International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064

Privacy Data Publishing Using Slicing and Tuple Grouping Strategy

E. Ashwini Kumari[1], N. Chandra Sekhar Reddy[2], G.V Geetha Madhavi[3]

[1]Student, M. Tech CSE Department, Institute of Aeronautical Engineering,

[2]Professor, CSE Department, Institute of Aeronautical Engineering,

[3]Associate Professor, CSE Department Department, Institute of Aeronautical Engineering,

Abstract: ***The important factor while publishing the information is better data utility, the information contains individual specific records like employees records, patients records etc. There are many techniques introduced for providing privacy, the existing system has designed with generalization and bucketization techniques along with the slicing technique. Consider the loss of information is the problem of those methods and they doesn't protect membership disclosure. There is no clear division between sensitive attributes and quasi identifiers. In order to make the system more effective we are using tuple grouping algorithm with slicing. In slicing the data is partitioned both vertically and horizontally. These provide us better data utility than generalization and protect from membership disclosure. it will also handle high dimensional data. For research purpose the data is published and shared by the organization and companies.***

Keywords: Data utility, Generalization, Bucketization , slicing, Data publishing.

1. Introduction

Data mining is the extraction meaningful information from the large data such as data warehouse [4] ,micro data contains records information about an individual entity , such as a person, household or an organization ,Several micro data techniques[4] have been introduced. The most popular are Generalization [1] and Bucketization [1] [2]. There are three categories in both approaches attributes:

1) Some attributes are identifiers those can uniquely identify an individual like name or social security number.
2) Some attributes are Quasi identifiers (QI)[3][2], when taken together , can potentially identify an individual data ,eg : birth date ,sex ,and zip code;
3) Some attributes are sensitive attributes (SA's) [1][2] , which are unknown to the adversary and are considered sensitive , like disease and salary.

The two techniques differ in the next step. Generalization and bucketization[5] both , one first remove identifiers from the data and then partitions tuples into buckets . Generalization transform the QI- values [2] in each bucket into "less specific but semantically consistent"[6] [7] values so that tuples in the same bucket cannot be distinguished by their QI values. In bucketization , one separates the SA's from the QI's by randomly permuting the SA values in each bucket. The data consists of a set of buckets with SA values [8]. So avoid these attacks using different techniques. In both generalization and bucketization removes the identifiers from the data and also partitions those tuples in the form of buckets in order to avoid those attacks. Buckets contain the subset of tuples[10] . In bucketization [14][2] techniques all the sensitive information denoted "the values are well represented "[14] . We need to measure the disclosure risk of a table. The property that each record is indistinguishable with at least k-1 other records with respect to the QI's [6]. In other words k-anonymity requires that each equivalence class contains at least K records [7]. The proposed slicing algorithm [10] with tuple grouping algorithm is partitioned the data both vertically and horizontally. The random values are combined within each bucket and also can handle in high dimensional data. It is more data utility than generalization and bucketization.

Table-1 (Original data)

Sex	*ZIP code*	*Age*	*Disease*
F	5671	29	Heart Disease
F	5672	22	Heart Disease
M	5673	35	Heart Disease
F	5674	34	Flu
M	5685	40	Cancer
M	5687	32	Cancer
F	5689	50	Flu
M	5688	22	Flu
F	5680	43	Cancer

Example: Table 1 is the original data table and of it satisfying privacy. The Disease attribute is sensitive. Suppose Alice knows that Bob is a 27-year old man with zip 5678 and Bob's record is in the table. From table 2 we can conclude that Bob's record and that must have heart disease. Other example if zip code and age of Carl's is known by Alice then he can correspond to a record in the last equivalence class in table 2.

2. Literature Survey

Slicing has several advantages when compared with generalization and bucketization as it has better utility. When we use tuple grouping to achieve even more data utility and data storage than slicing. Tuple grouping can also handle high-dimensional data and data without a clear separation of QIs and SAs.

