Hybrid Techniques for Data Privacy Preserving

Vinita Mishra[1], Smita Jangale[2]

[1] Assistant Professor, Information Technology, V.E.S Institute of Technology
Mumbai, INDIA, E-mail: *vinita.mishra@ves.ac.in*

[2]Associate professor, Information Technology, V.E.S Institute of Technology
Mumbai, INDIA, E-mail: *smita.jangale@ves.ac.in*

Abstract: Today, most enterprises have a need of collecting and storing data in large databases. It has been realized that these data are information source for making business decisions. Privacy-preserving data publishing (PPDP) provides methods for publishing useful information while preserving data privacy. In this paper, a brief review of several anonymization techniques such as generalization and bucketization, slicing, overlap slicing and generalized- slicing have been designed for privacy preserving micro data publishing. Recent work has shown that generalization loses considerable amount of information and doesn't protect attribute disclosure, 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 focus on effective method that can be used for providing better data utility and can also protect against attribute, membership and identity disclosures.

Keywords: Bucketization, Generalization, Slicing, Overlap-slicing, Generalized-slicing

I. INTRODUCTION

Society is experiencing exponential growth in the number and variety of data collections containing person-specific information. Huge databases exist in today's society. The huge amount of data available means that it is possible to learn lot of information about individuals from public data. While doing, the privacy of the data should be maintained. So, to preserve the privacy of the data, privacy preserving methods are implemented. Three of the most widely used techniques are generalization, bucketization and Slicing. Bucketization doesn't prevent membership disclosure and it doesn't apply for data that don't have a clear distinction between quasi-identifiers and sensitive attribute. Generalization loses high amount of data and doesn't preserve identity disclosure Slicing provides better data utility but still its prone to attacks. Slicing protects the data against membership and attribute disclosure but it doesn't provide any details about identity disclosure. To overcome this an efficient technique generalized-slicing has been introduced to increase the overall utility and privacy of data .In this paper, another efficient method for preserving Privacy is introduced in which data can be partitioned both vertically and horizontally and overlapped. Major advantage of overlap slicing is that it works on high-dimensional data. Also, it gives better membership disclosure than slicing. Table 1 is the original table on which the techniques will be performed.

Table 1: Original Data

Age	*Work class*	*Education*	*Marital status*	*Race*	*Sex*	*Occupation*
39	State-Emp	bachelors	Never Married	white	Male	Adm-Clerical
50	Self-Emp	bachelors	Married	White	Male	Exec-Managerial
38	Private	HS-grade	Divorced	White	Male	Handlers-Cleaners
53	Private	11^{th}-grade	Married	Black	Male	Handlers-Cleaners
28	Private	bachelors	Married	Black	Female	Prof-Specialty
37	Private	Masters	Married	White	Female	Exec-Managerial
49	Private	9^{th} grade	Married	Black	Female	Other-service
52	Self-Emp	HS--grade	Married	White	Male	Exec-Managerial
31	Private	Masters	Never-Married	white	Female	Prof-Specialty

II. BACKGROUND OF THE PROPOSED WORK

A. *Generalization*

Generalization is one of the most common anonymization technique, 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. 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. 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 have extremely high information loss, leaving data useless. 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.

Limitation of Generalization:

The main problems with generalization are: 1) it fails on high-dimensional data due to the curse of dimensionality 2) It causes too much information loss due to the uniform-distribution assumption. The Generalized data is shown in the table 2.

Table 2: Generalized Data

Age	*Work class*	*Education*	*Marital status*	*Race*	*Sex*	*Occupation*
[30-50]	employed	educated	*	person	*	Adm-Clerical
[30-50]	employed	educated	*	person	*	Exec-Managerial
[30-50]	employed	educated	*	person	*	Handlers-Cleaners
[20-60]	employed	educated	*	person	*	Handlers-Cleaners
[20-60]	employed	educated	*	person	*	Prof-Specialty
[20-60]	employed	educated	*	person	*	Exec-Managerial
[30-60]	employed	educated	*	person	*	Other-service
[30-60]	employed	educated	*	person	*	Exec-Managerial
[30-60]	employed	educated	*	person	*	Prof-Specialty

B. Bucketization

Bucketization, partitions the tuples in T into buckets, and then separates the sensitive attribute from the non-sensitive ones by randomly permuting the sensitive attribute values within each bucket. The final data then consists of the buckets with permuted sensitive values. 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. Mostly, bucketization has been used for anonymizing high-dimensional data. However, this approach assumes a clear separation between QIs and SAs.

Limitation of Bucketization:

1) Bucketization doesn't prevent membership disclosure.
2) Requrires clear seperation between QI ans SA.
3) It breaks correlation between QI and SA.

