Chapter 2

A General Survey of Privacy-Preserving Data Mining Models and Algorithms

Charu C. Aggarwal

IBM T. J. Watson Research Center
Hawthorne, NY 10532

charu@us.ibm.com

Philip S. Yu

University of Illinois at Chicago
Chicago, IL 60607

psyu@cs.uic.edu

Abstract   In recent years, privacy-preserving data mining has been studied extensively, because of the wide proliferation of sensitive information on the internet. A number of algorithmic techniques have been designed for privacy-preserving data mining. In this paper, we provide a review of the state-of-the-art methods for privacy. We discuss methods for randomization, k-anonymization, and distributed privacy-preserving data mining. We also discuss cases in which the output of data mining applications needs to be sanitized for privacy-preservation purposes. We discuss the computational and theoretical limits associated with privacy-preservation over high dimensional data sets.

Keywords: Privacy-preserving data mining, randomization, k-anonymity.

2.1   Introduction

In recent years, data mining has been viewed as a threat to privacy because of the widespread proliferation of electronic data maintained by corporations. This has lead to increased concerns about the privacy of the underlying data. In recent years, a number of techniques have been proposed for modifying or transforming the data in such a way so as to preserve privacy. A survey on some of the techniques used for privacy-preserving data mining may be found

in [123]. In this chapter, we will study an overview of the state-of-the-art in privacy-preserving data mining.

Privacy-preserving data mining finds numerous applications in surveillance which are naturally supposed to be "privacy-violating" applications. The key is to design methods [113] which continue to be effective, without compromising security. In [113], a number of techniques have been discussed for bio-surveillance, facial de-dentification, and identity theft. More detailed discussions on some of these sssues may be found in [96, 114–116].

Most methods for privacy computations use some form of transformation on the data in order to perform the privacy preservation. Typically, such methods reduce the granularity of representation in order to reduce the privacy. This reduction in granularity results in some loss of effectiveness of data management or mining algorithms. This is the natural trade-off between information loss and privacy. Some examples of such techniques are as follows:

- *The randomization method:* The randomization method is a technique for privacy-preserving data mining in which noise is added to the data in order to mask the attribute values of records [2, 5]. The noise added is sufficiently large so that individual record values cannot be recovered. Therefore, techniques are designed to derive aggregate distributions from the perturbed records. Subsequently, data mining techniques can be developed in order to work with these aggregate distributions. We will describe the randomization technique in greater detail in a later section.

- *The k-anonymity model and l-diversity:* The k-anonymity model was developed because of the possibility of indirect identification of records from public databases. This is because combinations of record attributes can be used to exactly identify individual records. In the k-anonymity method, we reduce the granularity of data representation with the use of techniques such as generalization and suppression. This granularity is reduced sufficiently that any given record maps onto at least k other records in the data. The l-diversity model was designed to handle some weaknesses in the k-anonymity model since protecting identities to the level of k-individuals is not the same as protecting the corresponding sensitive values, especially when there is homogeneity of sensitive values within a group. To do so, the concept of intra-group diversity of sensitive values is promoted within the anonymization scheme [83].

- *Distributed privacy preservation:* In many cases, individual entities may wish to derive *aggregate results* from data sets which are partitioned across these entities. Such partitioning may be horizontal (when the records are distributed across multiple entities) or vertical (when the attributes are distributed across multiple entities). While the individual

entities may not desire to share their entire data sets, they may consent to limited information sharing with the use of a variety of protocols. The overall effect of such methods is to maintain privacy for each individual entity, while deriving aggregate results over the entire data.

- *Downgrading Application Effectiveness:* In many cases, even though the data may not be available, the output of applications such as association rule mining, classification or query processing may result in violations of privacy. This has lead to research in downgrading the effectiveness of applications by either data or application modifications. Some examples of such techniques include association rule hiding [124], classifier downgrading [92], and query auditing [1].

In this paper, we will provide a broad overview of the different techniques for privacy-preserving data mining. We will provide a review of the major algorithms available for each method, and the variations on the different techniques. We will also discuss a number of combinations of different concepts such as k-anonymous mining over vertically- or horizontally-partitioned data. We will also discuss a number of unique challenges associated with privacy-preserving data mining in the high dimensional case.

This paper is organized as follows. In section 2, we will introduce the randomization method for privacy preserving data mining. In section 3, we will discuss the k-anonymization method along with its different variations. In section 4, we will discuss issues in distributed privacy-preserving data mining. In section 5, we will discuss a number of techniques for privacy which arise in the context of sensitive output of a variety of data mining and data management applications. In section 6, we will discuss some unique challenges associated with privacy in the high dimensional case. A number of applications of privacy-preserving models and algorithms are discussed in Section 7. Section 8 contains the conclusions and discussions.

2.2 The Randomization Method

In this section, we will discuss the randomization method for privacy-preserving data mining. The randomization method has been traditionally used in the context of distorting data by probability distribution for methods such as surveys which have an evasive answer bias because of privacy concerns [74, 129]. This technique has also been extended to the problem of privacy-preserving data mining [2].

The method of randomization can be described as follows. Consider a set of data records denoted by $X = \{x_1 \ldots x_N\}$. For record $x_i \in X$, we add a noise component which is drawn from the probability distribution $f_Y(y)$. These noise components are drawn independently, and are denoted $y_1 \ldots y_N$. Thus, the new set of distorted records are denoted by $x_1 + y_1 \ldots x_N + y_N$. We

denote this new set of records by $z_1 \ldots z_N$. In general, it is assumed that the variance of the added noise is large enough, so that the original record values cannot be easily guessed from the distorted data. Thus, the original records cannot be recovered, but the distribution of the original records can be recovered.

Thus, if X be the random variable denoting the data distribution for the original record, Y be the random variable describing the noise distribution, and Z be the random variable denoting the final record, we have:

$$Z = X + Y$$
$$X = Z - Y$$

Now, we note that N instantiations of the probability distribution Z are known, whereas the distribution Y is known publicly. For a large enough number of values of N, the distribution Z can be approximated closely by using a variety of methods such as kernel density estimation. By subtracting Y from the approximated distribution of Z, it is possible to approximate the original probability distribution X. In practice, one can combine the process of approximation of Z with subtraction of the distribution Y from Z by using a variety of iterative methods such as those discussed in [2, 5]. Such iterative methods typically have a higher accuracy than the sequential solution of first approximating Z and then subtracting Y from it. In particular, the EM method proposed in [5] shows a number of optimal properties in approximating the distribution of X.

We note that at the end of the process, we only have a *distribution* containing the behavior of X. Individual records are not available. Furthermore, the distributions are available only along individual dimensions. Therefore, new data mining algorithms need to be designed to work with the uni-variate distributions rather than the individual records. This can sometimes be a challenge, since many data mining algorithms are inherently dependent on statistics which can only be extracted from either the individual records or the multi-variate probability distributions associated with the records. While the approach can certainly be extended to multi-variate distributions, density estimation becomes inherently more challenging [112] with increasing dimensionalities. For even modest dimensionalities such as 7 to 10, the process of density estimation becomes increasingly inaccurate, and falls prey to the curse of dimensionality.

One key advantage of the randomization method is that it is relatively simple, and does not require knowledge of the distribution of other records in the data. This is not true of other methods such as k-anonymity which require the knowledge of other records in the data. Therefore, the randomization method can be implemented at *data collection time*, and does not require the use of a trusted server containing all the original records in order to perform the anonymization process. While this is a strength of the randomization method,

it also leads to some weaknesses, since it treats all records equally irrespective of their local density. Therefore, outlier records are more susceptible to adversarial attacks as compared to records in more dense regions in the data [10]. In order to guard against this, one may need to be needlessly more aggressive in adding noise to all the records in the data. This reduces the utility of the data for mining purposes.

The randomization method has been extended to a variety of data mining problems. In [2], it was discussed how to use the approach for classification. A number of other techniques [143, 145] have also been proposed which seem to work well over a variety of different classifiers. Techniques have also been proposed for privacy-preserving methods of improving the effectiveness of classifiers. For example, the work in [51] proposes methods for privacy-preserving boosting of classifiers. Methods for privacy-preserving mining of association rules have been proposed in [47, 107]. The problem of association rules is especially challenging because of the discrete nature of the attributes corresponding to presence or absence of items. In order to deal with this issue, the randomization technique needs to be modified slightly. Instead of adding quantitative noise, random items are dropped or included with a certain probability. The perturbed transactions are then used for aggregate association rule mining. This technique has shown to be extremely effective in [47]. The randomization approach has also been extended to other applications such as OLAP [3], and SVD based collaborative filtering [103].

2.2.1 Privacy Quantification

The quantity used to measure privacy should indicate how closely the original value of an attribute can be estimated. The work in [2] uses a measure that defines privacy as follows: If the original value can be estimated with $c\%$ confidence to lie in the interval $[\alpha_1, \alpha_2]$, then the interval width $(\alpha_2 - \alpha_1)$ defines the amount of privacy at $c\%$ confidence level. For example, if the perturbing additive is uniformly distributed in an interval of width 2α, then α is the amount of privacy at confidence level 50% and 2α is the amount of privacy at confidence level 100%. However, this simple method of determining privacy can be subtly incomplete in some situations. This can be best explained by the following example.

EXAMPLE 2.1 *Consider an attribute X with the density function $f_X(x)$ given by:*

$$f_X(x) = 0.5 \quad 0 \le x \le 1$$
$$0.5 \quad 4 \le x \le 5$$
$$0 \quad otherwise$$

Assume that the perturbing additive Y is distributed uniformly between $[-1, 1]$. Then according to the measure proposed in [2], the amount of privacy is 2 at confidence level 100%.

However, after performing the perturbation and subsequent reconstruction, the density function $f_X(x)$ will be approximately revealed. Let us assume for a moment that a large amount of data is available, so that the distribution function is revealed to a high degree of accuracy. Since the (distribution of the) perturbing additive is publically known, the two pieces of information can be combined to determine that if $Z \in [-1, 2]$, then $X \in [0, 1]$; whereas if $Z \in [3, 6]$ then $X \in [4, 5]$.

Thus, in each case, the value of X can be localized to an interval of length 1. This means that the actual amount of privacy offered by the perturbing additive Y is at most 1 at confidence level 100%. We use the qualifier 'at most' since X can often be localized to an interval of length less than one. For example, if the value of Z happens to be -0.5, then the value of X can be localized to an even smaller interval of $[0, 0.5]$.

This example illustrates that the method suggested in [2] does not take into account the distribution of original data. In other words, the (aggregate) reconstruction of the attribute value also provides a certain level of knowledge which can be used to guess a data value to a higher level of accuracy. To accurately quantify privacy, we need a method which takes such side-information into account.

A key privacy measure [5] is based on the *differential entropy* of a random variable. The differential entropy $h(A)$ of a random variable A is defined as follows:

$$h(A) = -\int_{\Omega_A} f_A(a) \log_2 f_A(a) \, da \qquad (2.1)$$

where Ω_A is the domain of A. It is well-known that $h(A)$ is a measure of uncertainty inherent in the value of A [111]. It can be easily seen that for a random variable U distributed uniformly between 0 and a, $h(U) = \log_2(a)$. For $a = 1$, $h(U) = 0$.