2.1 Generalization

There are several types of recordings for generalization

[1][2][3]. It preserves the information in database storage [7] [8]. First we will group tuples into buckets and then for each bucket, one replaces all values of one attribute with generalized value. Such record is called as local because the same attribute value may appear differently 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 [9]. Rather than using a generalized value to replace more specific attribute values, one uses the

Table 2: Protected data

Sex	*Zip code*	*Age*	*Disease*
*	567**	2*	Heart Disease
*	568**	2*	Heart Disease
*	567**	2*	Flu
*	567**	3*	Heart Disease
*	568**	3*	Cancer
*	567**	3*	Flu
*	568**	>=40	Cancer
*	568**	>=40	Cancer
*	567**	>=40	Flu

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.

Table 3: Generalization table

Age	*ZIP Code*	*Sex*	*Disease*
[20-50]	567*	*	Heart disease
[20-50]	567*	*	Flu
[20-50]	567*	*	Cancer
[20-40]	567*	*	Heart disease
[20-40]	567*	*	Flu
[20-40]	568*	*	Heart disease
[20-40]	568*	*	Flu
[20-40]	568*	*	Flu
[20-40]	568*	*	Cancer

2.2 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 [12] 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 [11]. 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 [1], 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 [9]. 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 background knowledge in privacy quantification [3]. A robust privacy notion has to take background knowledge into consideration. Since an adversary can easily learn background knowledge from various sources.

Table 4: Bucketized Table

Age	*ZIP Code*	*Sex*	*Disease*
22	5672	F	Heart disease
29	5671	F	Flu
34	5674	F	Cancer
35	5673	M	Heart disease
40	5675	M	Flu
22	5688	M	Heart disease
32	5687	M	Flu
43	5680	F	Flu
50	5689	M	Cancer

2.3 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 [8]. 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 [15] 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. 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 re-identified [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.

3. Existing Method

Generally in privacy preservation there is a loss of security. The privacy protection is impossible due to the presence of the knowledge in real life application. Data in its original form contains sensitive information about individuals. These data when published violate the privacy [18]. The current practice in data publishing[13] relies mainly on policies and guidelines as to what types of data can be published and on agreements on the use of published data. Privacy-preserving data publishing (PPDP) [1][7] 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 [14] called slicing is used.

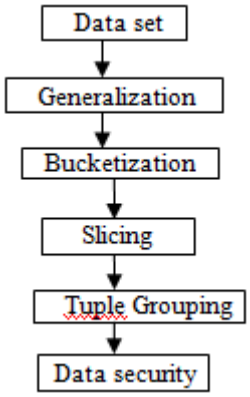

Figure 1: Functional and Slicing Architecture

4. Proposed Method

We present in this paper a novel technique called Slicing for privacy-preserving data during publishing alone with tuple grouping. Firstly, we introduce generalization and bucketization techniques for the data hiding and providing variations in the data storage. Second, we describe the Slicing techniques which have the better advantages than the above techniques. It preserves better data utility than generalization and bucketization. It correlates the SA's than bucketization. It can handle high- dimensional [6][7]data with a clear separation of QI's and SA's. We can show that slicing can effectively used for data disclosure based on the data privacy [10]. Our Slicing algorithm partitions the attributes of the database [9] into columns, applying column generalization, and buckets are formed by partitioning the tuples in same column. In our algorithm we apply column generalization [11] to make partition attributes into columns and partition tuples into buckets. Highly correlated attributes are in the same column. For better privacy we make association breaks between the correlated attributes Consider the loss of information is the problem of those methods and they doesn't protect membership disclosure. There is no clear division between sensitive attributes and quasi identifiers.

These types of attributes are less frequently and potentially identified. Our results shows that slicing provides much better data utility than generalization. . Our experiments also show the limitations of bucketization in membership disclosure protection and slicing remedies these limitations. Finally we evaluate the performance by using the tuple grouping algorithm [20][15]of two dimensional. Although we have notion of slicing in the existing algorithms, and many other techniques would possibly suit the real time databases.

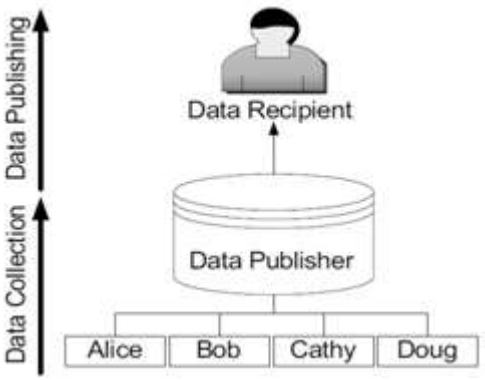

Figure 2: Process

5. Methods Used

The various methods used in this paper are as follows

5.1 Slicing Algorithms

Our algorithm consists of three phases: attribute partitioning, column generalization, and tuple partitioning. We now describe the three phases .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 [9], column generalization, and tuple partitioning. We now describe the three phases.

