International Journal of
Emerging Research in Management &Technology
ISSN: 2278-9359 (Volume-2, Issue-6)

Research Article | **June 2013**

A Review of Privacy Preserving Data Publishing Technique

Amar Paul Singh
School of CSE
Bahra University Shimla Hills, India

Ms. Dhanshri Parihar
Asst. Prof (School of CSE)
Bahra University Shimla Hills, India

Abstract:-

Today, most enterprises are actively collecting and storing data in large databases. Many of them have recognized the potential value of these data as an information source for making business decisions. Privacy-preserving data publishing (PPDP) provides methods and tools for publishing useful information while preserving data privacy. In this paper, a brief yet systematic review of several anonymization techniques such as generalization and bucketization, have been designed for privacy preserving micro data publishing. Recent work has shown that generalization loses considerable amount of information, especially for high-dimensional data. On the other hand, bucketization does not prevent membership disclosure. Whereas slicing preserves better data utility than generalization and also prevents membership disclosure. This paper focus on effective method that can be used for providing better data utility and can handle high-dimensional data.

Keywords: - Data Anonymization, Privacy Preservation, Data publishing, Data Security, PPDP.

I. INTRODUCTION

Data Mining which is sometimes also called as Knowledge Discovery Data (KDD) is the process of analysing data from different perspectives and summarizing it into useful information. Today, data mining is used by many companies with a strong consumer focus such as retail, financial, communication, and marketing organizations. Extraction of hidden predictive information from large databases is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Various algorithms and techniques like Classification, Clustering, Regression, Artificial Intelligence, Neural Networks, Association Rules, Decision Trees, Genetic Algorithm etc., are used for knowledge discovery from databases. In recent years, data mining has been used widely in the areas of science and engineering, such as bioinformatics, genetics, medicine, education and electrical power engineering. It has been said that knowledge is power, and this is exactly what data mining is about. It is the acquisition of relevant knowledge that can allow making strategic decisions.

- **Data Collection and Data Publishing**

A typical scenario of data collection and publishing is described in Figure 1.1. In the data collection phase, the data holder collects data from record owners (e.g., Alice and Bob). In the data publishing phase, the data holder releases the collected data to a data miner or the public, called the data recipient, who will then conduct data mining on the published data.

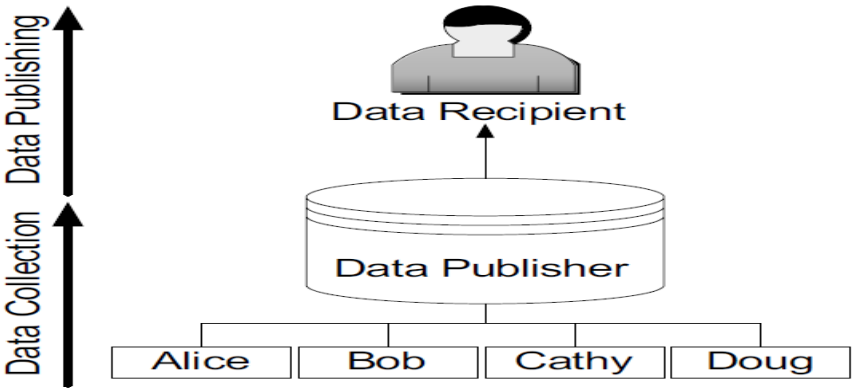

FIGURE 1.1: Data collection and Data Publishing [21]

- **Privacy-Preserving Data Publishing:**

In the most basic form of privacy-preserving data publishing (PPDP), the data holder has a table of the form:

D (Explicit Identifier, Quasi Identifier, Sensitive Attributes, non-Sensitive Attributes), where Explicit Identifier is a set of attributes, such as name and social security number SSN), containing information that explicitly identifies

record owners, Quasi Identifier is a set of attributes that could potentially identify record owners, Sensitive Attributes consist of sensitive person-specific information such as disease, salary, and disability status and Non-Sensitive Attributes contains all attributes that do not fall into the previous three categories. Most works assume that the four sets of attributes are disjoint. Most works assume that each record in the table represents a distinct record owner.