In [5], it was proposed that $2^{h(A)}$ is a measure of privacy inherent in the random variable A. This value is denoted by $\Pi(A)$. Thus, a random variable U distributed uniformly between 0 and a has privacy $\Pi(U) = 2^{log_2(a)} = a$. For a general random variable A, $\Pi(A)$ denote the length of the interval, over which a uniformly distributed random variable has the same uncertainty as A.

Given a random variable B, the *conditional* differential entropy of A is defined as follows:

$$h(A|B) = -\int_{\Omega_{A,B}} f_{A,B}(a, b) \log_2 f_{A|B=b}(a) \, da \, db \qquad (2.2)$$

Thus, the average conditional privacy of A given B is $\Pi(A|B) = 2^{h(A|B)}$. This motivates the following metric $\mathcal{P}(A|B)$ for the conditional privacy loss of A, given B:

$$\mathcal{P}(A|B) = 1 - \Pi(A|B)/\Pi(A) = 1 - 2^{h(A|B)}/2^{h(A)} = 1 - 2^{-I(A;B)}.$$

where $I(A;B) = h(A) - h(A|B) = h(B) - h(B|A)$. $I(A;B)$ is also known as the *mutual information* between the random variables A and B. Clearly, $\mathcal{P}(A|B)$ is the fraction of privacy of A which is lost by revealing B.

As an illustration, let us reconsider Example 2.1 given above. In this case, the differential entropy of X is given by:

$$h(X) = -\int_{\Omega_X} f_X(x) \log_2 f_X(x) \, dx$$
$$= -\int_0^1 0.5 \log_2 0.5 \, dx - \int_4^5 0.5 \log_2 0.5 \, dx$$
$$= 1$$

Thus the privacy of X, $\Pi(X) = 2^1 = 2$. In other words, X has as much privacy as a random variable distributed uniformly in an interval of length 2. The density function of the perturbed value Z is given by $f_Z(z) = \int_{-\infty}^{\infty} f_X(\nu) f_Y(z - \nu) \, d\nu$.

Using $f_Z(z)$, we can compute the differential entropy $h(Z)$ of Z. It turns out that $h(Z) = 9/4$. Therefore, we have:

$$I(X;Z) = h(Z) - h(Z|X) = 9/4 - h(Y) = 9/4 - 1 = 5/4$$

Here, the second equality $h(Z|X) = h(Y)$ follows from the fact that X and Y are independent and $Z = X + Y$. Thus, the fraction of privacy loss in this case is $\mathcal{P}(X|Z) = 1 - 2^{-5/4} = 0.5796$. Therefore, after revealing Z, X has privacy $\Pi(X|Z) = \Pi(X) \times (1 - \mathcal{P}(X|Z)) = 2 \times (1.0 - 0.5796) = 0.8408$. This value is less than 1, since X can be localized to an interval of length less than one for many values of Z.

The problem of privacy quantification has been studied quite extensively in the literature, and a variety of metrics have been proposed to quantify privacy. A number of quantification issues in the measurement of privacy breaches has been discussed in [46, 48]. In [19], the problem of privacy-preservation has been studied from the broader context of the tradeoff between the privacy and the information loss. We note that the quantification of privacy alone is not sufficient without quantifying the utility of the data created by the randomization process. A framework has been proposed to explore this tradeoff for a variety of different privacy transformation algorithms.

2.2.2   Adversarial Attacks on Randomization

In the earlier section on privacy quantification, we illustrated an example in which the reconstructed distribution on the data can be used in order to reduce the privacy of the underlying data record. In general, a systematic approach can be used to do this in multi-dimensional data sets with the use of spectral filtering or PCA based techniques [54, 66]. The broad idea in techniques such as PCA [54] is that the correlation structure in the original data can be estimated fairly accurately (in larger data sets) even after noise addition. Once the broad correlation structure in the data has been determined, one can then try to remove the noise in the data in such a way that it fits the aggregate correlation structure of the data. It has been shown that such techniques can reduce the privacy of the perturbation process significantly since the noise removal results in values which are fairly close to their original values [54, 66]. Some other discussions on limiting breaches of privacy in the randomization method may be found in [46].

A second kind of adversarial attack is with the use of public information. Consider a record $X = (x_1 \ldots x_d)$, which is perturbed to $Z = (z_1 \ldots z_d)$. Then, since the distribution of the perturbations is known, we can try to use a maximum likelihood fit of the *potential perturbation* of Z to a public record. Consider the publicly public record $W = (w_1 \ldots w_d)$. Then, the *potential perturbation* of Z with respect to W is given by $(Z - W) = (z_1 - w_1 \ldots z_d - w_d)$. Each of these values $(z_i - w_i)$ should fit the distribution $f_Y(y)$. The corresponding log-likelihood fit is given by $-\sum_{i=1}^{d} \log(f_y(z_i - w_i))$. The higher the log-likelihood fit, the greater the probability that the record W corresponds to X. If it is known that the public data set always includes X, then the maximum likelihood fit can provide a high degree of certainty in identifying the correct record, especially in cases where d is large. We will discuss this issue in greater detail in a later section.

2.2.3   Randomization Methods for Data Streams

The randomization approach is particularly well suited to privacy-preserving data mining of streams, since the noise added to a given record is independent of the rest of the data. However, streams provide a particularly vulnerable target for adversarial attacks with the use of PCA based techniques [54] because of the large volume of the data available for analysis. In [78], an interesting technique for randomization has been proposed which uses the auto-correlations in different time series while deciding the noise to be added to any particular value. It has been shown in [78] that such an approach is more robust since the noise correlates with the stream behavior, and it is more difficult to create effective adversarial attacks with the use of correlation analysis techniques.

2.2.4   Multiplicative Perturbations

The most common method of randomization is that of additive perturbations. However, multiplicative perturbations can also be used to good effect for privacy-preserving data mining. Many of these techniques derive their roots in the work of [61] which shows how to use multi-dimensional projections in order to reduce the dimensionality of the data. This technique preserves the inter-record distances approximately, and therefore the transformed records can be used in conjunction with a variety of data mining applications. In particular, the approach is discussed in detail in [97, 98], in which it is shown how to use the method for privacy-preserving clustering. The technique can also be applied to the problem of classification as discussed in [28]. Multiplicative perturbations can also be used for distributed privacy-preserving data mining. Details can be found in [81]. A number of techniques for multiplicative perturbation in the context of masking census data may be found in [70]. A variation on this theme may be implemented with the use of distance preserving fourier transforms, which work effectively for a variety of cases [91].

As in the case of additive perturbations, multiplicative perturbations are not entirely safe from adversarial attacks. In general, if the attacker has no prior knowledge of the data, then it is relatively difficult to attack the privacy of the transformation. However, with some prior knowledge, two kinds of attacks are possible [82]:

- **Known Input-Output Attack:** In this case, the attacker knows some linearly independent collection of records, and their corresponding perturbed version. In such cases, linear algebra techniques can be used to reverse-engineer the nature of the privacy preserving transformation.

- **Known Sample Attack:** In this case, the attacker has a collection of independent data samples from the same distribution from which the original data was drawn. In such cases, principal component analysis techniques can be used in order to reconstruct the behavior of the original data.

2.2.5   Data Swapping

We note that noise addition or multiplication is not the only technique which can be used to perturb the data. A related method is that of data swapping, in which the values across different records are swapped in order to perform the privacy-preservation [49]. One advantage of this technique is that the lower order marginal totals of the data are completely preserved and are not perturbed at all. Therefore certain kinds of aggregate computations can be exactly performed without violating the privacy of the data. We note that this technique does not follow the general principle in randomization which allows the

value of a record to be perturbed independent;y of the other records. There-fore, this technique can be used in combination with other frameworks such as k-anonymity, as long as the swapping process is designed to preserve the definitions of privacy for that model.

2.3   Group Based Anonymization

The randomization method is a simple technique which can be easily im-plemented at *data collection time*, because the noise added to a given record is independent of the behavior of other data records. This is also a weakness be-cause outlier records can often be difficult to mask. Clearly, in cases in which the privacy-preservation does not need to be performed at data-collection time, it is desirable to have a technique in which the level of inaccuracy depends upon the behavior of the locality of that given record. Another key weakness of the randomization framework is that it does not consider the possibility that publicly available records can be used to identify the identity of the owners of that record. In [10], it has been shown that the use of publicly available records can lead to the privacy getting heavily compromised in high-dimensional cases. This is especially true of outlier records which can be easily distinguished from other records in their locality. Therefore, a broad approach to many privacy transformations is to construct groups of anonymous records which are trans-formed in a group-specific way.

2.3.1   The k-Anonymity Framework

In many applications, the data records are made available by simply remov-ing key identifiers such as the name and social-security numbers from personal records. However, other kinds of attributes (known as pseudo-identifiers) can be used in order to accurately identify the records. Foe example, attributes such as age, zip-code and sex are available in public records such as census rolls. When these attributes are also available in a given data set, they can be used to infer the identity of the corresponding individual. A combination of these attributes can be very powerful, since they can be used to narrow down the possibilities to a small number of individuals.

In k-anonymity techniques [110], we reduce the granularity of representa-tion of these pseudo-identifiers with the use of techniques such as *general-ization* and *suppression*. In the method of *generalization*, the attribute values are generalized to a range in order to reduce the granularity of representation. For example, the date of birth could be generalized to a range such as year of birth, so as to reduce the risk of identification. In the method of *suppression*, the value of the attribute is removed completely. It is clear that such methods reduce the risk of identification with the use of public records, while reducing the accuracy of applications on the transformed data.

In order to reduce the risk of identification, the k-anonymity approach requires that every tuple in the table be indistinguishability related to no fewer than k respondents. This can be formalized as follows:

DEFINITION 2.2 *Each release of the data must be such that every combination of values of quasi-identifiers can be indistinguishably matched to at least k respondents.*

The first algorithm for k-anonymity was proposed in [110]. The approach uses *domain generalization hierarchies* of the quasi-identifiers in order to build k-anonymous tables. The concept of k-minimal generalization has been proposed in [110] in order to limit the level of generalization for maintaining as much data precision as possible for a given level of anonymity. Subsequently, the topic of k-anonymity has been widely researched. A good overview and survey of the corresponding algorithms may be found in [31].

We note that the problem of optimal anonymization is inherently a difficult one. In [89], it has been shown that the problem of optimal k-anonymization is NP-hard. Nevertheless, the problem can be solved quite effectively by the use of a number of heuristic methods. A method proposed by Bayardo and Agrawal [18] is the k-*Optimize* algorithm which can often obtain effective solutions.

The approach assumes an ordering among the quasi-identifier attributes. The values of the attributes are discretized into intervals (quantitative attributes) or grouped into different sets of values (categorical attributes). Each such grouping is an *item*. For a given attribute, the corresponding items are also ordered. An index is created using these attribute-interval pairs (or items) and a set enumeration tree is constructed on these attribute-interval pairs. This set enumeration tree is a systematic enumeration of all possible generalizations with the use of these groupings. The root of the node is the null node, and every successive level of the tree is constructed by appending one item which is lexicographically larger than all the items at that node of the tree. We note that the number of possible nodes in the tree increases exponentially with the data dimensionality. Therefore, it is not possible to build the entire tree even for modest values of n. However, the k-Optimize algorithm can use a number of pruning strategies to good effect. In particular, a node of the tree can be pruned when it is determined that no descendent of it could be optimal. This can be done by computing a bound on the quality of all descendents of that node, and comparing it to the quality of the current best solution obtained during the traversal process. A branch and bound technique can be used to successively improve the quality of the solution during the traversal process. Eventually, it is possible to terminate the algorithm at a maximum computational time, and use the current solution at that point, which is often quite good, but may not be optimal.