5.1.1 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 [8] 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 [7][4]values is much less frequent and thus more identifiable.

5.1.2 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 [7], as in the case of generalization/bucketization where each tuple can belong to only one equivalence-class/bucket.

5.1.3 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]. This provides us better data utility than generalization and protect from membership disclosure. it will also handle high dimensional data

If the sliced table after the split satisfies, 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.

Algorithm partition (T, B)

1. A = {B}; SB = {};
2. while A is not empty
3. remove the first bucket B from A;
   A = A −{B}.
4. split B into two buckets B1 and B2
5. if check(T, A ∪ {B1,B2} ∪ SB)
6. QA= A ∪ {B1,B2}.
7. else SB = SB ∪ {B}.
8. return SB.

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. This provides us better data utility than generalization and protect from membership disclosure. it will also handle high dimensional data. It is also permutated the frequency of the value in each one of algorithm checks the diversity when the

Algorithm:
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
Step4: Diversity maintains the multiple matching buckets.
Step 5: Random tuples are computed
Step 6: Attributes are combined and secure data Displayed.

6. Experimental Work

To allow direct comparison, we use two 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 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). It shows the computational time as a function of data.For simplicity of discussion, we consider only one sensi-tive attribute S[6]. This provides us better data utility than generalization and protect from membership disclosure. it will also handle high dimensional data If the data contains multiple sensitive at-tributes, one can either consider them separately or consider their joint distribution [23]. Exactly one of the c columns contains S. Without loss of generality, let the column that contains S be the last column C_c . This column is also called the **sensitive column**. All other columns $\{C_1 , C_2 , . . . , C_{c-1}\}$ contain only QI attributes.

For example, Table 1(e) and Table 1(f) are two sliced tables. In Table 1(e), the attribute partition is {{Age}, {Sex}, {Zipcode}, {Disease}} and the tuple partition is in Table 1, the attribute partition is {{Age, Sex}, {Zipcode, Disease}} and the tuple.

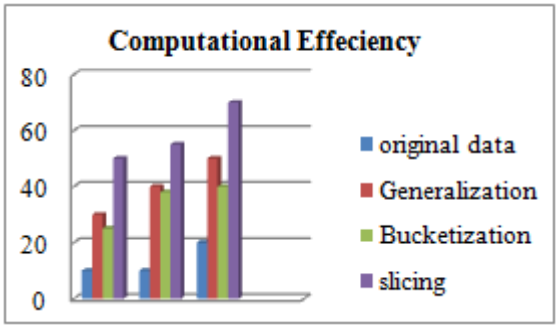

Figure 3: Graph of Slicing

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

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.

The important factor while publishing the information is better data utility, the information contains individual specific records like employees records, patients records etc. There are many techniques introduced for providing privacy, the existing system has designed with generalization and bucketization techniques along with the sclicing technique [9]. Consider the loss of information is the problem of those methods and they doesn't protect membership disclosure [10]. There is no clear division between sensitive attributes and quasi identifiers [1][2] . In order to make the system more effective we are using tuple grouping algorithm [5][9]with slicing. Consider the loss of information is the problem of those methods and they doesn't protect membership disclosure. There is no clear division between sensitive attributes and quasi identifiers. In slicing the data is partitioned both vertically and horizontally. These provide us better data utility than generalization and protect from membership disclosure. it will also handle high dimensional data. For research purpose the data is published and shared by the organization and companies.