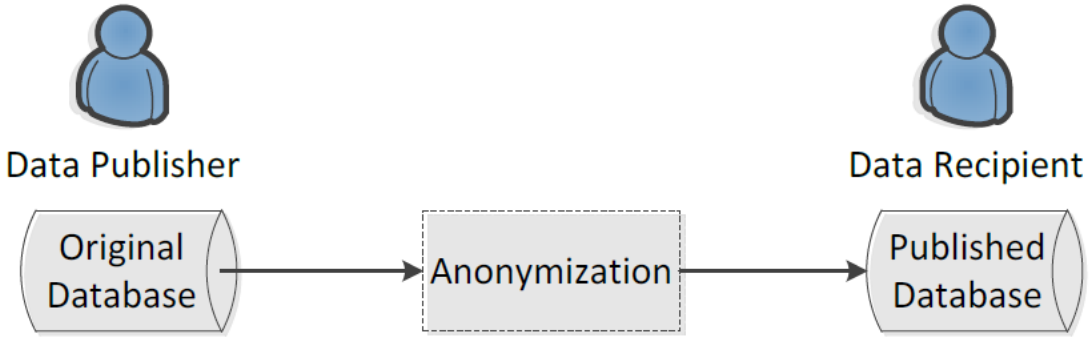

Figure 1.2: A Simple Model of PPDP [13].

- **Data Anonymization**

Data Anonymization is a technology that convert clear text into a non-human readable form.Data anonymization technique for privacy-preserving data publishing has received a lot of attention in recent years. Detailed data (also called as micro-data) contains information about a person, a household or an organization. Most popular anonymization techniques are Generalization and Bucketization. [1]There are number of attributes in each record which can be categorized as 1) Identifiers such as Name or Social Security Number are the attributes that can be uniquely identify the individuals. 2) some attributes may be Sensitive Attributes(SAs) such as disease and salary and 3) some may be Quasi-Identifiers (QI) such as zipcode, age, and sex whose values, when taken together, can potentially identify an individual.

Data is considered anonymized even when conjoined with pointer or pedigree values that direct the user to the originating system, record, and value (e.g., supporting selective revelation) and when anonymized records can be associated, matched, and/or conjoined with other anonymized records. Data anonymization enables the transfer of information across a boundary, such as between two departments within an agency or between two agencies, while reducing the risk of unintended disclosure, and in certain environments in a manner that enables evaluation and analytics post-anonymization.

[1] The two techniques differ in the next step. Generalization transforms the QI-values in each bucket into "less specific but semantically consistent" values so that tuples in the same bucket cannot be distinguished by their QI values. In bucketization, one separates the SAs from the QIs by randomly permuting the SA values in each bucket. The anonymized data consist of a set of buckets with permuted sensitive attribute values.

II. VARIOUS ANONYMIZATION TECHNIQUES

Two widely studied data anonymization technique are generalization and bucketization. The main difference between the two anonymization techniques lies in that bucketization does not generalize the QI attributes.

A. Generalization

Generalization is one of the commonly anonymized approaches, which replaces quasi-identifier values with values that are less-specific but semantically consistent. Then, all quasi-identifier values in a group would be generalized to the entire group extent in the QID space. [12] If at least two transactions in a group have distinct values in a certain column (i.e. one contains an item and the other does not), then all information about that item in the current group is lost. The QID used in this process includes all possible items in the log. Due to the high-dimensionality of the quasi-identifier, with the number of possible items in the order of thousands, it is likely that any generalization method would incur extremely high information loss, rendering the data useless [8]. In order for generalization to be effective, records in the same bucket must be close to each other so that generalizing the records would not lose too much information. However, in high-dimensional data, most data points have similar distances with each other. To perform data analysis or data mining tasks on the generalized table, the data analyst has to make the uniform distribution assumption that every value in a generalized interval/set is equally possible, as no other distribution assumption can be justified. This significantly reduces the data utility of the generalized data. And also because each attribute is generalized separately, correlations between different attributes are lost. In order to study attribute correlations on the generalized table, the data analyst has to assume that every possible combination of attribute values is equally possible. This is an inherent problem of generalization that prevents effective analysis of attribute correlations.