In [75], the *Incognito* method has been proposed for computing a k-minimal generalization with the use of bottom-up aggregation along domain generalization hierarchies. The Incognito method uses a bottom-up breadth-first search of the domain generalization hierarchy, in which it generates all the possible minimal k-anonymous tables for a given private table. First, it checks k-anonymity for each single attribute, and removes all those generalizations which do not satisfy k-anonymity. Then, it computes generalizations in pairs, again pruning those pairs which do not satisfy the k-anonymity constraints. In general, the Incognito algorithm computes $(i + 1)$-dimensional generalization *candidates* from the i-dimensional generalizations, and removes all those those generalizations which do not satisfy the k-anonymity constraint. This approach is continued until, no further candidates can be constructed, or all possible dimensions have been exhausted. We note that the methods in [76, 75] use a more general model for k-anonymity than that in [110]. This is because the method in [110] assumes that the value generalization hierarchy is a tree, whereas that in [76, 75] assumes that it is a graph.

Two interesting methods for top-down specialization and bottom-up generalization for k-anonymity have been proposed in [50, 125]. In [50], a top-down heuristic is designed, which starts with a general solution, and then specializes some attributes of the current solution so as to increase the information, but reduce the anonymity. The reduction in anonymity is always controlled, so that k-anonymity is never violated. At the same time each step of the specialization is controlled by a goodness metric which takes into account both the gain in information and the loss in anonymity. A complementary method to top down specialization is that of *bottom up generalization*, for which an interesting method is proposed in [125].

We note that generalization and suppression are not the only transformation techniques for implementing k-anonymity. For example in [38] it is discussed how to use micro-aggregation in which clusters of records are constructed. For each cluster, its representative value is the average value along each dimension in the cluster. A similar method for achieving anonymity via clustering is proposed in [15]. The work in [15] also provides constant factor approximation algorithms to design the clustering. In [8], a related method has been independently proposed for condensation based privacy-preserving data mining. This technique generates pseudo-data from clustered groups of k-records. The process of pseudo-data generation uses principal component analysis of the behavior of the records within a group. It has been shown in [8], that the approach can be effectively used for the problem of classification. We note that the use of pseudo-data provides an additional layer of protection, since it is difficult to perform adversarial attacks on synthetic data. At the same time, the aggregate behavior of the data is preserved, and this can be useful for a variety of data mining problems.

Since the problem of k-anonymization is essentially a search over a space of possible multi-dimensional solutions, standard heuristic search techniques such as genetic algorithms or simulated annealing can be effectively used. Such a technique has been proposed in [130] in which a simulated annealing algorithm is used in order to generate k-anonymous representations of the data. Another technique proposed in [59] uses genetic algorithms in order to construct k-anonymous representations of the data. Both of these techniques require high computational times, and provide no guarantees on the quality of the solutions found.

The only known techniques which provide guarantees on the quality of the solution are *approximation algorithms* [13, 14, 89], in which the solution found is guaranteed to be within a certain factor of the cost of the optimal solution. An approximation algorithm for k-anonymity was proposed in [89], and it provides an $O(k \cdot \log k)$ optimal solution. A number of techniques have also been proposed in [13, 14], which provide $O(k)$-approximations to the optimal cost k-anonymous solutions. In [100], a large improvement was proposed over these different methods. The technique in [100] proposes an $O(\log(k))$-approximation algorithm. This is significantly better than competing algorithms. Furthermore, the work in [100] also proposes a $O(\beta \cdot \log(k))$ approximation algorithm, where the parameter β can be gracefully adjusted based on running time constraints. Thus, this approach not only provides an approximation algorithm, but also gracefully explores the tradeoff between accuracy and running time.

In many cases, associations between pseudo-identifiers and sensitive attributes can be protected by using multiple views, such that the pseudo-identifiers and sensitive attributes occur in different views of the table. Thus, only a small subset of the selected views may be made available. It may be possible to achieve k-anonymity because of the lossy nature of the join across the two views. In the event that the join is not lossy enough, it may result in a violation of k-anonymity. In [140], the problem of violation of k-anonymity using multiple views has been studied. It has been shown that the problem is NP-hard in general. It has been shown in [140] that a polynomial time algorithm is possible if functional dependencies exist between the different views.

An interesting analysis of the safety of k-anonymization methods has been discussed in [73]. It tries to model the effectiveness of a k-anonymous representation, given that the attacker has some prior knowledge about the data such as a sample of the original data. Clearly, the more similar the sample data is to the true data, the greater the risk. The technique in [73] uses this fact to construct a model in which it calculates the expected number of items identified. This kind of technique can be useful in situations where it is desirable

to determine whether or not anonymization should be used as the technique of choice for a particular situation.

2.3.2 Personalized Privacy-Preservation

Not all individuals or entities are equally concerned about their privacy. For example, a corporation may have very different constraints on the privacy of its records as compared to an individual. This leads to the natural problem that we may wish to treat the records in a given data set very differently for anonymization purposes. From a technical point of view, this means that the value of k for anonymization is not fixed but may vary with the record. A condensation-based approach [9] has been proposed for privacy-preserving data mining in the presence of variable constraints on the privacy of the data records. This technique constructs groups of non-homogeneous size from the data, such that it is guaranteed that each record lies in a group whose size is at least equal to its anonymity level. Subsequently, pseudo-data is generated from each group so as to create a synthetic data set with the same aggregate distribution as the original data.

Another interesting model of personalized anonymity is discussed in [132] in which a person can specify the level of privacy for his or her *sensitive values*. This technique assumes that an individual can specify a node of the domain generalization hierarchy in order to decide the level of anonymity that he can work with. This approach has the advantage that it allows for direct protection of the sensitive values of individuals than a vanilla k-anonymity method which is susceptible to different kinds of attacks.

2.3.3 Utility Based Privacy Preservation

The process of privacy-preservation leads to loss of information for data mining purposes. This loss of information can also be considered a loss of *utility* for data mining purposes. Since some negative results [7] on the curse of dimensionality suggest that a lot of attributes may need to be suppressed in order to preserve anonymity, it is extremely important to do this carefully in order to preserve utility. We note that many anonymization methods [18, 50, 83, 126] use cost measures in order to measure the information loss from the anonymization process. examples of such utility measures include generalization height [18], size of anonymized group [83], discernability measures of attribute values [18], and privacy information loss ratio[126]. In addition, a number of metrics such as the classification metric [59] explicitly try to perform the privacy-preservation in such a way so as to tailor the results with use for specific applications such as classification.

The problem of utility-based privacy-preserving data mining was first studied formally in [69]. The broad idea in [69] is to ameliorate the curse of

dimensionality by separately publishing marginal tables containing attributes which have utility, but are also problematic for privacy-preservation purposes. The generalizations performed on the marginal tables and the original tables in fact do not need to be the same. It has been shown that this broad approach can preserve considerable utility of the data set without violating privacy.

A method for utility-based data mining using local recoding was proposed in [135]. The approach is based on the fact that different attributes have different utility from an application point of view. Most anonymization methods are *global*, in which a particular tuple value is mapped to the same generalized value globally. In local recoding, the data space is partitioned into a number of regions, and the mapping of the tuple to the generalizes value is local to that region. Clearly, this kind of approach has greater flexibility, since it can tailor the generalization process to a particular region of the data set. In [135], it has been shown that this method can perform quite effectively because of its local recoding strategy.

Another indirect approach to utility based anonymization is to make the privacy-preservation algorithms more aware of the workload [77]. Typically, data recipients may request only a subset of the data in many cases, and the union of these different requested parts of the data set is referred to as the workload. Clearly, a workload in which some records are used more frequently than others tends to suggest a different anonymization than one which is based on the entire data set. In [77], an effective and efficient algorithm has been proposed for workload aware anonymization.

Another direction for utility based privacy-preserving data mining is to anonymize the data in such a way that it remains useful for particular kinds of data mining or database applications. In such cases, the utility measure is often affected by the underlying application at hand. For example, in [50], a method has been proposed for k-anonymization using an information-loss metric as the utility measure. Such an approach is useful for the problem of classification. In [72], a method has been proposed for anonymization, so that the accuracy of the underlying queries is preserved.

2.3.4   Sequential Releases

Privacy-preserving data mining poses unique problems for dynamic applications such as data streams because in such cases, the data is released sequentially. In other cases, different views of the table may be released sequentially. Once a data block is released, it is no longer possible to go back and increase the level of generalization. On the other hand, new releases may sharpen an attacker's view of the data and may make the overall data set more susceptible to attack. For example, when different views of the data are released sequentially, then one may use a join on the two releases [127] in order to sharpen the

ability to distinguish particular records in the data. A technique discussed in [127] relies on lossy joins in order to cripple an attack based on global quasi-identifiers. The intuition behind this approach is that if the join is lossy enough, it will reduce the confidence of the attacker in relating the release from previous views to the current release. Thus, the inability to link successive releases is key in preventing further discovery of the identity of records.

While the work in [127] explores the issue of sequential releases from the point of view of adding additional attributes, the work in [134] discusses the same issue when records are added to or deleted from the original data. A new generalization principle called m-invariance is proposed, which effectively limits the risk of privacy-disclosure in re-publication. Another method for handling sequential updates to the data set is discussed in [101]. The broad idea in this approach is to progressively and consistently increase the generalization granularity, so that the released data satisfies the k-anonymity requirement both with respect to the current table, as well as with respect to the previous releases.

2.3.5 The l-diversity Method

The k-anonymity is an attractive technique because of the simplicity of the definition and the numerous algorithms available to perform the anonymization. Nevertheless the technique is susceptible to many kinds of attacks especially when background knowledge is available to the attacker. Some kinds of such attacks are as follows:

- **Homogeneity Attack:** In this attack, all the values for a sensitive attribute within a group of k records are the same. Therefore, even though the data is k-anonymized, the value of the sensitive attribute for that group of k records can be predicted exactly.

- **Background Knowledge Attack:** In this attack, the adversary can use an association between one or more quasi-identifier attributes with the sensitive attribute in order to narrow down possible values of the sensitive field further. An example given in [83] is one in which background knowledge of low incidence of heart attacks among Japanese could be used to narrow down information for the sensitive field of what disease a patient might have. A detailed discussion of the effects of background knowledge on privacy may be found in [88].

Clearly, while k-anonymity is effective in preventing *identification* of a record, it may not always be effective in preventing inference of the sensitive values of the attributes of that record. Therefore, the technique of l-diversity was proposed which not only maintains the minimum group size of k, but also

focusses on maintaining the diversity of the sensitive attributes. Therefore, the l-diversity model [83] for privacy is defined as follows:

DEFINITION 2.3 *Let a q^*-block be a set of tuples such that its non-sensitive values generalize to q^*. A q^*-block is l-diverse if it contains l "well represented" values for the sensitive attribute S. A table is l-diverse, if every q^*-block in it is l-diverse.*

A number of different instantiations for the l-diversity definition are discussed in [83]. We note that when there are multiple sensitive attributes, then the l-diversity problem becomes especially challenging because of the curse of dimensionality. Methods have been proposed in [83] for constructing l-diverse tables from the data set, though the technique remains susceptible to the curse of dimensionality [7]. Other methods for creating l-diverse tables are discussed in [133], in which a simple and efficient method for constructing the l-diverse representation is proposed.

