Volume 3, Issue 3, March 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Tuple Grouping Strategy for Privacy Preservation of Microdata Disclosure

R.Maheswari*
**Student,M.E(CSE)*
Srinivasan Engineering college
Perambalur,India

V.Gayathri**
***Assistant professor(CSE)*
Srinivasan Engineering college
Perambalur,India

S.Jaya Prakash***
**** Assistant professor(CSE)*
Srinivasan Engineering college
Perambalur,India

Abstract- Privacy preservation is important one while publishing the information containing individual specific records like employee's records and patient's records. Generally Sensitive information about individual's records will violate the privacy. There are several techniques have been introduced for preserving the privacy. The existing system has been designed with the anonymization techniques of generalization and bucketization. Those methods were revealed the privacy of individual information to the adversaries. Generalization involved considerable loss of the information and bucketization method does not protection from membership disclosure and there is no clear separation between sensitive attributes and quasi identifier attributes. The proposed system of slicing with Tuple grouping algorithm partitioned the data both vertically and horizontally. It provides better data utility than generalization and protection from membership disclosure and also can handle in high dimensional data. Government agencies, and organization and companies are shared the data and publication of the data for research purpose.

Keywords- Privacy Preservation, Microdata, Sensitive information, Data anonymization, Data publishing, Data security.

I. INTRODUCTION

Data mining is the extracting the meaningful information from the large data sets such as data warehouse, Micro data contains records each of which contains information about an individual entity, such as a person, an organization or household. Several micro data anonymization techniques have been introduced. The most popular that generalization for k-anonymity [1] and bucketization for l-diversity [1][6]. In both approaches attributes are partitioned into three categories: 1) some attributes are identifiers that can uniquely identify an individual like Name or Social Security Number; 2) some attributes are Quasi Identifiers(QI), which the adversary may already know and which, when taken together, can potentially identify an individual, e.g., Birth date, Sex, and Zip code; 3) some attributes are Sensitive Attributes (SAs), which are unknown to the adversary and are considered sensitive, like Disease and Salary. Generally when the micro data publishing the various attacks occurred like record linkage model [1][7]attack and attribute linkage model attack. So avoid these attacks the various anonymization techniques was introduced. In both generalization and bucketization removes the identifiers from the data and also partitions tuples into buckets. Buckets contain the subset of tuples. Generalization transforms the QI values in each bucket into "less specific but semantically consistent" values. So that tuples of the same bucket cannot be distinguished by their QI values. In bucketization separates the SAs from the QIs but randomly permuting the SA values in each bucket.

The major limitation of the traditional approach of k Anonymity is that link the external data with shared data for research purpose. In generalization all the attributes are suppressed until each row is identical. It is used for prevent identifier disclosure but it is not guarantee to the entire privacy and lose the information in high dimensional data. In bucketization technique all the sensitive information denoted "The values are well represented". This technique has several limitations first one is does not prevent membership disclosure. Because bucketization publishes the quasi identifier (QI) values in their original forms, an adversary can find out whether an individual has a record in the already published data or not. The proposed Slicing algorithm with Tuple grouping algorithm is partitioned the data both vertically and horizontally. The random values are permutated within each bucket and also can handle in high dimensional data .It is more data utility than generalization and bucketization.

II. Related Work

Although there has been a notion of Anonymization in the existing algorithms, and various technique would possibly suit the real time databases. Here we discuss about the existing methods/algorithms for Privacy preserving of micro data publishing.

A. Generalization

There are several types of recodings for generalization. The recoding that preserves the most information is local recoding [6]. In local recoding, one first groups tuples into buckets and then for each bucket, one replaces all values of one attribute with a generalized value. Such a recoding is local because the same attribute value may be generalized

differently when they appear in different buckets. We now show that slicing preserves more information than such a local recoding approach, assuming that the same tuple partition is used.

We achieve this by showing that slicing is better than the following enhancement of the local recoding approach. Rather than using a generalized value to replace more specific attribute values, one uses the multiset [4]of exact values in each bucket. The multiset of exact values provides more information about the distribution of values in each attribute than the generalized interval. Therefore, using multisets of exact values preserves more information than generalization.

B. Bucketization

We first note that bucketization can be viewed as a special case of slicing, where there are exactly two columns: one column contains only the SA, and the other contains all the QIs. The advantages of slicing over bucketization [9] can be understood as follows: First, by partitioning attributes into more than two columns, slicing [8] can be used to prevent membership leak. Our empirical evaluation on a real data set shows that bucketization does not prevent membership disclosure. Second, unlike bucketization, which requires a clear separation of QI attributes and the sensitive attribute, slicing can be used without such a separation. For data set such as the census data, one often cannot clearly separate QIs from SAs because there is no single external public database that one can use to determine which attributes the adversary already knows. Slicing can be useful for such data.