B. Bucketization

The first, which we term bucketization, is to partition the tuples in T into buckets, and then to separate the sensitive attribute from the non-sensitive ones by randomly permuting the sensitive attribute values within each bucket. The sanitized data then consists of the buckets with permuted sensitive values. In this paper [13] we use bucketization as the method of constructing the published data from the original table T, although all our results hold for full-domain generalization as well. We now specify our notion of bucketization more formally. Partitition the tuples into buckets (i.e., horizontally partition the table T according to some scheme), and within each bucket, we apply an independent random permutation to the column containing S-values. The resulting set of buckets, denoted by B, is then published. For example, if the underlying table T, then the publisher might publish bucketization B .Of course, for added privacy, the publisher can completely mask the identifying attribute (Name) and may partially mask some of the other non-sensitive attributes (Age, Sex, Zip). For a bucket b ϵ B, we use the following notation.

While bucketization [1, 13] has better data utility than generalization, it has several limitations. First, bucketization does not prevent membership disclosure. Because bucketization publishes the QI values in their original forms, an adversary can find out whether an individual has a record in the published data or not. As shown in, 87 percent of the individuals in the United States can be uniquely identified using only three attributes (Birth date, Sex, and Zipcode). A micro data (e.g., census data) usually contains many other attributes besides those three attributes. This means that the membership information of most individuals can be inferred from the bucketized table.

Second, bucketization requires a clear separation between QIs and SAs. However, in many data sets, it is unclear which attributes are QIs and which are SAs. Third, by separating the sensitive attribute from the QI attributes, bucketization breaks the attribute correlations between the QIs and the SAs.

Bucketization first partitions tuples in the table into buckets and then separates the quasi identifiers with the sensitive attribute by randomly permuting the sensitive attribute values in each bucket. The anonymized data consist of a set of buckets with permuted sensitive attribute values. In particular, bucketization has been used for anonymzing high-dimensional data. However, their approach assumes a clear separation between QIs and SAs. In addition, because the exact values of all QIs are released, membership information is disclosed.

C. Slicing

To improve the current state of the art in this paper, we introduce a novel data anonymization technique called slicing [1]. Slicing partitions the data set both vertically and horizontally. Vertical partitioning is done by grouping attributes into columns based on the correlations among the attributes. Each column contains a subset of attributes that are highly correlated. Horizontal partitioning is done by grouping tuples into buckets. Finally, within each bucket, values in each column are randomly permutated (or sorted) to break the linking between different columns. The basic idea of slicing is to break the association cross columns, but to preserve the association within each column. This reduces the dimensionality of the data and preserves better utility than generalization and bucketization.

Slicing preserves utility because it groups highly correlated attributes together, and preserves the correlations between such attributes. Slicing protects privacy because it breaks the associations between uncorrelated attributes, which are infrequent and thus identifying. Note that when the data set contains QIs and one SA, bucketization has to break their correlation; slicing, on the other hand, can group some QI attributes with the SA, preserving attribute correlations with the sensitive attribute.

The key intuition that slicing provides privacy protection is that the slicing process ensures that for any tuple, there are generally multiple matching buckets. Slicing first partitions attributes into columns. Each column contains a subset of attributes. Slicing also partition tuples into buckets. Each bucket contains a subset of tuples. This horizontally partitions the table. Within each bucket, values in each column are randomly permutated to break the linking between different columns.