2.3.6   The t-closeness Model

The t-closeness model is a further enhancement on the concept of l-diversity. One characteristic of the l-diversity model is that it treats all values of a given attribute in a similar way irrespective of its distribution in the data. This is rarely the case for real data sets, since the attribute values may be very skewed. This may make it more difficult to create feasible l-diverse representations. Often, an adversary may use background knowledge of the global distribution in order to make inferences about sensitive values in the data. Furthermore, not all values of an attribute are equally sensitive. For example, an attribute corresponding to a disease may be more sensitive when the value is positive, rather than when it is negative. In [79], a t-closeness model was proposed which uses the property that the distance between the distribution of the sensitive attribute within an anonymized group should not be different from the global distribution by more than a threshold t. The Earth Mover distance metric is used in order to quantify the distance between the two distributions. Furthermore, the t-closeness approach tends to be more effective than many other privacy-preserving data mining methods for the case of numeric attributes.

2.3.7   Models for Text, Binary and String Data

Most of the work on privacy-preserving data mining is focussed on numerical or categorical data. However, specific data domains such as strings, text, or market basket data may share specific properties with some of these general data domains, but may be different enough to require their own set of techniques for privacy-preservation. Some examples are as follows:

- **Text and Market Basket Data:** While these can be considered a case of text and market basket data, they are typically too high dimensional to work effectively with standard k-anonymization techniques. However, these kinds of data sets have the special property that they are extremely *sparse*. The sparsity property implies that only a few of the attributes are non-zero, and most of the attributes take on zero values. In [11], techniques have been proposed to construct anonymization methods which take advantage of this sparsity. In particular sketch based methods have been used to construct anonymized representations of the data. Variations are proposed to construct anonymizations which may be used at data collection time.

- **String Data:** String Data is considered challenging because of the variations in the lengths of strings across different records. Typically methods for k-anonymity are attribute specific, and therefore constructions of anonymizations for variable length records are quite difficult. In [12], a condensation based method has been proposed for anonymization of string data. This technique creates clusters from the different strings, and then generates synthetic data which has the same aggregate properties as the individual clusters. Since each cluster contains at least k-records, the anonymized data is guaranteed to at least satisfy the definitions of k-anonymity.

2.4   Distributed Privacy-Preserving Data Mining

The key goal in most distributed methods for privacy-preserving data mining is to allow computation of useful aggregate statistics over the entire data set without compromising the privacy of the individual data sets within the different participants. Thus, the participants may wish to collaborate in obtaining aggregate results, but may not fully trust each other in terms of the distribution of their own data sets. For this purpose, the data sets may either be *horizontally partitioned* or be *vertically partitioned*. In horizontally partitioned data sets, the individual records are spread out across multiple entities, each of which have the same set of attributes. In vertical partitioning, the individual entities may have different attributes (or views) of the same set of records. Both kinds of partitioning pose different challenges to the problem of distributed privacy-preserving data mining.

The problem of distributed privacy-preserving data mining overlaps closely with a field in cryptography for determining secure multi-party computations. A broad overview of the intersection between the fields of cryptography and privacy-preserving data mining may be found in [102]. The broad approach to cryptographic methods tends to compute functions over inputs provided by multiple recipients without actually sharing the inputs with one another. For

example, in a 2-party setting, Alice and Bob may have two inputs x and y respectively, and may wish to both compute the function $f(x, y)$ without revealing x or y to each other. This problem can also be generalized across k parties by designing the k argument function $h(x_1 \ldots x_k)$. Many data mining algorithms may be viewed in the context of repetitive computations of many such primitive functions such as the scalar dot product, secure sum etc. In order to compute the function $f(x, y)$ or $h(x_1 \ldots, x_k)$, a *protocol* will have to designed for exchanging information in such a way that the function is computed without compromising privacy. We note that the robustness of the protocol depends upon the level of trust one is willing to place on the two participants Alice and Bob. This is because the protocol may be subjected to various kinds of adversarial behavior:

- **Semi-honest Adversaries:** In this case, the participants Alice and Bob are curious and attempt to learn from the information received by them during the protocol, but do not deviate from the protocol themselves. In many situations, this may be considered a realistic model of adversarial behavior.

- **Malicious Adversaries:** In this case, Alice and Bob may vary from the protocol, and may send sophisticated inputs to one another to learn from the information received from each other.

A key building-block for many kinds of secure function evaluations is the 1 out of 2 oblivious-transfer protocol. This protocol was proposed in [45, 105] and involves two parties: a *sender*, and a *receiver*. The sender's input is a pair (x_0, x_1), and the receiver's input is a bit value $\sigma \in \{0, 1\}$. At the end of the process, the receiver learns x_σ only, and the sender learns nothing. A number of simple solutions can be designed for this task. In one solution [45, 53], the receiver generates two random public keys, K_0 and K_1, but the receiver knows only the decryption key for K_σ. The receiver sends these keys to the sender, who encrypts x_0 with K_0, x_1 with K_1, and sends the encrypted data back to the receiver. At this point, the receiver can only decrypt x_σ, since this is the only input for which they have the decryption key. We note that this is a semi-honest solution, since the intermediate steps require an assumption of trust. For example, it is assumed that when the receiver sends two keys to the sender, they indeed know the decryption key to only one of them. In order to deal with the case of malicious adversaries, one must ensure that the sender chooses the public keys according to the protocol. An efficient method for doing so is described in [94]. In [94], generalizations of the 1 out of 2 oblivious transfer protocol to the 1 out N case and k out of N case are described.

Since the oblivious transfer protocol is used as a building block for secure multi-party computation, it may be repeated many times over a given function

evaluation. Therefore, the computational effectiveness of the approach is important. Efficient methods for both semi-honest and malicious adversaries are discussed in [94]. More complex problems in this domain include the computation of probabilistic functions over a number of multi-party inputs [137]. Such powerful techniques can be used in order to abstract out the primitives from a number of computationally intensive data mining problems. Many of the above techniques have been described for the 2-party case, though generic solutions also exist for the multiparty case. Some important solutions for the multiparty case may be found in [25].

The oblivious transfer protocol can be used in order to compute several data mining primitives related to vector distances in multi-dimensional space. A classic problem which is often used as a primitive for many other problems is that of computing the scalar dot-product in a distributed environment [58]. A fairly general set of methods in this direction are described in [39]. Many of these techniques work by sending changed or encrypted versions of the inputs to one another in order to compute the function with the different alternative versions followed by an oblivious transfer protocol to retrieve the correct value of the final output. A systematic framework is described in [39] to transform normal data mining problems to secure multi-party computation problems. The problems discussed in [39] include those of clustering, classification, association rule mining, data summarization, and generalization. A second set of methods for distributed privacy-preserving data mining is discussed in [32] in which the secure multi-party computation of a number of important data mining primitives is discussed. These methods include the secure sum, the secure set union, the secure size of set intersection and the scalar product. These techniques can be used as data mining primitives for secure multi-party computation over a variety of horizontally and vertically partitioned data sets. Next, we will discuss algorithms for secure multi-party computation over horizontally partitioned data sets.

2.4.1   Distributed Algorithms over Horizontally Partitioned Data Sets

In horizontally partitioned data sets, different sites contain different sets of records with the same (or highly overlapping) set of attributes which are used for mining purposes. Many of these techniques use specialized versions of the general methods discussed in [32, 39] for various problems. The work in [80] discusses the construction of a popular decision tree induction method called ID3 with the use of approximations of the best splitting attributes. Subsequently, a variety of classifiers have been generalized to the problem of horizontally-partitioned privacy preserving mining including the Naive Bayes Classifier [65], and the SVM Classifier with nonlinear kernels [141].

An extreme solution for the horizontally partitioned case is discussed in [139], in which privacy-preserving classification is performed in a *fully* distributed setting, where each customer has private access to only their own record. A host of other data mining applications have been generalized to the problem of horizontally partitioned data sets. These include the applications of association rule mining [64], clustering [57, 62, 63] and collaborative filtering [104]. Methods for cooperative statistical analysis using secure multi-party computation methods are discussed in [40, 41].

A related problem is that of information retrieval and document indexing in a network of content providers. This problem arises in the context of multiple providers which may need to cooperate with one another in sharing their content, but may essentially be business competitors. In [17], it has been discussed how an adversary may use the output of search engines and content providers in order to reconstruct the documents. Therefore, the level of trust required grows with the number of content providers. A solution to this problem [17] constructs a centralized privacy-preserving index in conjunction with a distributed access control mechanism. The privacy-preserving index maintains strong privacy guarantees even in the face of colluding adversaries, and even if the entire index is made public.

2.4.2 Distributed Algorithms over Vertically Partitioned Data

For the vertically partitioned case, many primitive operations such as computing the scalar product or the secure set size intersection can be useful in computing the results of data mining algorithms. For example, the methods in [58] discuss how to use to scalar dot product computation for frequent itemset counting. The process of counting can also be achieved by using the secure size of set intersection as described in [32]. Another method for association rule mining discussed in [119] uses the secure scalar product over the vertical bit representation of itemset inclusion in transactions, in order to compute the frequency of the corresponding itemsets. This key step is applied repeatedly within the framework of a roll up procedure of itemset counting. It has been shown in [119] that this approach is quite effective in practice.

The approach of vertically partitioned mining has been extended to a variety of data mining applications such as decision trees [122], SVM Classification [142], Naive Bayes Classifier [121], and k-means clustering [120]. A number of theoretical results on the ability to learn different kinds of functions in vertically partitioned databases with the use of cryptographic approaches are discussed in [42].

2.4.3   Distributed Algorithms for k-Anonymity

In many cases, it is important to maintain k-anonymity across different distributed parties. In [60], a k-anonymous protocol for data which is vertically partitioned across two parties is described. The broad idea is for the two parties to agree on the quasi-identifier to generalize to the same value before release. A similar approach is discussed in [128], in which the two parties agree on how the generalization is to be performed before release.

In [144], an approach has been discussed for the case of horizontally partitioned data. The work in [144] discusses an extreme case in which each site is a customer which owns exactly one tuple from the data. It is assumed that the data record has both sensitive attributes and quasi-identifier attributes. The solution uses encryption on the sensitive attributes. The sensitive values can be decrypted only if therefore are at least k records with the same values on the quasi-identifiers. Thus, k-anonymity is maintained.

The issue of k-anonymity is also important in the context of hiding identification in the context of distributed location based services [20, 52]. In this case, k-anonymity of the user-identity is maintained even when the location information is released. Such location information is often released when a user may send a message at any point from a given location.

A similar issue arises in the context of communication protocols in which the anonymity of senders (or receivers) may need to be protected. A message is said to be *sender k-anonymous*, if it is guaranteed that an attacker can at most narrow down the identity of the sender to k individuals. Similarly, a message is said to be *receiver k-anonymous*, if it is guaranteed that an attacker can at most narrow down the identity of the receiver to k individuals. A number of such techniques have been discussed in [56, 135, 138].

2.5   Privacy-Preservation of Application Results

In many cases, the output of applications can be used by an adversary in order to make significant inferences about the behavior of the underlying data. In this section, we will discuss a number of miscellaneous methods for privacy-preserving data mining which tend to preserve the privacy of the end results of applications such as association rule mining and query processing. This problem is related to that of disclosure control [1] in statistical databases, though advances in data mining methods provide increasingly sophisticated methods for adversaries to make inferences about the behavior of the underlying data. In cases, where the commercial data needs to be shared, the association rules may represent sensitive information for target-marketing purposes, which needs to be protected from inference.

