# A Slicing Approach for Security In Data Publishing

A. R. Thorat[1], P. R. Sapkal[2], M. B. Mahajan[3],Prof. A. L. Salunke[4]
Department of Information Technology, JSPM'S BSIOTR(w),Wagholi,Pune,India

***Abstract*— We present anonymization techniques which have been made for to maintain privateness in large amount of data the anonymization technique are classified as generalization and bucketization. In generalization it loses a some amount of data and specially for high dimensional data. And another technique is bucketization which does not prevent membership disclosure and does not have clear separation between quasi-identifying and sensitive attributes. In this paper we show that new technique such as slicing which give better data utility than generalization and bucketization also it overcomes the disadvantages of anonymization techniques that are generalization and bucketization.**

***Keywords*— Data confidentiality, Anonymity, data security and data conserve**

## I. INTRODUCTION

Data mining is used to store the large amount of data and collect it into useful information. Microdata contain records of each entity which contain information about it. Generalization [6] [7] for k-anonymity [7] and bucketization [8], [5], [3] for l-diversity [4] are the microdata anonymization technique have been proposed.

In both methods attributes are into three types first is identifiers that can uniquely identify an entity like name or security number; second is quasi-identifiers. To reidentify the data, combinations of set of attributes are linked with external information such as birth date, sex and zip code.Thired is sensitive attributes. these attribute are unknown to the opponent such as disease and salary. in both generalization and bucketization techniques firstly removed identifiers from the data and partition the tuple into buckets. Generalization transforms the quasi-identifying values in each bucket into less specific as well as semantically constant hence tuple in same bucket can't be separated by their QI values. In bucketization, one separates SAs from the QI values in individual bucket. The anonymized data consist of set of bucket with rearranged SA values. Both techniques are not that much efficient for preserving data for patient. so, we are studied new technique for preserving patient data and publishing by slicing. In slicing is efficient for high dimensional data and conserves better data utility and is also used to prevent membership disclosure.

## II. EXISTING SYSTEM

### *A. Methods:*

In several anonymization methods have been introduced that are generalization and bucketization. And the term anonymization means simply text data into a human non readable format. First, generalization loses the meaningful data in the micro data mainly high dimensional data also in generalization every attribute is generalize separately and a connection between different attribute are lost. On the other hand second, the bucketization method does not forbid membership because bucketization shows the quasi-identifiers (QI) values in their original form and does not applicable to the data that does not have the clear distinguish between sensitive attributes (SA) and Quasi-identifying attributes (QI). In resent year it is difficult to privateness preserving data mining has become more important so increasing ability to store personal data about user. Above anonymization method has introduced briefly as below:

### *B. Anonymization Techniques:*

In generalization and bucketization, have been designed for privateness preserving micro data publishing. First remove identifiers from microdata and offer that partitions tuple into data.

a) Generalization: Generalization [6] [7] [5] takes over from the QI values in each bucket into "less specific but semantically consistent " so that tuple in same bucket cannot be distinguish by their QI values. There are three types of encoding scheme have been suggest for generalization:

1) Global Recording
2) Regional Recording
3) Local Recording

**1) Global Recording [8]:**

Global recording has the property that multiple occurances of the same value are always replaced by the same generalized value.

**2) Regional Recording [9]:**

It also called multi dimentional recording which partitions the domain space into non intersect region and data points in same region are represented by the region they are in.

**3) Local Recording [10]:**

Local recording does not have the above constraint and allows different occurrences of the same value to be generalized different.

| Age | Sex | Zipcode | Disease |
|---|---|---|---|
| 21 | M | 45606 | Gastritis |
| 22 | F | 45606 | Sinus |
| 31 | F | 45603 | Sinus |
| 50 | F | 45603 | Bronchitis |
| 51 | M | 45021 | Sinus |
| 58 | M | 45021 | Gastritis |
| 58 | M | 45023 | Cancer |
| 64 | F | 45023 | Cancer |