III. BACKGROUND

Two main Privacy preserving paradigms have been established: k-anonymity [7], which prevents identification of individual records in the data, and l-diversity [1], which prevents the association of an individual record with a sensitive attribute value.

K-anonymity

The database is said to be K-anonymous where attributes are suppressed or generalized until each row is identical with at least k-1 other rows. K-Anonymity thus prevents definite database linkages. K-Anonymity guarantees that the data released is accurate. K-anonymity proposal focuses on two techniques in particular: generalization and suppression. [2] To protect respondents' identity when releasing micro data, data holders often remove or encrypt explicit identifiers, such as names and social security numbers. De-identifying data, however, provide no guarantee of anonymity. Released information often contains other data, such as birth date, sex, and ZIP code that can be linked to publicly available information to re-identify respondents and to infer information that was not intended for release. One of the emerging concepts in micro data protection is k-anonymity, which has been recently proposed as a property that captures the protection of a microdata table with respect to possible re-identification of the respondents to which the data refer. K-anonymity demands that every tuple in the microdata table released be indistinguishably related to no fewer than k

respondents. One of the interesting aspects of k-anonymity is its association with protection techniques that preserve the truthfulness of the data. The first approach toward privacy protection in data mining was to perturb the input (the data) before it is mined. The drawback of the perturbation approach is that it lacks a formal framework for proving how much privacy is guaranteed. At the same time, a second branch of privacy preserving data mining was developed, using cryptographic techniques. Thus, it falls short of providing a complete answer to the problem of privacy preserving data mining. One definition of privacy which has come a long way in the public arena and is accepted today by both legislators and corporations is that of k-anonymity [3]. The guarantee given by k-anonymity is that no information can be linked to groups of less than k individuals. Generalization for k-anonymity losses considerable amount of information, especially for high-dimensional data.

[4] Limitations of k-anonymity are: (1) it does not hide whether a given individual is in the database, (2) it reveals individuals' sensitive attributes , (3) it does not protect against attacks based on background knowledge , (4) mere knowledge of the k-anonymization algorithm can violate privacy, (5) it cannot be applied to high-dimensional data without complete loss of utility , and (6) special methods are required if a dataset is anonymized and published more than once.

l- Diversity

The next concept is "l-diversity". Say you have a group of k different records that all share a particular quasi-identifier. That's good, in that an attacker cannot identify the individual based on the quasi-identifier. But what if the value they're interested in, (e.g. the individual's medical diagnosis) is the same for every value in the group. The distribution of target values within a group is referred to as "l-diversity". [8] Currently, there exist two broad categories of l-diversity techniques: generalization and permutation-based. An existing generalization method would partition the data into disjoint groups of transactions, such that each group contains sufficient records with l-distinct, well represented sensitive items.

IV. SLICING ALGORITHM

Slicing algorithm here we compare it with generalization and bucketization, and discuss privacy threats that slicing can address [8]. Generally in privacy preservation there is a loss of security. The privacy protection is impossible due to the presence of the adversary's background knowledge in real life application. Data in its original form contains sensitive information about individuals. These data when published violate the privacy. The current practice in data publishing relies mainly on policies and guidelines as to what types of data can be published and on agreements on the use of published data. The approach alone may lead to excessive data distortion or insufficient protection. Privacy-preserving data publishing (PPDP) provides methods [8] and tools for publishing useful information while preserving data privacy. Many algorithms like bucketization, generalization have tried to preserve privacy however they exhibit attribute disclosure. So to overcome this problem an algorithm called slicing is used. This algorithm consists of three phases: attribute partitioning, column generalization, and tuple partitioning.

- **Attribute Partitioning**

This algorithm partitions attributes so that highly correlated attributes are in the same column. This is good for both utility and privacy. In terms of data utility, grouping highly correlated attributes preserves the correlations among those attributes. In terms of privacy, the association of uncorrelated attributes presents higher identification risks than the association of highly correlated attributes because the associations of uncorrelated attribute values is much less frequent and thus more identifiable.

