k-1$ other users are present, such that changing all pseudonyms prevents the linkage between the old and new pseudonyms.

With this approach, and similarly to the k -anonymity privacy model, k may be used as a privacy metric.

In data obfuscation, the idea is to generate synthetic data or to add noise in order to degrade the quality of the spatial, and sometimes temporal, data. The assumption is that the LBS provider is untrustworthy. Simple examples include giving multiple locations and/or imprecise locations [126]. In [136] a solution to "cloak" users' locations using an intermediary anonymiser server (between the user and the LBS) is proposed. The user

queries the intermediary server (named *CacheCloak*) and if this server has the correct data for the location in cache, the data is sent to the user without querying the LBS. If the location data is not cached, the CacheCloak server creates a prediction path from the queried point until reaching a point in another cached path, and then queries the LBS for all these points. The received data is then cached and the correct data is forwarded to the user. As the user moves through the predicted path, the CacheCloak will return the cached information. When the user changes from the predicted path, and if the new position is not yet cached, then the same process is repeated. Since the predicted path is queried at the same time to the LBS, the service provider has no way to know the exact user location nor the movement direction. The authors present a metric based on the concept of (location) entropy to measure the achieved privacy level and how their solution to location privacy can work in real-time LBS services. Furthermore, an implementation to work under the assumption of an untrusted CacheCloak server is also discussed.

Another type of technique to achieve location privacy is to implement private queries, that is, location queries that do not disclose user location to the LBS provider. In [137] an approach using a secure protocol is presented, that allows users to query the LBS server through an encrypted query that does not reveal user's location. The protocol used is the private information retrieval (PIR), that has many similarities with the oblivious transfer protocol. With the encrypted query, the server computes the nearest neighbour, to retrieve the closest point of interest from the user location. The authors implement data mining techniques to optimise the performance of their solution, to identify redundant partial products, and show through simulation that the final cost in server time and the cost of communications is reasonable for location-based applications. This solution achieves full privacy in the sense that it is computationally infeasible for the server to decipher the encrypted query.

Vehicular communication privacy may be seen as a particular case of location privacy. These location-based systems are essentially networks, where cars and roadside units are nodes that communicate wirelessly to exchange spatiotemporal information [138]. Location-based services (LBS) make use of this data to provide drivers with useful content, such as traffic conditions, targeted advertising, and others. In this scenario, the highly accurate spatiotemporal information provided by the GPS is transferred to a third party server, that accumulates routes information that can be used to track drivers [138]. Privacy preservation is thus required, to protect drivers from being tracked. In [138], a privacy preservation approach under the assumption of untrusted collector is presented. This technique uses synthetic data generation to obfuscate the real trajectory of the car, by providing consistent fake locations. The authors present three measures of privacy: the tracking process, which is measured by the attacker's belief (probability) that a given location-time sample corresponds to the real location of the car; the location entropy, to measure the location uncertainty of an attacker; and the tracking success ratio, which measures the chance that the attacker's belief is correct when targeting a driver over some time t .

In this section we provide an overview of a set of relevant applications of PPDM methods, yet several other applications for the aforementioned domains and others exist, as listed in Tables 2, 3 and 4.

SECTION VI.
Lessons Learned and Open Issues

While PPDM is a fairly recent concept, extensive research has been ongoing by different scientific communities, such as cryptography, database management and data mining. This results in a variety of techniques, metrics, and specific applications. Nevertheless, it is essential to understand the underlying assumptions of each problem

to understand the underlying assumptions of each problem.

In this survey, PPDM techniques were partitioned according to the phase of the data lifecycle at which the privacy preserving technique is applied. This natural separation comes as a consequence of the different assumptions at each phase. These assumptions, which have been highlighted as a scenario description throughout Tables 1, 2, 3 and 4, condition the design of the PPDM techniques to address disclosure of data at different phases of the data lifecycle. These different phases are tied to distinct *user roles* and corresponding privacy concerns/assumptions on the adversary [24].

Even at a given data phase, there is no single optimal PPDM technique. The appropriate choice is often a matter of weighting the different trade-offs between the desired privacy level, the information loss, which is measured by data utility metrics, the complexity and even the practical feasibility of the techniques. Another aspect to take into consideration is the type of adversarial behaviour and corresponding privacy breaches that can be explored. The evolution in the research of the group anonymization techniques from k -anonymity to l -diversity and t -closeness presented in subsection III-B witnesses how different types of attacks can compromise privacy (in this case, anonymity) and how different techniques can be applied to protect from these invasions.