Fig (a): Original Table

| Age | Sex | Zipcode | Disease |
|---|---|---|---|
| [20-50] | * | 4560* | Gastritis |
| [20-50] | * | 4560* | Sinus |
| [20-50] | * | 4560* | Sinus |
| [20-50] | * | 4560* | Bronchitis |
| [51-64] | * | 4502* | Sinus |
| [51-64] | * | 4502* | Gastritis |
| [51-64] | * | 4502* | Cancer |
| [51-64] | * | 4502* | Cancer |

Fig (b): Generalization Table

| Age | Sex | Zipcode | Disease |
|---|---|---|---|
| 21 | M | 45606 | Gastritis |
| 22 | F | 45606 | Sinus |
| 31 | F | 45603 | Sinus |
| 50 | F | 45603 | Bronchitis |
| 51 | M | 45021 | Sinus |
| 58 | M | 45021 | Gastritis |
| 58 | M | 45023 | Cancer |
| 64 | f | 45023 | Cancer |

Fig(c): Bucketization Table

| Age | Sex | Zipcode | Disease |
|---|---|---|---|
| 21:1,22:1,31:1,52:1 | M:1,F:3 | 45606:2,45606:2 | Sinus |
| 21:1,22:1,31:1,52:1 | M:1,F:3 | 45606:2,45606:2 | Gastritis |
| 21:1,22:1,31:1,52:1 | M:1,F:3 | 45606:2,45606:2 | Bronchitis |
| 21:1,22:1,31:1,52:1 | M:1,F:3 | 45606:2,45606:2 | Sinus |
| 51:1,58:2,64:1 | M:3,F:1 | 45021:2,45023:2 | Sinus |
| 51:1,58:2,64:1 | M:3,F:1 | 45021:2,45023:2 | Cancer |
| 51:1,58:2,64:1 | M:3,F:1 | 45021:2,45023:2 | Gastritis |
| 51:1,58:2,64:1 | M:3,F:1 | 45021:2,45023:2 | Cancer |

Fig (d):Multiset-based Generalization

| Age | Sex | Zipcode | Disease |
|---|---|---|---|
| 21 | M | 45606 | Sinus |
| 22 | F | 45606 | Gastritis |
| 31 | F | 45603 | Bronchitis |
| 50 | F | 45603 | Sinus |
| 51 | M | 45021 | sinus |
| 58 | M | 45021 | cancer |
| 58 | M | 45023 | Gastritis |
| 64 | F | 45023 | Cancer |

| | | | |
|---|---|---|---|

Fig (e): One-Attribute per Column Slicing

| (Age,Sex) | (Zipcode,Disease) |
|---|---|
| (21,M) | (45606,Sinus) |
| (22,F) | (45606,Gastritis) |
| (31,F) | (45603,Bronchitis) |
| (50,F) | (45603,Sinus) |
| (51,M) | (45021,Sinus) |
| (58,M) | (45023, Cancer) |
| (58,M) | (45023, Gastritis) |
| (64,F) | (45023,Cancer) |

Fig (f): Sliced Table

## III. PROPOSED WORK

In our paper we are posing an innovative technique called slicing for security in data publishing. Our works include the following points:

1) We establish slicing as a new technique for security in data publishing. There are some advantages of slicing when differentiate with generalization and bucketization. 1. It helps to conserve more attribute with the sensitive attributes (SA) than bucketization. 2. It conserves better data utility than generalization. It is also able to deal with high-dimensional data and data which do not have a clear detachment of quasi identifiers (QI) and sensitive attributes (SA).

2)Here we introduce a expression called *l*-diverse slicing, which guarantee that the opponent cannot find out the sensitive value of *any* individual with a probability greater than 1/*l*. Depend on the privateness requirement of *l*-diversity we demonstrate that slicing can be adequately used for preventing attribute exposure.

3) We establish an efficient algorithm for computing the sliced table that satisfies *l*-diversity. Firstly this algorithm divides attributes into columns, then do column generalization, and divides tuples into buckets. Attributes having high correlativity are in the same column; this conserves the correlativity between such attributes. The associations between uncorrelated attributes are broken; better privateness is provided because the associations between such attributes are not regular and identifying.

4) Then we explain the inspiration behind membership exposure and explain how slicing prevents membership exposure. a bucket of size k matches *kc* tuples where *c* is the number of columns. Because only k of the *kc* tuples are actually in the original data, the existence of the other $kc - k$ tuples hides the membership information of tuples in the original data.