The bucketized data is given in table 3.

Table 3: Bucketized data

Quasi -Identifiers						***Sensitive-attribute***
Age	***Work class***	***Education***	***Marital status***	***Race***	***Sex***	***Occupation***
39	State-Emp	bachelors	Never	white	Male	Exec-Managerial
50	Self-Emp	bachelors	Married	White	Male	Handlers-Cleaners
38	Private	HS-grade	Married Divorced	White	Male	Adm-Clerical
53	Private	11th-grade	Married	Black	Male	Exec-Managerial
28	Private	bachelors	Married	Black	Female	Handlers-Cleaners
37	Private	Masters	Married	White	Female	Prof-Specialty
49	Private	9th grade	Married	Black	Female	Prof-Specialty
52	Self-Emp	HS--grade	Married	White	Male	Other-service
31	Private	Masters	Never-Married	white	Female	Exec-Managerial

C. Slicing

A next method developed for privacy-preserving is slicing. Slicing is better than Generalization and Bucketization in terms that it can handle high-dimensional data without a clear separation of QIs and SAs. Next, it provides better data utility than generalization. Slicing preserves more attribute correlations with the Sensitive Attributes (SAs) than Bucketization. An efficient algorithm is developed to compute the sliced table that satisfies l-diversity. This algorithm divides attributes into columns, and then divides tuples into buckets. Attributes that are highly correlated are in the same column; this preserves the correlations between such attributes. In Slicing, it partitions the dataset both vertically and horizontally. Horizontal partitioning group's tuples into buckets. In each bucket, values in each column are randomly sorted to break the relation between different columns. On the other hand vertical partitioning groups attributes into columns based on the correlations between the attributes. Each column consists of a subset of attributes that are highly correlated. The main idea behind slicing is to break the association cross columns, but with this it

also preserves the association within each column. Slicing decreases the dimensionality of the data. Slicing preserves better utility than generalization and Bucketization.

The tuple-partition algorithm:

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. In 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 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. The main part of the tuple-partition algorithm is to check whether a sliced table satisfies l-diversity.

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 information about one matching bucket B, matching probability p(t,B) and the distribution of candidate sensitive values D(t,B). The algorithm first scan each bucket B once to record the frequency f (v) of each column value v in bucket B. 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 sensitive values D(t,B) for each candidate, which are added to the list L[t] . At the end of line, we have obtained the list of statistics L[t] for each tuple about its matching buckets. Based on the law of total probability a final scan of the tuples in T will compute the p (t, s).

Slicing overcomes the limitations of generalization and bucketization and preserves better utility while protecting against privacy threats. Slicing prevents membership disclosure. It is in general hard to learn sensitive information about an individual if you don't even know whether this individual's record is in the data or not.

Limitation of Slicing:

1) Slicing doesn't prevent attribute correlation.2) When more number of similar attribute value and sensitive value may be present in different tuple may give the original tuple while performing random permutation.3) the utility of dataset is lost by generation of fake tuples**.** The sliced data is given in the table 4.

Table 4: Sliced data

(Age, Work class, Marital status,Race,Sex)	***(Education, Occupation)***
(39,State-Emp,Never Married,white,Male)	(bachelors ,Exec-Managerial)
(50,Self-Emp,Married,White,Male)	(HS-grade ,Handlers-Cleaners)
(38,Private,Divorced,White,Male)	(bachelors ,Adm-Clerical)
(53,Private,Married,Black,Male)	(Masters ,Exec-Managerial)
(28,Private,Married,Black,Female)	(11^{th}-grade ,Handlers-Cleaners)
(37,Private,Married,White,Female)	(bachelors ,Prof-Specialty)
(49,Private,Married,Black,Female)	(HS--grade ,Exec-Managerial)
(52,Self-Emp,Married,White,Male)	(Masters,Prof-Specialty)
(31,Private,Never-Married,white,Female)	(9^{th} grade ,Other-service)

III. PROPOSED METHODS

Overlap Slicing:

Here in Overlap slicing, the attributes are duplicated in more than one column. This helps to achieve more correlation between the attributes. Overlap slicing partitions attribute both horizontally and vertically. In vertical partitioning more correlated attributed are taken into one group and uncorrelated attributed are grouped separately. In horizontal partitioning tuples are grouped to form buckets, after grouping tuples values of column are randomly permuted.

Overlapping slicing works in two main steps:

1. Attribute partitioning
2. Tuple partitioning

Attribute partitioning: In attribute partitioning, correlations of the attribute are measured to form there group. To measure the correlation mean square contingency coefficient is used.

Tuple partitioning: In this step tuples are grouped to form bucket. Mondrian algorithm is used for tuple partitioning.

