Slicing: Privacy Preserving Data Publishing Technique

Ashwini Andhalkar[#1], Pradnya Ingawale[#2]
[#]*Computer Dept, PVPIT, University of Pune, India*
[#]*Computer Dept, PVPIT, University of Pune, 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 focuses 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 Anonymization*

Data Anonymization is a technology that converts 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 zip code, 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.

- *Various Anonymization Techniques*

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. [2] 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 [3]. 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 [4] 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. Partition 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, 4] 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 Zip code). 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 partitions the 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.

II. RELATED WORK

Two popular Anonymization techniques are generalization and bucketization. Generalization [5], replaces a value with a “less-specific but semantically consistent” value. The main problems with generalization are: 1) it fails on high-dimensional data due to the curse of dimensionality and 2) it causes too much information loss due to the uniform-distribution assumption.

Bucketization [4] 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 anonymizing 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.

The key idea of slicing is to preserve correlations between highly correlated attributes and to break correlations between uncorrelated attributes thus achieving both better utility and better privacy. Third, existing data analysis (e.g., query answering) methods can be easily used on the sliced data.

III. SYSTEM 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.

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: Multi set values generated and displayed.

Step 5: Attributes are combined and secure data Displayed.

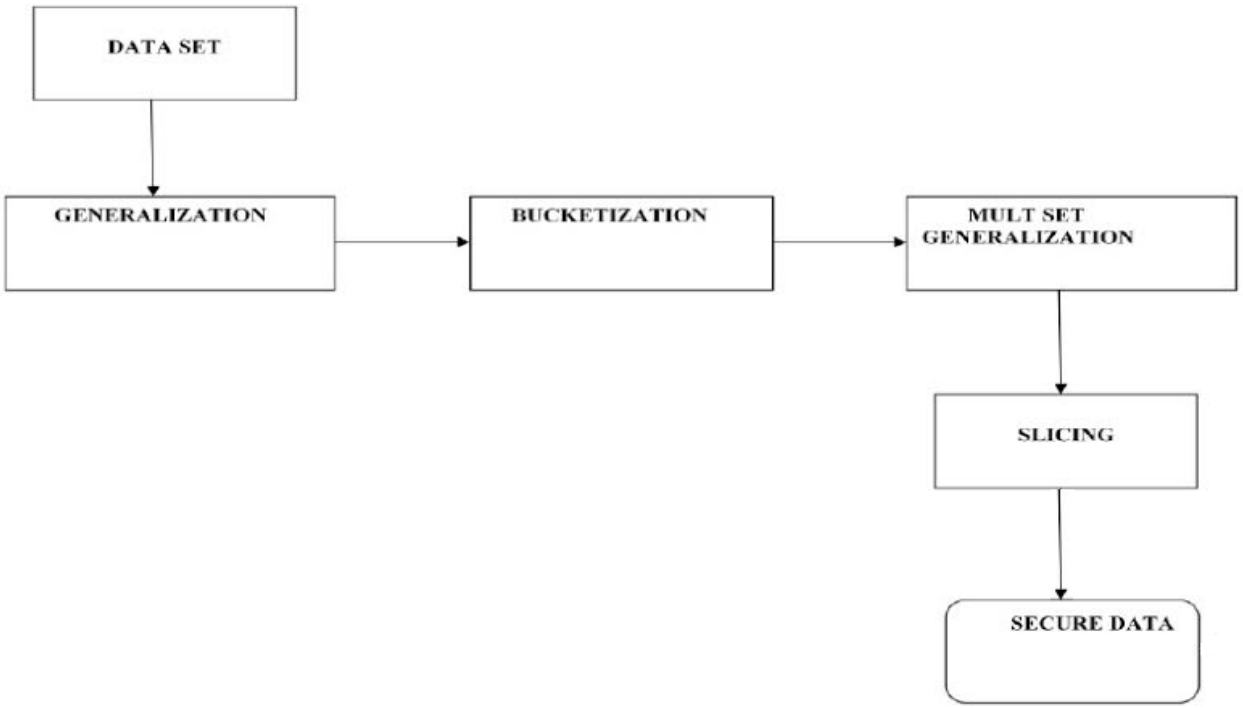

Fig 1 Slicing Architecture

IV. SLICING ALGORITHM

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

Although column generalization is not a required phase, it can be useful in several aspects. First, column generalization may be required for identity/membership disclosure protection. If a column value is unique in a column (i.e., the column value appears only once in the 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. The main problem is that this unique column value can be identifying. In this case, it would be useful to apply column generalization to ensure that each column value appears with at least some frequency. Second, when column generalization is applied, to achieve the same level of privacy against attribute disclosure, bucket sizes can be smaller. While column generalization may result in information loss, smaller bucket-sizes allow better data utility. Therefore, there is a trade-off between column generalization and tuple partitioning.

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. 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.

V. FUTURE SCOPE AND CONCLUSION

Slicing overcomes the limitations of generalization and bucketization and preserves better utility while protecting against privacy threats. Slicing prevents attribute disclosure and membership disclosure. Slicing preserves better data utility than generalization and is more effective than bucketization in workloads involving the sensitive attribute.

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. We plan to design more effective tuple grouping algorithms. Another direction is to design data mining tasks using the anonymized data [6] computed by various Anonymization techniques.

Slicing protect privacy by breaking the association of uncorrelated attributes and preserve data utility by preserving the association between highly correlated attributes. Another important advantage of slicing is that it can handle high-dimensional data.

ACKNOWLEDGMENT

We are greatly indebted to our college Padmabhooshan Vasantdada Patil Institute of Technology that has provided a healthy environment to drive us to do this project and thankful to our management for their guidance.

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] 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.

[3] 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.

[4] 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.

[5] P. Samarati, *"Protecting Respondent's Privacy in Micro data Release,"* IEEE Trans. Knowledge and Data Eng., vol. 13, no. 6, pp. 1010- 1027,Nov/Dec. 2001.

[6] A. Inan, M. Kantarcioglu, and E. Bertino, *"Using Anonymized Data for Classification,"* Proc. IEEE 25th Int'l Conf. Data Eng. (ICDE), pp. 429-440, 2009.