The evolution of PPDM is motivated by the privacy requirements of applications and fields/domains that handle data. Different application domains have different assumptions, requirements and concerns related to privacy. While this heterogeneity leads to a vast diversity of algorithms and techniques, the underlying concepts are often transversal. However, PPDM is far from being a closed subject [1]. Aside from the classical information technology requirements, such as scalability and efficiency, PPDM still presents several challenges with respect to data privacy.

A. Open Issues

Due to the broadness of the term, defining privacy is quite challenging. Even in the limited scope of information privacy, several definitions have been presented. In fact, there is always a fair amount of subjectiveness due to individuals' own privacy concept, beliefs and risk assertions. It is, therefore, necessary to develop systems that implement the concept of personalised privacy. This notion allows users to have a level of control over the specificity of their data. However, personalised privacy is challenging to implement. One one hand, it has been shown that users' concerns about privacy do not mirror users' actions [5], [139], that is, users' tend to trade their privacy for utility. Therefore, a personalised privacy solution could give users control over the data, but that control can become harmful, specially when users are unaware of the privacy risks of data disclosure [139]. On the other hand, the fact that users have no access to the overall distribution of sensitive values can lead to more protective decisions over their data, thus negatively affecting the utility of data [33]. Seeking novel solutions to this well-known trade-off between privacy and utility is therefore required in the context of personalised privacy solutions.

The oblivious transfer protocol and homomorphic encryption are two techniques for preserving privacy and security that are able to achieve full privacy without incurring in a loss of utility. However, these techniques are often not efficient for real-time applications [92]. Moreover, homomorphic encryption often requires a trade-off between functionality (supported functions) and efficiency. The development of more efficient secure protocols with better functionality trade-offs could increase the appliance of such techniques.

One important term that is raising interest in ubiquitous computing is that of context-aware privacy. In the envisioned world of the Internet of Things, sensors shall constantly monitor and sense the environment,

allowing easier inference of a user's context [140]. Context-aware privacy is achieved when a system can change its privacy policy depending on the user context [141]. Such systems may grant users added control over the collection of data by adapting privacy preferences to the context without being intrusive for the user. Nevertheless, while defining policies in an automated way according to context seems a promising direction, this may be difficult to achieve when faced with new and unknown contexts due to the complexity and incompleteness of context information [141], combined with users' uncertainty about what, when, by whom and how their information is collected [5]. Thus, further research on how to build/model efficient context-aware privacy systems is required.

Another consequence that arises from the absence of a standard definition is the hurdle in measuring privacy. Proposed metrics are often application specific, which renders a difficult comparison between the existing privacy preserving techniques. More generally applicable metrics, such as based on information entropy, are required for an effective comparison of different privacy-enhancing methodologies, thus leading to conscious choices of the adequate method for a given application.

Data publishing privacy is achieved with privacy models that sanitise data. However, due to the access to other publicly available sources,

adversaries can try to de-anonymize or to infer sensitive information [36], [142]. As the amount of published data continues to grow in both quantity and complexity, modelling background knowledge of adversaries presents several difficulties [143], such as the identification of what data can be used to de-anonymize and the amount of public data sources that can be linked together. This calls for the development of more evolved and realistic models of background knowledge available to adversaries, that can urge research on privacy mechanisms effective against these overhauled adversaries.

SECTION VII.
Conclusion

Businesses and institutions constantly collect data to offer or improve their existing services. However many of these services require the collection, analysis and sometimes publishing/sharing of private sensitive data. Information privacy is particularly critical with ubiquitous information systems capable of gathering data from several sources, therefore raising privacy concerns with respect to the disclosure of such data.

Privacy-Preserving Data Mining (PPDM) methods have been proposed to allow the extraction of knowledge from data while preserving the privacy of individuals. In this survey, an overview of data mining methods that are applicable to PPDM of large quantities of information is provided. This serves as preliminary background for the subsequent detailed description of the most common PPDM approaches. These PPDM methods are described according to the data lifecycle phase at which they can occur, namely collection, publishing, distribution and output of data.

The evaluation of these techniques is then addressed by analysing metrics to assess the privacy and quality levels of data as well as the complexity of the proposed PPDM approaches. Thereafter, the aforementioned PPDM techniques are considered from the point-of-view of their application to several practical domains and the rationale of their choice for those specific domains. Finally, some open issues and directions for future work are described.

Contents

Authors

Down!

PDF

References

Citations

Keywords

Metrics

Footnotes

Contents

Be the first to comment.

Down

PDF

Subscribe Add Disqus to your siteAdd DisqusAdd Do Not Sell My Data