In this section, we will discuss the issue of disclosure control for a number of applications such as association rule mining, classification, and query

processing. The key goal here is to prevent adversaries from making inferences from the end results of data mining and management applications. A broad discussion of the security and privacy implications of data mining are presented in [33]. We will discuss each of the applications below:

2.5.1   Association Rule Hiding

Recent years have seen tremendous advances in the ability to perform association rule mining effectively. Such rules often encode important target marketing information about a business. Some of the earliest work on the challenges of association rule mining for database security may be found in [16]. Two broad approaches are used for association rule hiding:

- **Distortion:** In distortion [99], the entry for a given transaction is modified to a different value. Since, we are typically dealing with binary transactional data sets, the entry value is flipped.

- **Blocking:** In blocking [108], the entry is not modified, but is left incomplete. Thus, unknown entry values are used to prevent discovery of association rules.

We note that both the distortion and blocking processes have a number of side effects on the non-sensitive rules in the data. Some of the non-sensitive rules may be lost along with sensitive rules, and new *ghost rules* may be created because of the distortion or blocking process. Such side effects are undesirable since they reduce the utility of the data for mining purposes.

A formal proof of the NP-hardness of the distortion method for hiding association rule mining may be found in [16]. In [16], techniques are proposed for changing some of the 1-values to 0-values so that the support of the corresponding sensitive rules is appropriately lowered. The utility of the approach was defined by the number of non-sensitive rules whose support was also lowered by using such an approach. This approach was extended in [34] in which both support and confidence of the appropriate rules could be lowered. In this case, 0-values in the transactional database could also change to 1-values. In many cases, this resulted in spurious association rules (or ghost rules) which was an undesirable side effect of the process. A complete description of the various methods for data distortion for association rule hiding may be found in [124]. Another interesting piece of work which balances privacy and disclosure concerns of sanitized rules may be found in [99].

The broad idea of blocking was proposed in [23]. The attractiveness of the blocking approach is that it maintains the truthfulness of the underlying data, since it replaces a value with an unknown (often represented by '?') rather than a false value. Some interesting algorithms for using blocking for association rule hiding are presented in [109]. The work has been further extended in

[108] with a discussion of the effectiveness of reconstructing the hidden rules. Another interesting set of techniques for association rule hiding with limited side effects is discussed in [131]. The objective of this method is to reduce the loss of non-sensitive rules, or the creation of ghost rules during the rule hiding process.

In [6], it has been discussed how blocking techniques for hiding association rules can be used to prevent discovery of sensitive entries in the data set by an adversary. In this case, certain entries in the data are classified as sensitive, and only rules which disclose such entries are hidden. An efficient depth-first association mining algorithm is proposed for this task [6]. It has been shown that the methods can effectively reduce the disclosure of sensitive entries with the use of such a hiding process.

2.5.2   Downgrading Classifier Effectiveness

An important privacy-sensitive application is that of classification, in which the results of a classification application may be sensitive information for the owner of a data set. Therefore the issue is to modify the data in such a way that the accuracy of the classification process is reduced, while retaining the utility of the data for other kinds of applications. A number of techniques have been discussed in [24, 92] in reducing the classifier effectiveness in context of classification rule and decision tree applications. The notion of *parsimonious downgrading* is proposed [24] in the context of blocking out inference channels for classification purposes while mining the effect to the overall utility. A system called Rational Downgrader [92] was designed with the use of these principles.

The methods for association rule hiding can also be generalized to rule based classifiers. This is because rule based classifiers often use association rule mining methods as subroutines, so that the rules with the class labels in their consequent are used for classification purposes. For a classifier downgrading approach, such rules are sensitive rules, whereas all other rules (with non-class attributes in the consequent) are non-sensitive rules. An example of a method for rule based classifier downgradation is discussed in [95] in which it has been shown how to effectively hide classification rules for a data set.

2.5.3   Query Auditing and Inference Control

Many sensitive databases are not available for public access, but may have a public interface through which *aggregate querying* is allowed. This leads to the natural danger that a smart adversary may pose a sequence of queries through which he or she may infer sensitive facts about the data. The nature of this inference may correspond to *full disclosure*, in which an adversary may determine the exact values of the data attributes. A second notion is that of

partial disclosure in which the adversary may be able to narrow down the values to a range, but may not be able to guess the exact value. Most work on query auditing generally concentrates on the full disclosure setting.

Two broad approaches are designed in order to reduce the likelihood of sensitive data discovery:

- **Query Auditing:** In query auditing, we deny one or more queries from a sequence of queries. The queries to be denied are chosen such that the sensitivity of the underlying data is preserved. Some examples of query auditing methods include [37, 68, 93, 106].

- **Query Inference Control:** In this case, we perturb the underlying data or the query result itself. The perturbation is engineered in such a way, so as to preserve the privacy of the underlying data. Examples of methods which use perturbation of the underlying data include [3, 26, 90]. Examples of methods which perturb the query result include [22, 36, 42–44].

An overview of classical methods for query auding may be found in [1]. The query auditing problem has an *online* version, in which we do not know the sequence of queries in advance, and an *offline* version, in which we do know this sequence in advance. Clearly, the offline version is open to better optimization from an auditing point of view.

The problem of query auditing was first studied in [37, 106]. This approach works for the online version of the query auditing problem. In these works, the sum query is studied, and privacy is protected by using restrictions on sizes and pairwise overlaps of the allowable queries. Let us assume that the query size is restricted to be at most k, and the number of common elements in pairwise query sets is at most m. Then, if q be the number of elements that the attacker already knows from background knowledge, it was shown that [37, 106] that the maximum number of queries allowed is $(2 \cdot k - (q + 1))/m$. We note that if N be the total number of data elements, the above expression is always bounded above by $2 \cdot N$. If for some constant c, we choose $k = N/c$ and $m = 1$, the approach can only support a constant number of queries, after which all queries would have to be denied by the auditor. Clearly, this is undesirable from an application point of view. Therefore, a considerable amount of research has been devoted to increasing the number of queries which can be answered by the auditor without compromising privacy.

In [67], the problem of sum auditing on sub-cubes of the data cube are studied, where a query expression is constructed using a string of 0, 1, and *. The elements to be summed up are determined by using matches to the query string pattern. In [71], the problem of auditing a database of boolean values is studied for the case of sum and max queries. In [21], and approach for query auditing

is discussed which is actually a combination of the approach of denying some queries and modifying queries in order to achieve privacy.

In [68], the authors show that denials to queries depending upon the answer to the current query can leak information. The authors introduce the notion of simulatable auditing for auditing sum and max queries. In [93], the authors devise methods for auditing max queries and bags of max and min queries under the partial and full disclosure settings. The authors also examine the notion of *utility* in the context of auditing, and obtain results for sum queries in the full disclosure setting.

A number of techniques have also been proposed for the offline version of the auditing problem. In [29], a number of variations of the offline auditing problem have been studied. In the offline auditing problem, we are given a sequence of queries which have been truthfully answered, and we need to determine if privacy has been breached. In [29], effective algorithms were proposed for the sum, max, and max and min versions of the problems. On the other hand, the sum and max version of the problem was shown to be NP-hard. In [4], an offline auditing framework was proposed for determining whether a database adheres to its disclosure properties. The key idea is to create an audit expression which specifies sensitive table entries.

A number of techniques have also been proposed for sanitizing or randomizing the data for query auditing purposes. These are fairly general models of privacy, since they preserve the privacy of the data even when the entire database is available. The standard methods for perturbation [2, 5] or k-anonymity [110] can always be used, and it is always guaranteed that an adversary may not derive anything more from the queries than they can from the base data. Thus, since a k-anonymity model guarantees a certain level of privacy even when the entire database is made available, it will continue to do so under any sequence of queries. In [26], a number of interesting methods are discussed for measuring the effectiveness of sanitization schemes in terms of balancing privacy and utility.

Instead of sanitizing the base data, it is possible to use summary constructs on the data, and respond to queries using only the information encoded in the summary constructs. Such an approach preserves privacy, as long as the summary constructs do not reveal sensitive information about the underlying records. A histogram based approach to data sanitization has been discussed in [26, 27]. In this technique the data is recursively partitioned into multi-dimensional cells. The final output is the exact description of the cuts along with the population of each cell. Clearly, this kind of description can be used for approximate query answering with the use of standard histogram query processing methods. In [55], a method has been proposed for privacy-preserving indexing of multi-dimensional data by using bucketizing of the underlying attribute values in conjunction with encryption of identification keys.

We note that a choice of larger bucket sizes provides greater privacy but less accuracy. Similarly, optimizing the bucket sizes for accuracy can lead to reductions in privacy. This tradeoff has been studied in [55], and it has been shown that reasonable query precision can be maintained at the expense of partial disclosure.

In the class of methods which use summarization structures for inference control, an interesting method was proposed by Mishra and Sandler in [90], which uses pseudo-random sketches for privacy-preservation. In this technique sketches are constructed from the data, and the sketch representations are used to respond to user queries. In [90], it has been shown that the scheme preserves privacy effectively, while continuing to be useful from a utility point of view.

Finally, an important class of query inference control methods changes the results of queries in order to preserve privacy. A classical method for aggregate queries such as the sum or relative frequency is that of random sampling [35]. In this technique, a random sample of the data is used to compute such aggregate functions. The random sampling approach makes it impossible for the questioner to precisely control the formation of query sets. The advantage of using a random sample is that the results of large queries are quite robust (in terms of *relative error*), but the privacy of individual records are preserved because of high *absolute error*.

Another method for query inference control is by adding noise to the results of queries. Clearly, the noise should be sufficient that an adversary cannot use small changes in the query arguments in order to infer facts about the base data. In [44], an interesting technique has been presented in which the result of a query is perturbed by an amount which depends upon the underlying sensitivity of the query function. This sensitivity of the query function is defined approximately by the change in the response to the query by changing one argument to the function. An important theoretical result [22, 36, 42, 43] shows that a surprisingly small amount of noise needs to be added to the result of a query, provided that the number of queries is sublinear in the number of database rows. With increasing sizes of databases today, this result provides fairly strong guarantees on privacy. Such queries together with their slightly noisy responses are referred to as the SuLQ primitive.

2.6   Limitations of Privacy: The Curse of Dimensionality

Many privacy-preserving data-mining methods are inherently limited by the curse of dimensionality in the presence of public information. For example, the technique in [7] analyzes the k-anonymity method in the presence of increasing dimensionality. The curse of dimensionality becomes especially important when adversaries may have considerable background information, as a result of which the boundary between pseudo-identifiers and sensitive

attributes may become blurred. This is generally true, since adversaries may be familiar with the subject of interest and may have greater information about them than what is publicly available. This is also the motivation for techniques such as l-diversity [83] in which background knowledge can be used to make further privacy attacks. The work in [7] concludes that in order to maintain privacy, a large number of the attributes may need to be suppressed. Thus, the data loses its utility for the purpose of data mining algorithms. The broad intuition behind the result in [7] is that when attributes are generalized into wide ranges, the combination of a large number of generalized attributes is so sparsely populated, that even two anonymity becomes increasingly unlikely. While the method of l-diversity has not been formally analyzed, some observations made in [83] seem to suggest that the method becomes increasingly infeasible to implement effectively with increasing dimensionality.