Slicing partitions the dataset into vertical and horizontal manner. Vertical partitioning: is a grouping attributes into columns based on the correlativity among the attributes. A subset of highly correlated attributes is hold within every column .Horizontal partitioning: is a grouping tuples into buckets.

5) At last to break the linkage between different columns values in each column are randomly sorted of every bucket. For reducing the dimensionality of the data and to conserve better utility than generalization and bucketization slicing breaks the association cross columns, and conserve the association within each column.
Slicing conserves data utility because it groups highly correlated attributes together, and conserves the correlativity between such attributes.privateness is protect by slicing because it breaks the associations between uncorrelated attributes, which are irregular and hence identifying. In bucketization when the dataset contains QIs and one SA, has to break their correlation but slicing, can group some QI attributes with the SA, preserving attribute correlativity with the sensitive attribute and it is one feature of slicing.

Membership exposure is stop by use of slicing. Slicing has improved data utility than generalization and can be regenerate for membership exposure shield. Larger amount of data can manage by slicing. Slicing conserves enhanced utility than generalization and is more flexible in case of assignments comprising the sensitive attribute.

**A) Slicing Algorithms:**
Here we presenting a useful slicing algorithm for obtain ℓ-diverse slicing. This includes a micro data table T and two values c and ℓ, the sliced table is calculated involving c columns and inspects the privateness requirement of ℓ-diversity. Algorithm contains three phases: 1) attribute partitioning2) column generalization and 3) tuple partitioning.

Phases are as follows:

1) Attribute Partitioning:
In this phase algorithm divides attributes such that largely related attributes are in the same column for better utility as well as privateness .Data utility is achieve by clustering highly related attributes conserves the relations among those attributes. In case of privateness the association of not related attributes shows more identification risks than that of the association of high related attributes since the association of unrelated attribute values are less common and therefore more identifiable. Thus, split the associations among uncorrelated attributes to save the privateness. This phase includes calculation of the relations within pairs of attributes and then group attributes on the basis of their correlativity.

2) Column Generalization
Records are generalized to gratify certain minimum frequency requisite. Column generalization is not an essential step in our algorithm.

3) Tuple Partitioning
This phase includes dividing records into buckets. We change Mondrian algorithm for tuple partition. We make use of the Mondrian for the reason of dividing tuples into buckets.

5) Sliced Data

Slicing can handle high-dimensional data and it is the advantage of it. Slicing reduces the dimensionality of the data by dividing attributes into columns. Each column of the table gives the output as a sub-table having lower dimensionality. Slicing is also different from the approach of publishing multiple independent sub-tables in that these sub tables are linked by the buckets in slicing.

### *A. SLICING ALGORITHM*

Algorithm of slicing: we are dividing algorithm of slicing in 2 parts.1) Tuple partition algorithm 2) L-diversity check algorithm.

**→Tuple partition algorithm[**12**]**

Step 1*:* Initially a queue of buckets Q and a set of sliced buckets SB are taken holds only single bucket which contains all tuples and SB is empty. Hence Q= {T}; SB=∅.
Step 2*:* In every Iteration the algorithm removes a bucket from Q and divides the bucket into two buckets. Q=Q-{B};
For l-diversity check (T, Q∪{B1, B2} ∪SB, l); main requirement of partitioning algorithm is to check condition that sliced table satisfies l-diversity.
Step 3: In the diversity check algorithm for every tuple t, it maintains a list of statistics L[t] contains Statistics about one matching bucket B. t∈T, L[t] =∅.The matching probability p (t, B) and the distribution of candidate sensitive values D (t, B).
Step 4: Q= Q∪{B1, B2} here two buckets are moved to the end of the Q
Step 5: else SB=SB∪{B} in this step we cannot split the bucket more so the bucket is sent to SB.
Step 6*:* Thus a final result return SB, here when Q becomes empty we have Computed the sliced table. The set of sliced buckets is SB .So, at last SB is return.