Algorithm tuple-partition (T, t)

1. Q = {T}
2. While Q is not empty
3. Remove the first bucket B from Q; Q = Q − {B}.
4. Split B into two buckets B1 and B2, as in Mondrian.
5. If t closeness-check (T, Q ∪ {B1, B2} ∪ SB, t)
6. Q = Q ∪ {B1, B2}.
7. Else SB = SB ∪ {B}.
8. Return SB

For example the first bucket in table 5 values of the attribute Age, Marital status, race, sex contains original values. In the next column duplicate attribute Occupation are randomly permutated. Membership disclosure is protected in Overlapped slicing. To protect membership information, we must ensure that at least some tuples should also have matching buckets. Otherwise, the adversary can differentiate by examining the number of matching buckets. When the number of fake tuples is 0, the membership information of every tuple can be determined. Membership information is protected because the adversary cannot distinguish original tuples from fake tuples. Overlap Slicing is an effective technique for membership disclosure protection .Here the value of sensitive attribute is randomly permuted to achieve more privacy. Sensitive attributes are partitioned with both attribute therefore more attribute correlation is achieved and utility of data is increased.

Generalized Slicing:

In this paper, a robust slicing technique called Generalized-slicing for privacy- preserving data publishing is proposed. Generalized-slicing, is introduced which ensures that the attacker cannot learn the sensitive value of any individual at any cost and the privacy is preserved.

Table 5: Overlap Slicing

(Age, Marital status,Race,Sex)	*(Work class, Occupation)*	*(Education, Occupation)*
(39,Never Married,white,Male)	(Self-Emp, Exec-managerial)	(HS-grade, Handlers-cleaner)
(50, Married,White,Male)	(Private, Handlers-Cleaners)	(bachelor, Exec-Managerial)
(38, Divorced,White,Male)	(state-Emp, Adm-Clerical)	(bachelors ,Adm-Clerical)
(53,Married,Black,Male)	(Private, Exec-Managerial)	(11^{th}grade ,Handlers-cleaner)
(28,Married,Black,Female)	(Private,Prof-Specialty)	(Masters, Exec-Managerial)
(37,Married,White,Female)	(Private, Handlers-Cleaners)	(bachelors ,Prof-Specialty)
(49, Married,Black,Female)	(Self-Emp, Exec-Managerial)	(HS--grade ,Exec-Managerial)
(52, Married,White,Male)	(Private, Other-service)	(Masters,Prof-Specialty)
(31,NeverMarried,white,Female)	(Private, Prof-Specialty)	(9^{th} grade ,Other-service)

Generalized-Slicing works in three main steps

1. Partitioning the attributes into columns
2. Partitioning tuples into buckets.
3. Generalized-Slicing

The first two steps are similar to slicing. In the last step the sliced table can be minimized by omitting QIs for reducing the dimensionality of the data and generalizing some QIs or providing maximum privacy and minimum utility. Highly correlated attributes are in the same column; this preserves the correlations between such attributes. The associations between uncorrelated attributes are broken; this provides better privacy as the associations between such attributes are less- frequent and potentially identifying. We describe the membership disclosure and explain how Generalized-slicing prevents membership disclosure. A bucket of size k can potentially match *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. Generalized-Slicing partitions the dataset both vertically and horizontally and perform minimization and masking of QI's .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 correlated. Horizontal partitioning is done by grouping tuples into buckets. Within each bucket, values in each column are randomly permutated. This 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 a bucketization.Generalized-Slicing group highly correlated attributes together, and preserves the correlations between such attributes and protects privacy as it breaks the associations between uncorrelated attributes that are infrequent and hence identifying. When the dataset contains QIs and one SA, bucketization has to break their correlation; Generalized-slicing, on the other hand, can group and minimizes some QI attributes with the SA, preserving attribute correlations with the sensitive attribute. Generalized-Slicing has improved data utility than generalization and slicing. Additional important benefit of Generalized-slicing is that it can manage data with greater dimension. Generalized-Slicing can completely stops membership exposure and also identity disclosure.

Table 6: Generalized Slicing

(Age, Work class, Marital status,Race,Sex)	*(Education, Occupation)*
([30-50],Employed, Never Married,Person,Male)	(bachelors ,Exec-Managerial)
([30-50],Employed, Married,Person,Male)	(HS-grade ,Handlers-Cleaners)
([30-50],Employed, Divorced,Person,Male)	(bachelors ,Adm-Clerical)
([20-60],Employed, Married,Person,Male)	(Masters ,Exec-Managerial)
(([20-60],Employed, Married,Person,Female)	(11^{th}-grade ,Handlers-Cleaners)
(([20-60],Employed, Married,Person,Female)	(bachelors ,Prof-Specialty)
([30-60],Employed, Married,Person,Female)	(HS--grade ,Exec-Managerial)
([30-60],Employed, Married,Person,Male)	(Masters,Prof-Specialty)
([30-60],Employed, Never-Married,Person,Female)	(9^{th} grade ,Other-service)