The method of randomization has also been analyzed in [10]. This paper makes a first analysis of the ability to re-identify data records with the use of maximum likelihood estimates. Consider a d-dimensional record $X = (x_1 \ldots x_d)$, which is perturbed to $Z = (z_1 \ldots z_d)$. For a given public record $W = (w_1 \ldots w_d)$, we would like to find the probability that it could have been perturbed to Z using the perturbing distribution $f_Y(y)$. If this were true, then the set of values given by $(Z - W) = (z_1 - w_1 \ldots z_d - w_d)$ should be all drawn from the distribution $f_Y(y)$. The corresponding log-likelihood fit is given by $-\sum_{i=1}^{d} \log(f_y(z_i - w_i))$. The higher the log-likelihood fit, the greater the probability that the record W corresponds to X. In order to achieve greater anonymity, we would like the perturbations to be large enough, so that some of the spurious records in the data have greater log-likelihood fit to Z than the true record X. It has been shown in [10], that this probability reduces rapidly with increasing dimensionality for different kinds of perturbing distributions. Thus, the randomization technique also seems to be susceptible to the curse of high dimensionality.

We note that the problem of high dimensionality seems to be a fundamental one for privacy preservation, and it is unlikely that more effective methods can be found in order to preserve privacy when background information about a large number of features is available to even a subset of selected individuals. Indirect examples of such violations occur with the use of trail identifications [84, 85], where information from multiple sources can be compiled to create a high dimensional feature representation which violates privacy.

2.7   Applications of Privacy-Preserving Data Mining

The problem of privacy-preserving data mining has numerous applications in homeland security, medical database mining, and customer transaction analysis. Some of these applications such as those involving bio-terrorism

and medical database mining may intersect in scope. In this section, we will discuss a number of different applications of privacy-preserving data mining methods.

2.7.1 Medical Databases: The Scrub and Datafly Systems

The scrub system [118] was designed for de-identification of clinical notes and letters which typically occurs in the form of textual data. Clinical notes and letters are typically in the form of text which contain references to patients, family members, addresses, phone numbers or providers. Traditional techniques simply use a global search and replace procedure in order to provide privacy. However clinical notes often contain cryptic references in the form of abbreviations which may only be understood either by other providers or members of the same institution. Therefore traditional methods can identify no more than 30-60% of the identifying information in the data [118]. The Scrub system uses numerous detection algorithms which compete in parallel to determine when a block of text corresponds to a name, address or a phone number. The Scrub System uses local knowledge sources which compete with one another based on the certainty of their findings. It has been shown in [118] that such a system is able to remove more than 99% of the identifying information from the data.

The Datafly System [117] was one of the earliest practical applications of privacy-preserving transformations. This system was designed to prevent identification of the subjects of medical records which may be stored in multi-dimensional format. The multi-dimensional information may include directly identifying information such as the social security number, or indirectly identifying information such as age, sex or zip-code. The system was designed in response to the concern that the process of removing only directly identifying attributes such as social security numbers was not sufficient to guarantee privacy. While the work has a similar motive as the k-anonymity approach of preventing record identification, it does not formally use a k-anonymity model in order to prevent identification through linkage attacks. The approach works by setting a minimum bin size for each field. The anonymity level is defined in Datafly with respect to this bin size. The values in the records are thus generalized to the ambiguity level of a bin size as opposed to exact values. Directly, identifying attributes such as the social-security-number, name, or zip-code are removed from the data. Furthermore, outlier values are suppressed from the data in order to prevent identification. Typically, the user of Datafly will set the anonymity level depending upon the profile of the data recipient in question. The overall anonymity level is defined between 0 and 1, which defines the minimum bin size for each field. An anonymity level of 0 results in Datafly providing the original data, whereas an anonymity level of 1 results in the

maximum level of generalization of the underlying data. Thus, these two values provide two extreme values of trust and distrust. We note that these values are set depending upon the recipient of the data. When the records are released to the public, it is desirable to set of higher level of anonymity in order to ensure the maximum amount of protection. The generalizations in the datafly system are typically done independently at the individual attribute level, since the bins are defined independently for different attributes. The Datafly system is one of the earliest systems for anonymization, and is quite simple in its approach to anonymization. A lot of work in the anonymity field has been done since the creation of the Datafly system, and there is considerable scope for enhancement of the Datafly system with the use of these models.

2.7.2   Bioterrorism Applications

In typical bioterrorism applications, we would like to analyze medical data for privacy-preserving data mining purposes. Often a biological agent such as anthrax produces symptoms which are similar to other common respiratory diseases such as the cough, cold and the flu. In the absence of prior knowledge of such an attack, health care providers may diagnose a patient affected by an anthrax attack of have symptoms from one of the more common respiratory diseases. The key is to quickly identify a true anthrax attack from a normal outbreak of a common respiratory disease, In many cases, an unusual number of such cases in a given locality may indicate a bio-terrorism attack. Therefore, in order to identify such attacks it is necessary to track incidences of these common diseases as well. Therefore, the corresponding data would need to be reported to public health agencies. However, the common respiratory diseases are not reportable diseases by law. The solution proposed in [114] is that of "selective revelation" which initially allows only limited access to the data. However, in the event of suspicious activity, it allows a "drill-down" into the underlying data. This provides more identifiable information in accordance with public health law.

2.7.3   Homeland Security Applications

A number of applications for homeland security are inherently intrusive because of the very nature of surveillance. In [113], a broad overview is provided on how privacy-preserving techniques may be used in order to deploy these applications effectively without violating user privacy. Some examples of such applications are as follows:

- **Credential Validation Problem:** In this problem, we are trying to match the subject of the credential to the person presenting the credential. For example, the theft of social security numbers presents a serious threat to homeland security. In the credential validation approach [113], an

attempt is made to exploit the semantics associated with the social security number to determine whether the person presenting the SSN credential truly owns it.

■ **Identity Theft:** A related technology [115] is to use a more *active* approach to avoid identity theft. The *identity angel* system [115], crawls through cyberspace, and determines people who are at risk from identity theft. This information can be used to notify appropriate parties. We note that both the above approaches to prevention of identity theft are relatively non-invasive and therefore do not violate privacy.

■ **Web Camera Surveillance:** One possible method for surveillance is with the use of publicly available webcams [113, 116], which can be used to detect unusual activity. We note that this is a much more invasive approach than the previously discussed techniques because of person-specific information being captured in the webcams. The approach can be made more privacy-sensitive by extracting only *facial count* information from the images and using these in order to detect unusual activity. It has been hypothesized in [116] that unusual activity can be detected only in terms of facial count rather than using more specific information about particular individuals. In effect, this kind of approach uses a domain-specific downgrading of the information available in the webcams in order to make the approach privacy-sensitive.

■ **Video-Surveillance:** In the context of sharing video-surveillance data, a major threat is the use of facial recognition software, which can match the facial images in videos to the facial images in a driver license database. While a straightforward solution is to completely black out each face, the result is of limited new, since all facial information has been wiped out. A more balanced approach [96] is to use selective downgrading of the facial information, so that it scientifically limits the ability of facial recognition software to reliably identify faces, while maintaining facial details in images. The algorithm is referred to as k-Same, and the key is to identify faces which are somewhat similar, and then construct new faces which construct combinations of features from these similar faces. Thus, the identity of the underlying individual is anonymized to a certain extent, but the video continues to remain useful. Thus, this approach has the flavor of a k-anonymity approach, except that it creates new synthesized data for the application at hand.

■ **The Watch List Problem:** The motivation behind this problem [113] is that the government typically has a list of known terrorists or suspected entities which it wishes to track from the population. The aim is to view transactional data such as store purchases, hospital admissions, airplane

manifests, hotel registrations or school attendance records in order to identify or track these entities. This is a difficult problem because the transactional data is private, and the privacy of subjects who do not appear in the watch list need to be protected. Therefore, the transactional behavior of non-suspicious subjects may not be identified or revealed. Furthermore, the problem is even more difficult if we assume that the watch list cannot be revealed to the data holders. The second assumption is a result of the fact that members on the watch list may only be suspected entities and should have some level of protection from identification as suspected terrorists to the general public. The watch list problem is currently an open problem [113].

2.7.4 Genomic Privacy

Recent years have seen tremendous advances in the science of DNA sequencing and forensic analysis with the use of DNA. As result, the databases of collected DNA are growing very fast in the both the medical and law enforcement communities. DNA data is considered extremely sensitive, since it contains almost uniquely identifying information about an individual.

As in the case of multi-dimensional data, simple removal of directly identifying data such as social security number is not sufficient to prevent re-identification. In [86], it has been shown that a software called *CleanGene* can determine the identifiability of DNA entries independent of any other demographic or other identifiable information. The software relies on publicly available medical data and knowledge of particular diseases in order to assign identifications to DNA entries. It was shown in [86] that 98-100% of the individuals are identifiable using this approach. The identification is done by taking the DNA sequence of an individual and then constructing a genetic profile corresponding to the sex, genetic diseases, the location where the DNA was collected etc. This genetic profile has been shown in [86] to be quite effective in identifying the individual to a much smaller group. One way to protect the anonymity of such sequences is with the use of *generalization lattices* [87] which are constructed in such a way that an entry in the modified database cannot be distinguished from at least $(k-1)$ other entities. Another approach discussed in [11] constructs synthetic data which preserves the aggregate characteristics of the original data, but preserves the privacy of the original records. Another method for compromising the privacy of genomic data is that of *trail re-identification*, in which the uniqueness of patient visit patterns [84, 85] is exploited in order to make identifications. The premise of this work is that patients often visit and leave behind genomic data at various distributed locations and hospitals. The hospitals usually separate out the clinical data from the genomic data and make the genomic data available for research purposes. While the data is seemingly anonymous, the visit location pattern of the patients is

encoded in the site from which the data is released. It has been shown in [84, 85] that this information may be combined with publicly available data in order to perform unique re-identifications. Some broad ideas for protecting the privacy in such scenarios are discussed in [85].

2.8   Summary

In this paper, we presented a survey of the broad areas of privacy-preserving data mining and the underlying algorithms. We discussed a variety of data modification techniques such as randomization and k-anonymity based techniques. We discussed methods for distributed privacy-preserving mining, and the methods for handling horizontally and vertically partitioned data. We discussed the issue of downgrading the effectiveness of data mining and data management applications such as association rule mining, classification, and query processing. We discussed some fundamental limitations of the problem of privacy-preservation in the presence of increased amounts of public information and background knowledge. Finally, we discussed a number of diverse application domains for which privacy-preserving data mining methods are useful.

References

[1] Adam N., Wortmann J. C.: Security-Control Methods for Statistical Databases: A Comparison Study. *ACM Computing Surveys*, 21(4), 1989.

[2] Agrawal R., Srikant R. Privacy-Preserving Data Mining. *Proceedings of the ACM SIGMOD Conference*, 2000.

[3] Agrawal R., Srikant R., Thomas D. Privacy-Preserving OLAP. *Proceedings of the ACM SIGMOD Conference*, 2005.

[4] Agrawal R., Bayardo R., Faloutsos C., Kiernan J., Rantzau R., Srikant R.: Auditing Compliance via a hippocratic database. *VLDB Conference*, 2004.