Finally, by allowing a column to contain both some QI attributes and the sensitive attribute, attribute correlations between the sensitive attribute and the QI attributes are preserved. For example Zip code and Disease form one column, enabling inferences about their correlations. Attribute correlations are important utility in data publishing. For workloads that consider attributes in isolation, one can simply publish two tables, one containing all QI attributes and one containing the sensitive attribute. Modeling adversary's background knowledge is most privacy models, such as k-anonymity, l-diversity, confidence bounding, and t-closeness, assume the adversary has only very limited background knowledge [4]. Specifically, they assume that the adversary's background knowledge is limited to knowing the quasi-identifier. Yet, recent work has shown the importance of integrating an adversary's background knowledge in privacy quantification. A robust privacy notion has to take background knowledge into consideration. Since an adversary can easily learn background knowledge from various sources.

C. Privacy Threats

When publishing micro data, there are three types of privacy disclosure threats. The first type is membership disclosure [8][1]. When the data set to be published is selected from a large population and the selection criteria are sensitive (e.g., only diabetes patients are selected), one needs to prevent adversaries from learning whether one's record is included in the published data set. The second type is identity disclosure, which occurs when an individual is linked to a particular record in the released table. In some of the situations, one wants to protect against identity disclosure when the adversary is uncertain of membership. In this case, protection against membership disclosure helps protect against identity disclosure. In other situations, some adversary may already know that an individual's record is in the published data set, in which case, membership disclosure protection either does not apply or is insufficient.

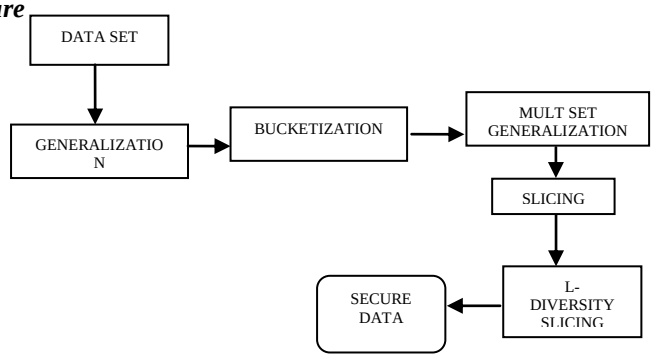
The third type is attribute disclosure, which occurs when new information about some individuals is revealed, i.e., the released data make it possible to infer the attributes of an individual more accurately than it would be possible before the release. Similar to the case of identity disclosure, we need to consider adversaries who already know the membership information. Identity leak leads to attribute disclosure. Once there is identity disclosure, an individual is reidentified [6] and the corresponding sensitive value is revealed. Attribute disclosure can occur with or without identity disclosure, e.g., when the sensitive values of all matching tuples are the same.

III. Problem Definition And Architecture

A. Problem Definition

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.

B. Functional and Slicing Architecture



C. 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.

D. Slicing algorithm

We then formalize slicing, 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. We now describe the three phases.

- ***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***

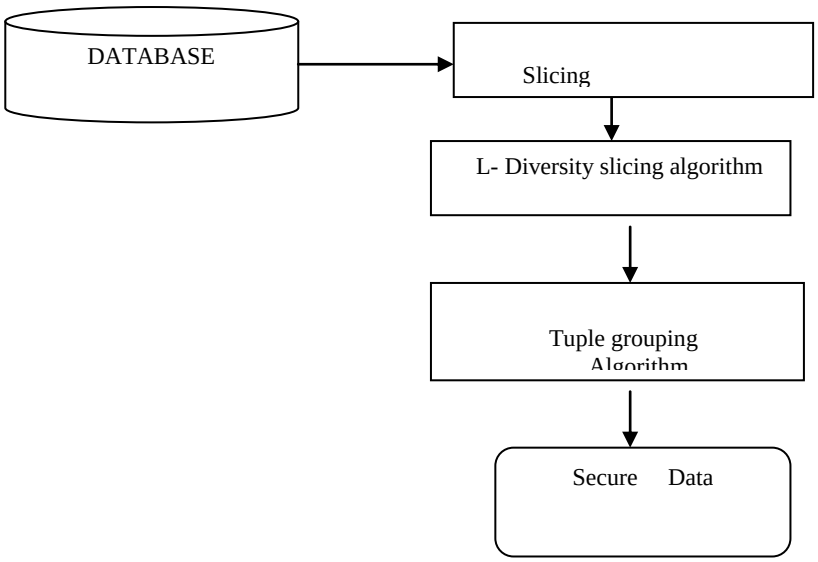
The algorithm maintains two data structures:
1) a queue of buckets Q and 2) a set of sliced buckets SB. Initially, Q contains only one bucket which includes all tuples and SB is empty. For each iteration, the algorithm removes a bucket from Q and splits the bucket into two buckets [5].