**→ Algorithm for l- Diversity-Check[**12**]**

Step 1: For each tuple t ∈ T, L[t] = ∅.
Step 2: For each bucket B in T.
Step 3: Record f (v) for each column value v in bucket B.
Step 4: For each tuple t ∈ T.
Step 5: Calculate P (t, B) and find D (t, B).
Step 6: L[t] = L[t] ∪{hp (t, B), D (t, B) i}.
Step 7: for each tuple t ∈ T.
Step 8: Calculate p (t, s) for each s based on L[t].
Step 9: if p (t, s) ≥1/L, return false.
Step 10: Return true

## IV. CONCLUSIONS

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 overlapping slicing, which duplicates an attribute in more than one columns. These releases more attribute correlations. For example, in Table, one could choose to include the Disease attribute also in the first column. That is, the two columns are {Age, Sex, and Disease} and {Zipcode, Disease}. This could provide better data utility, but the privateness implications need to be carefully studied and understood. It is interesting to study the tradeoff between privateness and utility.

Second, we plan to study membership disclosure protection in more details. Our experiments show that random grouping is not very effective. We plan to design more effective tuple grouping algorithms.

Third, slicing is a promising technique for handling high-dimensional data. By partitioning attributes into columns, we protect privateness by breaking the association of uncorrelated attributes and conserve data utility by preserving the association between highly-correlated attributes. For example, slicing can be used for anonymizing transaction databases, which has been studied recently.

Finally, while a number of anonymization techniques have been designed, it remains an open problem on how to use the anonymized data. In our experiments, we randomly generate the associations between column values of a bucket. This may lose data utility. Another direction to design data mining tasks using the anonymized data computed by various anonymization techniques.

## REFERENCES

[1]. Iancheng Li, Ninghui Li, Senior Member, IEEE, Jian Zhang, Member, IEEE, and Ian Molloy : A New Approach for Privacy Preserving Data Publishing",2012.

[2] Alphonsa Vedangi, V.Anandam Department of CSE, CMR Institute of Technology, Hyderabad, Andhra Pradesh, India: Data slicing technique to privacy preserving and data publishing.

[3] N.Koudas, D.Srivastava, T.Yu, and Q.Zhang, "Aggregate Query Answering on Anonymized Tables," Proc.IEEE 23rd Int'lConf.DataEng. (ICDE), pp.116-125, 2007.

[4] A Machanavajjhala, J.Gehrke, D.Kifer, and M.Venkitasubrama-niam, "l-Diversity: Privacy Beyond k-Anonymity," Proc.Int'l Conf. Data Eng. (ICDE), p.24, 2006.

[5] 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'lConf. Data Eng. (ICDE), pp.126-135, 2007.

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

**[7]** L.Sweeney,"k Anonymity: A Model for Protecting Privacy,"Int'lJ. Uncertainty Fuzziness and Knowledge-Based Systems, vol.10, 2002.

[8] X. Xiao and Y.Tao, "Anatomy: Simple and Effective Privacy Preservation,"Proc.Int'l Conf. Very Large Data Bases (VLDB), pp.139-150, 2006.

[9]. K. Le Fevre, D. De Witt, and R. Ramakrishnan, "In cognito: Efficient Full-Domaink-Anonymity," Proc. ACMSIGMOD Int'l Conf. Management of Data (SIGMOD), pp.49-60, 2005.

[10] K. Le Fevre ,D. DeWitt ,and R. Ramakrishnan, "Mondrian Multi-dimensional k-Anonymity," Proc. Int'l Conf .Data Eng.(ICDE), p.25,2006.

[11] J. Xu ,W. Wang ,J. Pei ,X. Wang ,B. Shi, and A.W.-C.Fu,"Utility-Based Anonymization Using Local Recoding,"Proc.12th ACMSIGKDD Int'l Conf .Knowledge Discovery and Data Mining(KDD), pp.785-790,2006.

[12] Alphonsa Vedangi *Student*, V.Anandam *Professor ,Department of CSE, CMR Institute of Technology, Hyderabad, Andhra Pradesh, India:"* DATA SLICING TECHNIQUE TO PRIVACY PRESERVING AND DATA PUBLISHING*",* Volume: 02 Issue: 10 ,Oct-2013.