[5] Agrawal D. Aggarwal C. C. On the Design and Quantification of Privacy-Preserving Data Mining Algorithms. *ACM PODS Conference*, 2002.

[6] Aggarwal C., Pei J., Zhang B. A Framework for Privacy Preservation against Adversarial Data Mining. *ACM KDD Conference*, 2006.

[7] Aggarwal C. C. On k-anonymity and the curse of dimensionality. *VLDB Conference*, 2005.

[8] Aggarwal C. C., Yu P. S.: A Condensation approach to privacy preserving data mining. *EDBT Conference*, 2004.

[9] Aggarwal C. C., Yu P. S.: On Variable Constraints in Privacy-Preserving Data Mining. *SIAM Conference*, 2005.

[10] Aggarwal C. C.: On Randomization, Public Information and the Curse of Dimensionality. *ICDE Conference*, 2007.

[11] Aggarwal C. C., Yu P. S.: On Privacy-Preservation of Text and Sparse Binary Data with Sketches. *SIAM Conference on Data Mining*, 2007.

[12] Aggarwal C. C., Yu P. S. On Anonymization of String Data. *SIAM Conference on Data Mining*, 2007.

[13] Aggarwal G., Feder T., Kenthapadi K., Motwani R., Panigrahy R., Thomas D., Zhu A.: Anonymizing Tables. *ICDT Conference*, 2005.

[14] Aggarwal G., Feder T., Kenthapadi K., Motwani R., Panigrahy R., Thomas D., Zhu A.: Approximation Algorithms for k-anonymity. *Journal of Privacy Technology*, paper 20051120001, 2005.

[15] Aggarwal G., Feder T., Kenthapadi K., Khuller S., Motwani R., Panigrahy R., Thomas D., Zhu A.: Achieving Anonymity via Clustering. *ACM PODS Conference*, 2006.

[16] Atallah, M., Elmagarmid, A., Ibrahim, M., Bertino, E., Verykios, V.: Disclosure limitation of sensitive rules, *Workshop on Knowledge and Data Engineering Exchange*, 1999.

[17] Bawa M., Bayardo R. J., Agrawal R.: Privacy-Preserving Indexing of Documents on the Network. *VLDB Conference*, 2003.

[18] Bayardo R. J., Agrawal R.: Data Privacy through Optimal k-Anonymization. *Proceedings of the ICDE Conference*, pp. 217–228, 2005.

[19] Bertino E., Fovino I., Provenza L.: A Framework for Evaluating Privacy-Preserving Data Mining Algorithms. *Data Mining and Knowledge Discovery Journal*, 11(2), 2005.

[20] Bettini C., Wang X. S., Jajodia S.: Protecting Privacy against Location Based Personal Identification. *Proc. of Secure Data Management Workshop*, Trondheim, Norway, 2005.

[21] Biskup J., Bonatti P.: Controlled Query Evaluation for Known Policies by Combining Lying and Refusal. *Annals of Mathematics and Artificial Intelligence*, 40(1-2), 2004.

[22] Blum A., Dwork C., McSherry F., Nissim K.: Practical Privacy: The SuLQ Framework. *ACM PODS Conference*, 2005.

[23] Chang L., Moskowitz I.: An integrated framwork for database inference and privacy protection. *Data and Applications Security*. Kluwer, 2000.

[24] Chang L., Moskowitz I.: Parsimonious downgrading and decision trees applied to the inference problem. *New Security Paradigms Workshop*, 1998.

[25] Chaum D., Crepeau C., Damgard I.: Multiparty unconditionally secure protocols. *ACM STOC Conference*, 1988.

[26] Chawla S., Dwork C., McSherry F., Smith A., Wee H.: Towards Privacy in Public Databases, *TCC*, 2005.

[27] Chawla S., Dwork C., McSherry F., Talwar K.: On the Utility of Privacy-Preserving Histograms, *UAI*, 2005.

[28] Chen K., Liu L.: Privacy-preserving data classification with rotation perturbation. *ICDM Conference*, 2005.

[29] Chin F.: Security Problems on Inference Control for SUM, MAX, and MIN Queries. *J. of the ACM*, 33(3), 1986.

[30] Chin F., Ozsoyoglu G.: Auditing for Secure Statistical Databases. *Proceedings of the ACM'81 Conference*, 1981.

[31] Ciriani V., De Capitiani di Vimercati S., Foresti S., Samarati P.: k-Anonymity. *Security in Decentralized Data Management*, ed. Jajodia S., Yu T., Springer, 2006.

[32] Clifton C., Kantarcioglou M., Lin X., Zhu M.: Tools for privacy-preserving distributed data mining. *ACM SIGKDD Explorations*, 4(2), 2002.

[33] Clifton C., Marks D.: Security and Privacy Implications of Data Mining., *Workshop on Data Mining and Knowledge Discovery*, 1996.

[34] Dasseni E., Verykios V., Elmagarmid A., Bertino E.: Hiding Association Rules using Confidence and Support, *4th Information Hiding Workshop*, 2001.

[35] Denning D.: Secure Statistical Databases with Random Sample Queries. *ACM TODS Journal*, 5(3), 1980.

[36] Dinur I., Nissim K.: Revealing Information while preserving privacy. *ACM PODS Conference*, 2003.

[37] Dobkin D., Jones A., Lipton R.: Secure Databases: Protection against User Influence. *ACM Transactions on Databases Systems*, 4(1), 1979.

[38] Domingo-Ferrer J,, Mateo-Sanz J.: Practical data-oriented micro-aggregation for statistical disclosure control. *IEEE TKDE*, 14(1), 2002.

[39] Du W., Atallah M.: Secure Multi-party Computation: A Review and Open Problems.*CERIAS Tech. Report* 2001-51, Purdue University, 2001.

[40] Du W., Han Y. S., Chen S.: Privacy-Preserving Multivariate Statistical Analysis: Linear Regression and Classification, Proc. SIAM Conf. Data Mining, 2004.

[41] Du W., Atallah M.: Privacy-Preserving Cooperative Statistical Analysis, 17th Annual Computer Security Applications Conference, 2001.

[42] Dwork C., Nissim K.: Privacy-Preserving Data Mining on Vertically Partitioned Databases, *CRYPTO*, 2004.

[43] Dwork C., Kenthapadi K., McSherry F., Mironov I., Naor M.: Our Data, Ourselves: Privacy via Distributed Noise Generation. *EUROCRYPT*, 2006.

[44] Dwork C., McSherry F., Nissim K., Smith A.: Calibrating Noise to Sensitivity in Private Data Analysis, *TCC*, 2006.

[45] Even S., Goldreich O., Lempel A.: A Randomized Protocol for Signing Contracts. *Communications of the ACM*, vol 28, 1985.

[46] Evfimievski A., Gehrke J., Srikant R. Limiting Privacy Breaches in Privacy Preserving Data Mining. *ACM PODS Conference*, 2003.

[47] Evfimievski A., Srikant R., Agrawal R., Gehrke J.: Privacy-Preserving Mining of Association Rules. *ACM KDD Conference*, 2002.

[48] Evfimievski A.: Randomization in Privacy-Preserving Data Mining. *ACM SIGKDD Explorations*, 4, 2003.

[49] Fienberg S., McIntyre J.: Data Swapping: Variations on a Theme by Dalenius and Reiss. *Technical Report, National Institute of Statistical Sciences*, 2003.

[50] Fung B., Wang K., Yu P.: Top-Down Specialization for Information and Privacy Preservation. *ICDE Conference*, 2005.

[51] Gambs S., Kegl B., Aimeur E.: Privacy-Preserving Boosting. *Knowledge Discovery and Data Mining Journal*, to appear.

[52] Gedik B., Liu L.: A customizable k-anonymity model for protecting location privacy, *ICDCS Conference*, 2005.

[53] Goldreich O.: Secure Multi-Party Computation, Unpublished Manuscript, 2002.

[54] Huang Z., Du W., Chen B.: Deriving Private Information from Randomized Data. pp. 37–48, *ACM SIGMOD Conference*, 2005.

[55] Hore B., Mehrotra S., Tsudik B.: A Privacy-Preserving Index for Range Queries. *VLDB Conference*, 2004.

[56] Hughes D, Shmatikov V.: Information Hiding, Anonymity, and Privacy: A modular Approach. *Journal of Computer Security*, 12(1), 3–36, 2004.

[57] Inan A., Saygin Y., Savas E., Hintoglu A., Levi A.: Privacy-Preserving Clustering on Horizontally Partitioned Data. *Data Engineering Workshops*, 2006.

[58] Ioannidis I., Grama A., Atallah M.: A secure protocol for computing dot products in clustered and distributed environments, *International Conference on Parallel Processing*, 2002.

[59] Iyengar V. S.: Transforming Data to Satisfy Privacy Constraints. *KDD Conference*, 2002.

[60] Jiang W., Clifton C.: Privacy-preserving distributed k-Anonymity. *Proceedings of the IFIP 11.3 Working Conference on Data and Applications Security*, 2005.

[61] Johnson W., Lindenstrauss J.: Extensions of Lipshitz Mapping into Hilbert Space, *Contemporary Math.* vol. 26, pp. 189-206, 1984.

[62] Jagannathan G., Wright R.: Privacy-Preserving Distributed k-means clustering over arbitrarily partitioned data. *ACM KDD Conference*, 2005.

[63] Jagannathan G., Pillaipakkamnatt K., Wright R.: A New Privacy-Preserving Distributed k-Clustering Algorithm. *SIAM Conference on Data Mining*, 2006.

[64] Kantarcioglu M., Clifton C.: Privacy-Preserving Distributed Mining of Association Rules on Horizontally Partitioned Data. *IEEE TKDE Journal*, 16(9), 2004.

[65] Kantarcioglu M., Vaidya J.: Privacy-Preserving Naive Bayes Classifier for Horizontally Partitioned Data. *IEEE Workshop on Privacy-Preserving Data Mining*, 2003.

[66] Kargupta H., Datta S., Wang Q., Sivakumar K.: On the Privacy Preserving Properties of Random Data Perturbation Techniques. *ICDM Conference*, pp. 99-106, 2003.

[67] Karn J., Ullman J.: A model of statistical databases and their security. *ACM Transactions on Database Systems*, 2(1):1–10, 1977.

[68] Kenthapadi K.,Mishra N., Nissim K.: Simulatable Auditing, *ACM PODS Conference*, 2005.

[69] Kifer D., Gehrke J.: Injecting utility into anonymized datasets. *SIGMOD Conference*, pp. 217-228, 2006.

[70] Kim J., Winkler W.: Multiplicative Noise for Masking Continuous Data, *Technical Report Statistics 2003-01, Statistical Research Division, US Bureau of the Census*, Washington D.C., Apr. 2003.

[71] Kleinberg J., Papadimitriou C., Raghavan P.: Auditing Boolean Attributes. *Journal of Computer and System Sciences*, 6, 2003.

[72] Koudas N., Srivastava D., Yu T., Zhang Q.: Aggregate Query Answering on Anonymized Tables. *ICDE Conference*, 2007.

[73] Lakshmanan L., Ng R., Ramesh G. To Do or Not To Do: The Dilemma of Disclosing Anonymized Data. *ACM SIGMOD Conference*, 2005.

[74] Liew C. K., Choi U. J., Liew C. J. A data distortion by probability distribution. *ACM TODS*, 10(3):395-411, 1985.