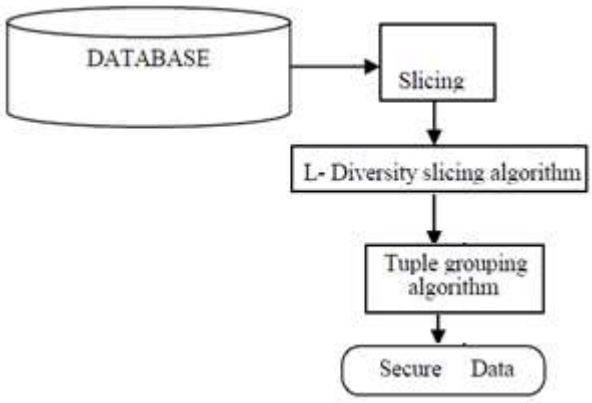

Figure 4: Tuple Grouping Architecture

7. Conclusion and Future Work

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 [15] computed by various anonymization techniques. Another important advantage of slicing is that it can handle high-dimensional data.

This work motivates several directions for future research. First, in this paper, we consider slicing where each attribute is in exactly one column. An extension is the notion of over-lapping slicing, which duplicates [5]an attribute in more than one columns. These releases more attribute correlations. For example, in Table 1, one could choose to include the Disease attribute also in the first column. That is, the two columns are {Age, Sex, Disease} and {Zip code, Disease}.This provide better data utility, but the privacy implications need to be carefully studied and understood. It is interesting to study the tradeoff between privacy and utility. These provides us better data utility than generalization and protect from membership disclosure [16] . it will also handle high dimensional data.

The implementation of previously existing systems provided clear view of the problem to be addressed. Slicing overcomes the limitations of generalization and bucketization[11] and preserves better utility while protecting against privacy threats. Consider the loss of information is the problem of those methods and they doesn't protect membership disclosure. There is no clear division between sensitive attributes [1][2] and quasi identifiers .Our experiments show that slicing preserves better data utility than generalization and is more effective than bucketization in workloads involving the sensitive attribute. These provide us better data utility than generalization and protect from membership disclosure. it will also handle high dimensional data. First, in this paper, we consider slicing where each attribute is in exactly one column. These provides us better data utility than generalization and protect from membership disclosure [4][5] . it will also handle high dimensional data

An extension is the notion of overlapping slicing, which duplicates an attribute in more than one column. Our experiments show that random grouping [9] is not very effective. The Proposed grouping algorithm [6][7]is optimized 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 [8] computed by various anonymization techniques.

References

[1] Slicing: A New Approach to Privacy Preserving Data Publishing .Tiancheng Li, Ninghui Li, Jian Zhang, Ian Molloy Purdue University, West Lafayette, 2009
[2] Ravindra S. Wanjari, Prof. Devi Kalpna/ International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 3, Issue 4, Jul-Aug 2013, pp. 119-122
[3] Enhanced K-Mean Clustering Algorithm to Reduce Number of Iterations and Time Complexity Azhar Rauf, Sheeba, Saeed Mahfooz, Shah Khusro and Huma Javed Middle-East Journal of Scientific Research 12 (7): 959-963, 2012ISSN 1990-9233© IDOSI Publications, 2012
[4] C. Aggarwal, "On k-Anonymity and the Curse of Dimensionality," Proc.of the Int'l Conf. on Very Large Data Bases (VLDB), pp. 901909, 2005.
[5] G. Aggarwal, T. Feder, K. Kenthapadi, S. Khuller, R. Panigrahy, D Thomas, and A. Zhu, "Achieving Anonymity via Clustering," Proc. Of the ACM Symp. on Principles of Database Systems (PODS), pp. 153-162, 2006.
[6] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, Network flows: theory, algorithms, and applications, Prentice-Hall, Inc., 1993.
[7] R. J. Bayardo and R. Agrawal, "Data Privacy through Optimal k-Anonymization,"217-228, 2005
[8] Tiancheng Li, Ninghui Li, Senior Member, IEEE, Jia Zhang, Member, IEEE, and Ian Molloy "Slicing: A New
[9] Approach for Privacy Preserving Data Publishing" Proc. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 24, NO. 3, MARCH 2012
[10] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati On K-Anonymity. In Springer US, Advances in Information Security (2007).
[11] 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.
[12] Ghinita.G,Tao.Y, and Kalnis.P, "OnThe Anonymization of Sparse High Dimensional Data," Proc. IEEE 24th Int"l Conf. Data Eng. (ICDE), 2008.
[13] 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.
[14] 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.
[15] 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.
[16] 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.
[17] 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.