- **Column Generalization**

First, column generalization may be required for identity/membership disclosure protection. If a column value is unique in a column, a tuple with this unique column value can only have one matching bucket. This is not good for privacy protection, as in the case of generalization/bucketization where each tuple can belong to only one equivalence-class/bucket.

- **Tuple Partitioning**

The algorithm maintains two data structures: 1) a queue of buckets Q and 2) a set of sliced buckets SB. Initially, Q contains only one bucket which includes all tuples and SB is empty. For each iteration, the algorithm removes a bucket from Q and splits the bucket into two buckets [5]. If the sliced table after the split satisfies l-diversity, then the algorithm puts the two buckets at the end of the queue Q Otherwise, we cannot split the bucket anymore and the algorithm puts the bucket into SB.When Q becomes empty, we have computed the sliced table. The set of sliced buckets is SB.

Slicing Architecture

Generally in privacy preservation there is a loss of security. The privacy protection is impossible due to the presence of the adversary's background knowledge in real life application. Data in its original form contains sensitive information about individuals. These data when published violate the privacy. The current practice in data publishing relies mainly on policies and guidelines as to what types of data can be published and on agreements on the use of published data. The approach alone may lead to excessive data distortion or insufficient protection. Privacy-preserving data publishing

(PPDP) provides methods and tools for publishing useful information while preserving data privacy. Many algorithms like bucketization, generalization have tried to preserve privacy however they exhibit attribute disclosure. So to overcome this problem an algorithm called slicing is used.

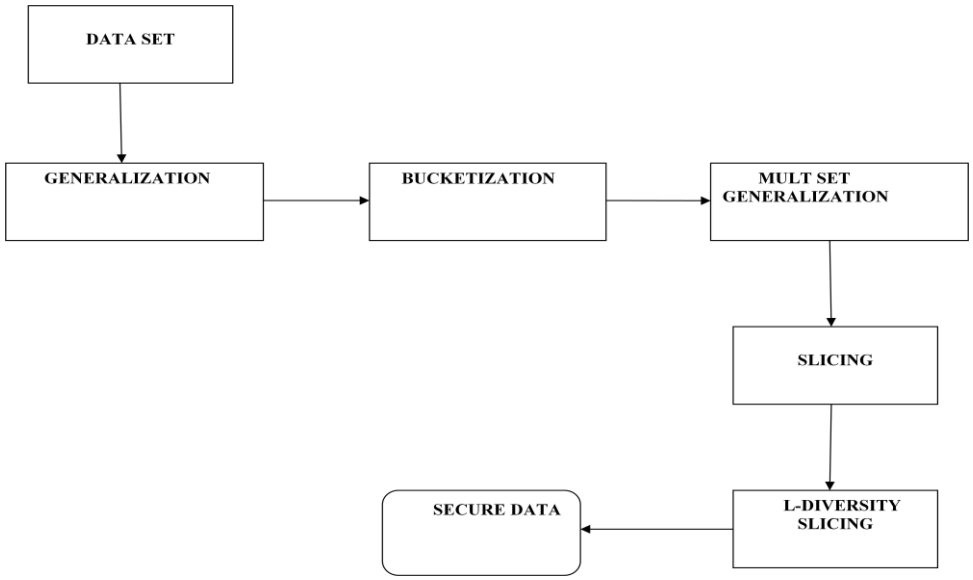

Figure 2.1. Slicing Architecture.

Functional procedure:-

Step 1: Extract the data set from the database.
Step 2: Anonymity process divides the records into two.
Step 3: Interchange the sensitive values.
Step 4: Multiset values generated and displayed.
Step 5: Attributes are combined and secure data Displayed.