[75] LeFevre K., DeWitt D., Ramakrishnan R.: Incognito: Full Domain K-Anonymity. *ACM SIGMOD Conference*, 2005.

[76] LeFevre K., DeWitt D., Ramakrishnan R.: Mondrian Multidimensional K-Anonymity. *ICDE Conference*, 25, 2006.

[77] LeFevre K., DeWitt D., Ramakrishnan R.: Workload Aware Anonymization. *KDD Conference*, 2006.

[78] Li F., Sun J., Papadimitriou S. Mihaila G., Stanoi I.: Hiding in the Crowd: Privacy Preservation on Evolving Streams through Correlation Tracking. *ICDE Conference*, 2007.

[79] Li N., Li T., Venkatasubramanian S: t-Closeness: Orivacy beyond k-anonymity and l-diversity. *ICDE Conference*, 2007.

[80] Lindell Y., Pinkas B.: Privacy-Preserving Data Mining. *CRYPTO*, 2000.

[81] Liu K., Kargupta H., Ryan J.: Random Projection Based Multiplicative Data Perturbation for Privacy Preserving Distributed Data Mining. *IEEE Transactions on Knowledge and Data Engineering*, 18(1), 2006.

[82] Liu K., Giannella C. Kargupta H.: An Attacker's View of Distance Preserving Maps for Privacy-Preserving Data Mining. *PKDD Conference*, 2006.

[83] Machanavajjhala A., Gehrke J., Kifer D., and Venkitasubramaniam M.: l-Diversity: Privacy Beyond k-Anonymity. *ICDE*, 2006.

[84] Malin B, Sweeney L. Re-identification of DNA through an automated linkage process. *Journal of the American Medical Informatics Association*, pp. 423–427, 2001.

[85] Malin B. Why methods for genomic data privacy fail and what we can do to fix it, *AAAS Annual Meeting*, Seattle, WA, 2004.

[86] Malin B., Sweeney L.: Determining the identifiability of DNA database entries. *Journal of the American Medical Informatics Association*, pp. 537–541, November 2000.

[87] Malin, B. Protecting DNA Sequence Anonymity with Generalization Lattices. *Methods of Information in Medicine*, 44(5): 687-692, 2005.

[88] Martin D., Kifer D., Machanavajjhala A., Gehrke J., Halpern J.: Worst-Case Background Knowledge. *ICDE Conference*, 2007.

[89] Meyerson A., Williams R. On the complexity of optimal k-anonymity. *ACM PODS Conference*, 2004.

[90] Mishra N., Sandler M.: Privacy vis Pseudorandom Sketches. *ACM PODS Conference*, 2006.

[91] Mukherjee S., Chen Z., Gangopadhyay S.: A privacy-preserving technique for Euclidean distance-based mining algorithms using Fourier based transforms, *VLDB Journal*, 2006.

[92] Moskowitz I., Chang L.: A decision theoretic system for information downgrading. *Joint Conference on Information Sciences*, 2000.

[93] Nabar S., Marthi B., Kenthapadi K., Mishra N., Motwani R.: Towards Robustness in Query Auditing. *VLDB Conference*, 2006.

[94] Naor M., Pinkas B.: Efficient Oblivious Transfer Protocols, *SODA Conference*, 2001.

[95] Natwichai J., Li X., Orlowska M.: A Reconstruction-based Algorithm for Classification Rules Hiding. *Australasian Database Conference*, 2006.

[96] Newton E., Sweeney L., Malin B.: Preserving Privacy by De-identifying Facial Images. *IEEE Transactions on Knowledge and Data Engineering, IEEE TKDE*, February 2005.

[97] Oliveira S. R. M., Zaane O.: Privacy Preserving Clustering by Data Transformation, *Proc. 18th Brazilian Symp. Databases*, pp. 304-318, Oct. 2003.

[98] Oliveira S. R. M., Zaiane O.: Data Perturbation by Rotation for Privacy-Preserving Clustering, *Technical Report TR04-17*, Department of Computing Science, University of Alberta, Edmonton, AB, Canada, August 2004.

[99] Oliveira S. R. M., Zaiane O., Saygin Y.: Secure Association-Rule Sharing. *PAKDD Conference*, 2004.

[100] Park H., Shim K. Approximate Algorithms for K-anonymity. *ACM SIGMOD Conference*, 2007.

[101] Pei J., Xu J., Wang Z., Wang W., Wang K.: Maintaining k-Anonymity against Incremental Updates. *Symposium on Scientific and Statistical Database Management*, 2007.

[102] Pinkas B.: Cryptographic Techniques for Privacy-Preserving Data Mining. *ACM SIGKDD Explorations*, 4(2), 2002.

[103] Polat H., Du W.: SVD-based collaborative filtering with privacy. *ACM SAC Symposium*, 2005.

[104] Polat H., Du W.: Privacy-Preserving Top-N Recommendations on Horizontally Partitioned Data. *Web Intelligence*, 2005.

[105] Rabin M. O.: How to exchange secrets by oblivious transfer, *Technical Report* TR-81, Aiken Corporation Laboratory, 1981.

[106] Reiss S.: Security in Databases: A combinatorial Study, *Journal of ACM*, 26(1), 1979.

[107] Rizvi S., Haritsa J.: Maintaining Data Privacy in Association Rule Mining. *VLDB Conference*, 2002.

[108] Saygin Y., Verykios V., Clifton C.: Using Unknowns to prevent discovery of Association Rules, *ACM SIGMOD Record*, 30(4), 2001.

[109] Saygin Y., Verykios V., Elmagarmid A.: Privacy-Preserving Association Rule Mining, *12th International Workshop on Research Issues in Data Engineering*, 2002.

[110] Samarati P.: Protecting Respondents' Identities in Microdata Release. IEEE Trans. Knowl. Data Eng. 13(6): 1010-1027 (2001).

[111] Shannon C. E.: The Mathematical Theory of Communication, University of Illinois Press, 1949.

[112] Silverman B. W.: Density Estimation for Statistics and Data Analysis. *Chapman and Hall*, 1986.

[113] Sweeney L.: Privacy Technologies for Homeland Security. *Testimony before the Privacy and Integrity Advisory Committee of the Deprtment of Homeland Scurity*, Boston, MA, June 15, 2005.

[114] Sweeney L.: Privacy-Preserving Bio-terrorism Surveillance. *AAAI Spring Symposium, AI Technologies for Homeland Security*, 2005.

[115] Sweeney L.: AI Technologies to Defeat Identity Theft Vulnerabilities. *AAAI Spring Symposium, AI Technologies for Homeland Security*, 2005.

[116] Sweeney L., Gross R.: Mining Images in Publicly-Available Cameras for Homeland Security. *AAAI Spring Symposium, AI Technologies for Homeland Security*, 2005.

[117] Sweeney L.: Guaranteeing Anonymity while Sharing Data, the Datafly System. *Journal of the American Medical Informatics Association*, 1997.

[118] Sweeney L.: Replacing Personally Identifiable Information in Medical Records, the Scrub System. *Journal of the American Medical Informatics Association*, 1996.

[119] Vaidya J., Clifton C.: Privacy-Preserving Association Rule Mining in Vertically Partitioned Databases. *ACM KDD Conference*, 2002.

[120] Vaidya J., Clifton C.: Privacy-Preserving k-means clustering over vertically partitioned Data. *ACM KDD Conference*, 2003.

[121] Vaidya J., Clifton C.: Privacy-Preserving Naive Bayes Classifier over vertically partitioned data. *SIAM Conference*, 2004.

[122] Vaidya J., Clifton C.: Privacy-Preserving Decision Trees over vertically partitioned data. *Lecture Notes in Computer Science*, Vol 3654, 2005.

[123] Verykios V. S., Bertino E., Fovino I. N., Provenza L. P., Saygin Y., Theodoridis Y.: State-of-the-art in privacy preserving data mining. *ACM SIGMOD Record*, v.33 n.1, 2004.

[124] Verykios V. S., Elmagarmid A., Bertino E., Saygin Y.,, Dasseni E.: Association Rule Hiding. *IEEE Transactions on Knowledge and Data Engineering*, 16(4), 2004.

[125] Wang K., Yu P., Chakraborty S.: Bottom-Up Generalization: A Data Mining Solution to Privacy Protection. *ICDM Conference*, 2004.

[126] Wang K., Fung B. C. M., Yu P. Template based Privacy -Preservation in classification problems. *ICDM Conference*, 2005.

[127] Wang K., Fung B. C. M.: Anonymization for Sequential Releases. *ACM KDD Conference*, 2006.

[128] Wang K., Fung B. C. M., Dong G.: Integarting Private Databases for Data Analysis. *Lecture Notes in Computer Science*, 3495, 2005.

[129] Warner S. L. Randomized Response: A survey technique for eliminating evasive answer bias. *Journal of American Statistical Association*, 60(309):63–69, March 1965.

[130] Winkler W.: Using simulated annealing for k-anonymity. *Technical Report 7, US Census Bureau*.

[131] Wu Y.-H., Chiang C.-M., Chen A. L. P.: Hiding Sensitive Association Rules with Limited Side Effects. *IEEE Transactions on Knowledge and Data Engineering*, 19(1), 2007.

[132] Xiao X., Tao Y.. Personalized Privacy Preservation. *ACM SIGMOD Conference*, 2006.

[133] Xiao X., Tao Y. Anatomy: Simple and Effective Privacy Preservation. *VLDB Conference*, pp. 139-150, 2006.

[134] Xiao X., Tao Y.: m-Invariance: Towards Privacy-preserving Republication of Dynamic Data Sets. *SIGMOD Conference*, 2007.

[135] Xu J., Wang W., Pei J., Wang X., Shi B., Fu A. W. C.: Utility Based Anonymization using Local Recoding. *ACM KDD Conference*, 2006.

[136] Xu S., Yung M.: k-anonymous secret handshakes with reusable credentials. *ACM Conference on Computer and Communications Security*, 2004.

[137] Yao A. C.: How to Generate and Exchange Secrets. *FOCS Conferemce*, 1986.

[138] Yao G., Feng D.: A new k-anonymous message transmission protocol. *International Workshop on Information Security Applications*, 2004.

[139] Yang Z., Zhong S., Wright R.: Privacy-Preserving Classification of Customer Data without Loss of Accuracy. *SDM Conference*, 2006.

[140] Yao C., Wang S., Jajodia S.: Checking for k-Anonymity Violation by views. *ACM Conference on Computer and Communication Security*, 2004.

[141] Yu H., Jiang X., Vaidya J.: Privacy-Preserving SVM using nonlinear Kernels on Horizontally Partitioned Data. *SAC Conference*, 2006.

[142] Yu H., Vaidya J., Jiang X.: Privacy-Preserving SVM Classification on Vertically Partitioned Data. *PAKDD Conference*, 2006.

[143] Zhang P., Tong Y., Tang S., Yang D.: Privacy-Preserving Naive Bayes Classifier. *Lecture Notes in Computer Science*, Vol 3584, 2005.

[144] Zhong S., Yang Z., Wright R.: Privacy-enhancing k-anonymization of customer data, In Proceedings of the ACM SIGMOD-SIGACT-SIGART Principles of Database Systems, Baltimore, MD. 2005.

[145] Zhu Y., Liu L. Optimal Randomization for Privacy- Preserving Data Mining. *ACM KDD Conference*, 2004.