Privacy Preserving Based on tuple Space Search Methods

Valeti Nagarjuna[1] , Chittineni Aruna[2],Dr. M.S.S.Sai[3]
[1]MTech Student, CSE Department, K.K.R & K.S.R Institute of Technology and Sciences, Vinjanampadu, Guntur (D.T).
[2]Assistant Professor , CSE Department, K.K.R & K.S.R Institute of Technology and Sciences, Vinjanampadu, Guntur (D.T).
[3]Professor, CSE Department,K.K.R & K.S.R Institute of Technology and Sciences, Vinjanampadu, Guntur (D.T).

Abstract: As generalization and bucketization, several anonymization techniques have been designed for publishing privacy preserving micro data. In existing approaches there is some amount of information losses by using generalization particularly in high dimensional data. There is no a clear separation between quasi-identifying attributes and sensitive attributes in case of bucketization. To overcome this problem, we proposed a approach called slicing. The data partitions both horizontally and vertically in slicing. Slicing provide better membership disclosure and better data utility over than generalization. Slicing also performs on high dimensional data. Slicing preserve efficient algorithm for to compute sliced data and protection to membership disclosure that obey the 'L-diversity requirement. Slicing provide better utility than generalization and more effective than bucketization in high dimensional data.

Index Terms—Privacy preservation, data anonymization, data publishing, data security.

I . INTRODUCTION

Privacy-preserving data mining is the area of data mining that used to safeguard sensitive information from unsanctioned disclosure. In recent years Privacy preserving publishing of micro data has been studied extensively. For privacy preserving publishing of micro data, several techniques has been developed like generalization for K-anonymity and bucketization for l- Diversity. In both generalization and bucketization attributes are partitioned into three categories they are

1. Identifiers
2. Quasi Identifiers (QI)
3. Sensitive Attributes (SAs)

Dimensionality problem arise in generalization for k- anonymity. In generalization, most of the data sets have the same distance with each other because we have to force huge amount of generalization to satisfy k-anonymity even for relatively small k's. So generalization reduces data utility in privacy preserving data.

Compared to generalization, bucketization provide better data utility. Bucketization has several limitation like it provides Quasi Identifiers (QI) values in their original forms because attacker may find out whether an individual has a record in the published data or not, so we can inferred the membership information from bucketization table. So bucketization does not prevent membership disclosure. In bucketization we need clear separation between Quasi Identifiers (QI) and Sensitive Attributes (SAs) but bucketization breaks the attribute correlations between the QIs and the SAs.

In this paper, we introduce slicing concept to improve better data utility. Slicing divide the data set both vertically and horizontally. Attributes grouped into columns based on the correlations among the attributes in vertical partitioning. Highly correlated attributes are present in columns.Tuples grouped into buckets in horizontal partitioning. Slicing provide better data utility by reducing the dimensionality of data over than generalization and bucketization.

II. EXISTING SYSTEM:

For privacy publishing micro data generalization and bucketization techniques based on k-anonymity, l- diversity approaches were used. Generalization fails to provide better data utility and high dimensional data. Bucketization fails to prevent membership disclosure. Bucketization does not provide clear separation between Quasi Identifiers and Sensitive Attributes. K-anonymity does not provide sufficient protection against information loss in case of high dimensional data. L-diversity protects against attribute disclosures but fails to prevent probabilistic attacks. So we introduce a novel approach called slicing.

Generalization:

The generalization is one of the commonly used anonymized approaches. That is used to replace quasi-identifier values with values that are less specific but semantically constant. All quasi –identifier values in a group would be generalized to the entire group context in the QID space.

Bucketization:

Bucketization is another approach, it is used to partition the tuple's in T into buckets and then to separate sensitive attributes from non sensitive attributes by randomly permitting the sensitive attribute values within each bucket.

III. PROPOSED WORK

We proposed a data anonymization technique called slicing to improve current state of the art. Slicing partitions the data both vertically and horizontally. Attributes grouped into columns based on the correlations among the attributes in vertical partitioning. Highly correlated attributes are present in columns.Tuples grouped into buckets in horizontal partitioning. values in each column are randomly permutated to break the linking between different columns in each column. Slicing is to break the association cross columns, but to preserve the association within each column. Slicing provide better data utility by reducing the dimensionality of data over than generalization and bucketization. Slicing groups highly correlated attributes and preserve better utility. Slicing breaks the associations between uncorrelated attributes because protects privacy.

Slicing:

Data slicing is a novel approach ,which is used to partitions the data both the horizontally and vertically. It is used to reduce the dimensionality of the data and preserves better data slicing method consists of three stages.