V. SLICING WITH TUPLE GROUPING ALGORITHM

Slicing with Tuple grouping algorithm provides efficient random tuple grouping for micro data publishing. Each column contains sliced bucket (SB) that permutated random values for each partitioned data. It is also permutated the frequency of the value in each one of the scan's-diversity algorithm checks the diversity when the each sliced table.

A. Architecture of slicing with tuple grouping

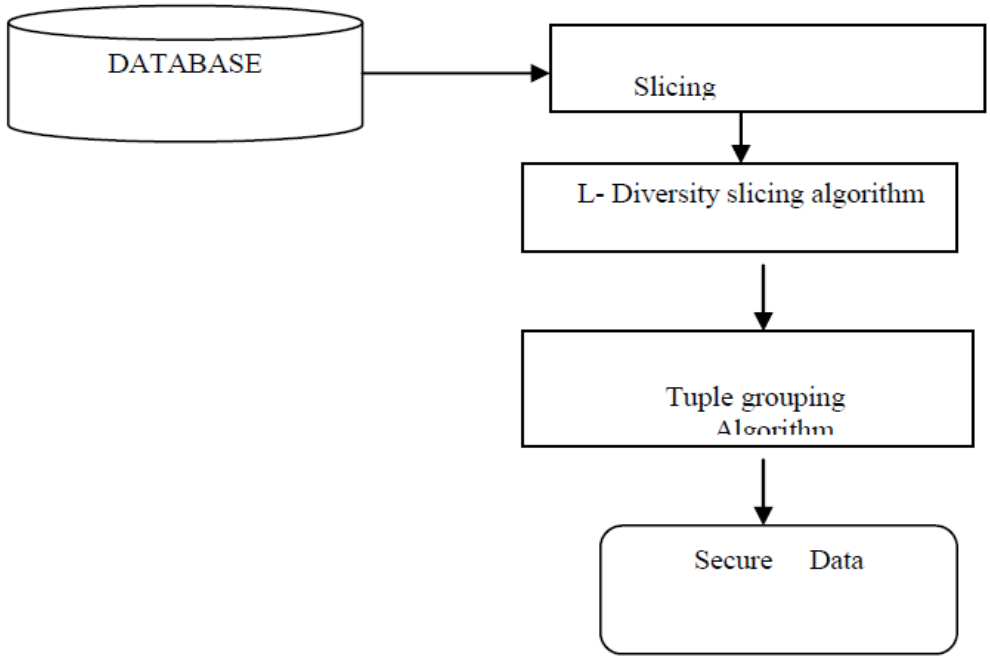

Figure 2.2 Slicing Architecture with tuple grouping.

B. Functional procedure

Step 1: Extract the data set from the database.
Step 2: Removes the queue of buckets and splits the Bucket into two
Step 3: computes the sliced table.
Step 4: Diversity maintains the multiple matching Buckets.
Step 3: Random tuples are computed.
Step 5: Attributes are combined and secure data Displayed.

The main part of the tuple-partition algorithm is to check whether a sliced table satisfies „l-diversity gives a description of the diversity-check algorithm. For each tuple t, the algorithm maintains a list of statistics L (t) about t's matching buckets. each element in the list L(t) contains statistics about one matching bucket b,the matching probability p(t,B) and the distribution of candidate sensitive values d(t,B). The algorithm first takes one scan of each bucket b to record the frequency f(v) of each column value v in bucket b. Then, the algorithm takes one scan of each tuple t in the table t to find out all tuples that match b and record their matching probability p(t,B) and the distribution of candidate sensitive values d(t,B) which are added to the list l(t). We have obtained, for each tuple t, the list of statistics L (t) about its matching buckets. A final scan of the tuples in t will compute the p (t, b) values based on the law of total probability.