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.

IV. 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



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.

$$P(t,s) = \sum_{e \in L(t)} e.p(t,B) * e.D(t,B)[s] \qquad (1)$$

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.

V. Experimental Evaluation

To allow direct comparison, we use the l-diversity for two anonymization techniques: slicing and optimized slicing for tuple grouping. This experiment demonstrates that: 1) slicing preserves better data utility than generalization; 2) slicing is more effective than bucketization in workloads involving the sensitive attribute; and 3) the sliced table can be computed efficiently. Both bucketization and slicing perform much better than generalization.

We compare slicing with optimized slicing in terms of computational efficiency. We fix l= 5 and vary the cardinality of the data (i.e., the number of records) and the dimensionality of the data (i.e., the number of attributes). Fig 4.1 shows the computational time as a function of data.

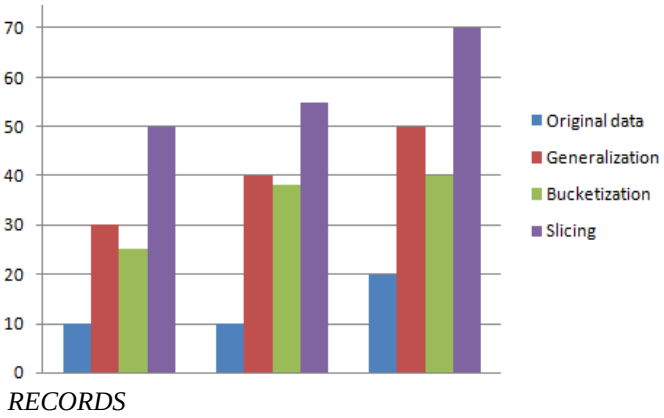


Fig 1: Computational efficiency.

VI. CONCLUSION

The implementation of previously existing systems provided clear view of the problem to be addressed. Slicing overcomes the limitations of generalization and bucketization and preserves better utility while protecting against privacy threats. Our experiments show that slicing preserves better data utility than generalization and is more effective than bucketization in workloads involving the sensitive attribute. First, in this paper, 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.

Acknowledgement

The authors thank the anonymous referees for their extremely useful comments and suggestions on an earlier draft of this paper.

References

[1] Aggarwal.C, "On K-Anonymity and the Curse of Dimensionality," Proc. Int'l Conf.Very Large Data Bases (VLDB), 2005.

[2] Brickell.J and 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), 2008.

[3] Ghinita.G,Tao.Y, and Kalnis.P, "OnThe Anonymization of Sparse High Dimensional Data," Proc. IEEE 24th Int'l Conf. Data Eng. (ICDE), 2008.

[4] He.Y and Naughton.J, "Anonymization of Set-Valued Data via Top-Down, local Generalization," Proc.IEEE 25th Int'l Conf.Data Engineering (ICDE), 2009.

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

[6] Li.T and Li.N, "On the Tradeoff between Privacy and Utility in Data Publishing," Proc.ACM SIGKDD Int'l Conf.Knowledge Discovery and Data Mining (KDD), 2009.

[7] Li.N, Li.T, "Slicing: The new Approach for Privacy Preserving Data publishing", IEEE Transaction on knowledge and data Engineering, vol.24, No, 3, March 2012.

[8] Li.N, Li.T, and Venkatasubramanian.S,"t-Closeness: Privacy Beyond K-Anonymity And L-Diversity,"Proc.IEEE 23rd Int'l Conf.Data Eng.(IDCE),2007.

[9] Machanavajjhala.A, Gehrke.J, Kifer.D, and M.Venkitasubramaniam, "L-diversity privacy Beyond K-Anonymity",Proc.IEEE 23 rd. Int'l Conf.Data Eng,(ICDE),2007.

Biographies

R.Maheswari received the B.E Degree computer science in odaiyappa college of engineering and technology, and now she is an M.E student in the Department Of Computer Science & Engineering, s Srinivasan Engineering College – Dhanalakshmi Srinivasan Group of Institutions, Perambalur,TN, India. His research interest includes Web Databases, Data mining.

V.Gayathri is working as Assistant Professor at Srinivasan Engineering College –Dhanalakshmi Srinivasan Group of Institutions, Perambalur, TN, India.His main research interest includes Data Mining and Networks.

S.Jayaprakash is working as **Assistant Professor**, Srinivasan Engineering College –Dhanalakshmi Srinivasan Group of Institutions, Perambalur,TN, India.

His research interest includes Multicasting Wireless in AdHoc Networks using ODMRP.