1. Partitiong attributes and columns
2. Partition tuple's and buckets
3. Generalization of buckets

IV. EXPERIMENTAL RESULTS

To achieve L-diverse slicing, we introduce new efficient slicing algorithm. The algorithm computes the sliced table that consists of c columns and satisfies the privacy requirement of L-diversity. L- diversity algorithm mainly consists of three phases they are

1.Attribute partitioning
2. Column generalization
3. Tuple partitioning

1. Attribute Partitioning:

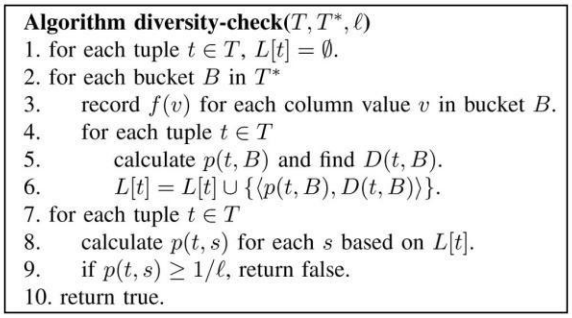
Attribute Partitioning algorithm partitions attributes so highly correlated attributes are in the same column. This algorithm provides utility and privacy. It provide utility by grouping the high correlated attributes and it is also provide privacy association of uncorrelated attributes presents higher identification risks than the association of highly correlated attributes because the association of uncorrelated attribute values is much less request and thus more identifiable.

2. Column generalization:

Column generalization is useful in several aspects. Column generalization may be required for identity/membership disclosure protection. To achieve the same level of privacy against attribute disclosure Column generalization technique is applied in case of small size buckets. While column generalization may result in information loss, smaller bucket-sizes allow better data utility**.**

3. Tuple partitioning

Tuple partitioning algorithm partitions tuples into buckets. Fig. 1 gives the description of the tuple-partition algorithm



```
Algorithm diversity-check($T, T^*, \ell$)
1. for each tuple $t \in T$, $L[t] = \emptyset$.
2. for each bucket $B$ in $T^*$
3.     record $f(v)$ for each column value $v$ in bucket $B$.
4.     for each tuple $t \in T$
5.         calculate $p(t, B)$ and find $D(t, B)$.
6.         $L[t] = L[t] \cup \{\langle p(t, B), D(t, B)\rangle\}$.
7. for each tuple $t \in T$
8.     calculate $p(t, s)$ for each $s$ based on $L[t]$.
9.     if $p(t, s) \geq 1/\ell$, return false.
10. return true.
```

Fig. 1. The tuple-partition algorithm

The tuple-partition algorithm mainly used to check whether a sliced table satisfies L-diversity. In each iteration (lines 2 to 7), the algorithm removes a bucket from Q and splits the bucket into two buckets. If the sliced table after the split satisfies ‘L-diversity (line 5), then the algorithm puts the two buckets at the end of the queue Q .we cannot split the bucket anymore and the algorithm puts the bucket into SB.

V. CONCLUSION

To privacy preserving publishing data, we introduce a new approach called slicing. Slicing is used overcome limitations of generalization and bucketization and preserves better utility while protecting against privacy threats. Our proposed approach provides better data utility and attribute membership disclosure over than generalization and bucketization. We ropose to replace random grouping with more effective tuple grouping algorithm such as tuple space serch algorithm.the main purpose of tuple space serch algorithm is to speed up over all slicing process to support large data. Slicing with tuple grouping algorithm provides random tuple grouping for micro data publishing.

VI. REFERENCE’s

[1] C. Aggarwal, “On k-Anonymity and the Curse of Dimensionality,” Proc. Int’l Conf. Very Large Data Bases (VLDB), pp. 901-909, 2005.

[2] A. Blum, C. Dwork, F. McSherry, and K. Nissim, “Practical Privacy: The SULQ ramework,” Proc. ACM Symp. Principles of Database Systems (PODS), pp. 128-138, 2005.

[3] 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.

[4] B.-C. Chen, K. LeFevre, and R. Ramakrishnan,

"Privacy Skyline: Privacy with Multidimensional Adversarial Knowledge," Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 770-781, 2007.

[5] H. Cramt'er, Mathematical Methods of Statistics. Princeton Univ. Press, 1948.

[6] I. Dinur and K. Nissim, "Revealing Information while Preserving Privacy," Proc. ACM Symp. Principles of Database Systems (PODS), pp. 202- 210, 2003.

[7] C. Dwork, "Differential Privacy," Proc. Int'l Colloquium Automata, Languages and Programming (ICALP), pp. 1-12, 2006.

[8] 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.

[9] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating Noise to Sensitivity in Private Data Analysis," Proc. Theory of Cryptography Conf. (TCC), pp. 265-284, 2006.

[10] J.H. Friedman, J.L. Bentley, and R.A. Finkel, "An Algorithm for Finding Best Matches in Logarithmic Expected Time," ACM Trans. Math. Software, vol. 3, no. 3, pp. 209-226, 1977.