VI. CONCLUSION

Slicing overcomes the limitations of generalization and bucketization and preserves better utility while protecting against privacy threats we consider slicing where each attribute is in exactly one column. An extension is the notion of overlapping slicing, which duplicates an attribute in more than one column. Our experiments show that random grouping is not very effective. The Proposed grouping algorithm is optimized L-diversity slicing check algorithm obtains the more effective tuple grouping and Provides secure data. Another direction is to design data mining tasks using the anonymized data computed by various anonymization techniques. Another important advantage of slicing is that it can handle high-dimensional data.

REFERENCES

[1] Tiancheng Li, Ninghui Li, Senior Member, IEEE, Jia Zhang, Member, IEEE, and Ian Molloy "Slicing: A New Approach for Privacy Preserving Data Publishing" Proc. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 24, NO. 3, MARCH 2012.
[2] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati On K-Anonymity. In Springer US, Advances in Information Security (2007).
[3] Latanya Sweeney. k-anonymity: a model for protecting privacy. International Journal on Uncertainty, Fuzziness and Knowledge-Based Systems, 10(5):557–570, 2002.
[4] J. Brickell and V. Shmatikov, "The Cost of Privacy: Destruction of Data-Mining Utility in Anonymized Data Publishing," Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 70-78, 2008

[5] N. Li, T. Li, and S. Venkatasubramanian, "t-Closeness: Privacy Beyond k-Anonymity and „-Diversity," Proc. IEEE 23rd Int'l Conf. Data Eng. (ICDE), pp. 106-115, 2007.
[6] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. "l-diversity: Privacy beyond k-anonymity". In ICDE, 2006.
[7] D. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J. Halpern. "Worst-case background knowledge for privacy-preserving data publishing". In ICDE, 2007.
[8] G.Ghinita, Y. Tao, and P. Kalnis, "On the Anonymization of Sparse High-Dimensional Data," Proc. IEEE 24th Int'l Conf. Data Eng. (ICDE), pp. 715-724, 2008.
[9] R. J. Bayardo and R. Agrawal, "Data Privacy through Optimal k- Anonymization," in Proc. of ICDE, 2005, pp. 217–228.
[10] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Incognito: Efficient Full-domain k-Anony Anonymity," in Proc. of ACM SIGMOD, 2005, pp. 49– 60.
[11] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Mondrian Multidimensional k-Anonymity," in Proc. of ICDE, 2006.
[12] Gabriel Ghinita, Member IEEE, Panos Kalnis, Yufei Tao," Anonymous Publication of Sensitive Transactional Data" in Proc. Of IEEE Transactions on Knowledge and Data Engineering February 2011 (vol. 23 no. 2) pp. 161-174.
[13] D.J. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J.Y. Halpern, "Worst-Case Background Knowledge for Privacy- Preserving Data Publishing," Proc. IEEE 23rd Int'l Conf. Data Eng. (ICDE), pp. 126-135, 2007.
[14] X. Xiao and Y. Tao, "Anatomy: Simple and Effective Privacy Preservation," Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 139-150, 2006.
[15] Y. He and J. Naughton, "Anonymization of Set-Valued Data via Top-Down, Local Generalization," Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 934-945, 2009.
[16] D. Kifer and J. Gehrke, "Injecting Utility into Anonymized Data Sets," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD), pp. 217-228, 2006.
[17] T. Li and N. Li, "On the Tradeoff between Privacy and Utility in Data Publishing," Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 517-526, 2009.
[18] Y. Xu, K. Wang, A.W.-C. Fu, and P.S. Yu, "Anonymizing Transaction Databases for Publication," Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 767-775, 2008.
[19] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A.W.-C. Fu, "Utility- Based Anonymization Using Local Recoding," Proc. 12th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 785-790, 2006.
[20] C. Dwork, "Differential Privacy: A Survey of Results," Proc. Fifth Int'l Conf. Theory and Applications of Models of Computation (TAMC), pp. 1-19, 2008.
[21] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu. Privacy-preserving data publishing: A survey on recent developments. ACM Computing Surveys, 42(4), December 2010.