IV. CONCLUSION

Slicing overcomes the limitations of generalization and bucketization and preserves better utility while protecting against privacy threats. It is hard to learn sensitive information about an individual if you don't even know whether this individual's record is in the data or not. Thus from our study we analyzed that Slicing overcomes the limitations of existing techniques of generalization and bucketization and pre-serves better utility while protecting against privacy threats. Slicing protects attribute disclosure and membership disclosure. Slicing preserves better data utility than generalization and is more operative than bucketization regarding the sensitive attribute. Anonymization technique is powerful method for privacy preserving of published data. Overlap Slicing overcomes the limitations of slicing and preserves better utility while protecting against privacy threats. Overlapping slicing is used to prevent attribute disclosures and better membership disclosure. In the last Generalized Slicing protects membership, identity and attribute disclosure.

Table 7: Comparison between different Privacy preserving techniques

Parameters	***Anonymization Techniques***				
	Generalization	***Bucketization***	***Slicing***	***Overlap slicing***	***Generalized slicing***
Membership disclosure	Protects membership disclosure	Doesn't protect membership disclosure	Protects membership disclosure	Protect membership disclosure	Protects membership disclosure
Identity disclosure	Protects identity disclosure	Doesn't protect identity disclosure	Doesn't protect identity disclosure	Doesn't protect identity disclosure	Protects identity disclosure
Attribute Disclosure	Doesn't protect attribute disclosure	Protects attribute disclosure	Protects attribute disclosure	Protects attribute disclosure	Protects attribute disclosure

V. REFERENCES

[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] B.-C. Chen, K. LeFevre, and R. Ramakrishna, "Privacy Skyline: Privacy with Multidimensional Adversarial Knowledge," Proc.Int'l Conf. Very Large Data Bases (VLDB), pp. 770-781, 2007.
[3] I. Dinur and K. Nissim, "Revealing Information while Preserving Privacy," Proc. ACM Symp. Principles of Database Systems (PODS), pp. 202-210, 2003.
[4] C. Dwork, "Differential Privacy," Proc. Int'l Colloquium Automata, Languages and Programming (ICALP), pp. 1-12, 2006.
[5] 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.
[6] B.C.M. Fung, K. Wang, and P.S. Yu, "Top-Down Specialization for Information and Privacy Preservation," Proc. Int'l Conf. Data Eng. (ICDE), pp. 205-216, 2005.
[7] 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.
[8] Y. He and J. Naughton, "Anonymization of Set-Valued Data via Top-Down, Local Generalization," Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 934-945, 2009.
[9] D. Kifer and J. Gehrke, "Injecting Utility into Anonymized Data Sets," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD), pp. 217-228, 2006.
[10] N. Koudas, D. Srivastava, T. Yu, and Q. Zhang, "Aggregate Query Answering on Anonymized Tables," Proc. IEEE 23rd Int'l Conf. Data Eng. (ICDE), pp. 116-125, 2007.
[11] K. LeFevre, D. DeWitt, and R. Ramakrishna, "Incognito: Efficient Full-Domain k-Anonymity," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD), pp. 49-60, 2005.
[12] K. LeFevre, D. DeWitt, and R. Ramakrishna, "Mondrian Multidimensional k-Anonymity," Proc. Int'l Conf. Data Eng. (ICDE), p. 25, 2006.
[13] K. LeFevre, D. DeWitt, and R. Ramakrishna, "Workload-Aware Anonymization," Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 277-286, 2006.
[14] N. Li, T. Li, and S. Venkatasubramanian, "t-Closeness: Privacy Beyond k-Anonymity and '-Diversity," Proc. IEEE 23rd Int'l Conf. Data Eng. (ICDE), pp. 106-115, 2007.
[15] T. Li and N. Li, "Injector: Mining Background Knowledge for Data Anonymization," Proc. IEEE 24th Int'l Conf. Data Eng. (ICDE), pp. 446-455, 2008.
[16] T. Li and N. Li, "On the Tradeoff between Privacy and Utility in Data Publishing," Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 517-526, 2009.
[17] T. Li, N. Li, and J. Zhang, "Modeling and Integrating Background Knowledge in Data Anonymization," Proc. IEEE 25th Int'l Conf. Data Eng. (ICDE), pp. 6-17, 2009.
[18] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "'l-Diversity: Privacy Beyond k-Anonymity," Proc. Int'l Conf.Data Eng. (ICDE), p. 24, 2006.
[19] 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.

