International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol. 3, Issue. 4, Jul - Aug. 2013 pp-2581-2584 ISSN: 2249-6645

A Novel Method for Privacy Preserving Micro data Publishing using Slicing

Shaik Khasimbee[1], Syed Sadat Ali[2]
[1]*M. Tech, Nimra College of Engineering & Technology, Vijayawada, A.P., India.*
[2]*Assoc. Professor & Head, Dept.of CSE, Nimra College of Engineering & Technology, Vijayawada, A.P., India.*

***Abstract*:** *Data anonymization techniques for privacy-preserving data publishing have received a lot of attention in recent years. Microdata or detailed data contains information about a person, a household or an organization. Most popular anonymization techniques are: Generalization and Bucketization. Generalization transforms the Quasi-Identifiers (QI) in each bucket into "less specific but semantically consistent" values so that tuples in the same bucket cannot be distinguished by their QI- values. In bucketization, one separates the Sensitive Attributes (SAs) from the QIs by randomly permuting the SA values in each bucket. The process of Generalization loses considerable amount of information, especially for high-dimensional data. Where as Bucketization does not prevent membership disclosure and does not apply for data that do not have a clear separation between quasi- identifying attributes and sensitive attributes. To improve the current state of the art, in this paper, we propose a novel data anonymization technique called slicing. Slicing method preserves utility because it groups highly correlated attributes together, and preserves the correlations between such attributes. Slicing protects privacy because it breaks the associations between the uncorrelated attributes, which are infrequent and thus identifying.*

Keywords: *Bucketization, Generalization, Identifier, Slicing.*

I. INTRODUCTION

Data sharing has become common now- a- days and there is an exponential growth in the amount of information. Data mining is the process of extraction of large amount of hidden useful information from large databases. Privacy-preserving data mining (PPDM) deals with obtaining valid data mining results without underlying the data values. The problem of privacy preserving data publishing has received a lot of attention in recent years. Agencies and other organizations often need to publish micro data, e.g., census data, medical data, etc for research and other purposes. When releasing microdata, the association of quasi identifiers with sensitive attributes in the public records has long been recognized as a privacy risk. Microdata contains records each of which contains information about an individual entity, a household, such as a person, or an organization. Typically, microdata is stored in a table, and each record or row corresponds to one individual. Each record has a number of attributes or fields, which can be divided into the following three categories:

- **Identifier:** Identifiers are attributes or fields that clearly identify individuals. Examples include Social Security Number (SSN) and Name.
- **Quasi-Identifier:** Quasi-identifiers (QI) are attributes whose values when taken together can potentially identify an individual. Examples include Birthdate, Zip-code, and Gender. An adversary may already know the QI- values of some individuals in the data. This knowledge can be either from personal contact or from other publicly available databases (e.g., a voter registration list) that include both explicit identifiers and the quasi-identifiers.
- **Sensitive Attribute:** Sensitive attributes are the attributes whose values should not be associated with an individual by the adversary. Examples include Salary and Disease.

An example of microdata table is shown in Table 1.

Table 1: Micro data Example

Age	Sex	Zipcode	Disease
22	M	47906	dyspepsia
22	F	47906	flu
33	F	47905	flu
52	F	47905	bronchitis
54	M	47302	flu
60	M	47302	dyspepsia
60	M	47304	dyspepsia
64	F	47304	gastritis

II. ANONYMIZATION METHODS

A. Generalization

Generalization [1] [2] is the process of replacing a value with a "less-specific but semantically consistent" value. Tuple suppression removes the entire record from the table. Unlike traditional privacy protection techniques such as data swapping and adding noise, information in a k- anonymized table through generalization process remains truthful. For

example, through generalization, Table 2 is an anonymized version of the microdata table in Table 1. Typically, generalization process utilizes a value generalization hierarchy (VGH) for each attribute. In a VGH, the leaf nodes correspond to actual attribute values, and internal nodes represent less-specific values.

Table 2: Generalization

Age	Sex	Zipcode	Disease
[20-52]	*	4790*	dyspepsia
[20-52]	*	4790*	flu
[20-52]	*	4790*	flu
[20-52]	*	4790*	bronchitis
[54-64]	*	4730*	flu
[54-64]	*	4730*	dyspepsia
[54-64]	*	4730*	dyspepsia
[54-64]	*	4730*	gastritis

B. Bucketization

Another anonymization method is called. Bucketization. Bucketization is also known as anatomy or permutation-based anonymization [3][4]. The bucketization process 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 consists of a set of buckets with the permuted sensitive attribute values. The main difference between generalization and bucketization methods lies in that bucketization does not generalize the QI attributes. When the adversary knows who are in the microdata table and their QI attribute values, the two anonymization techniques become equivalent. Table 3 gives the bucketization of data in Table 1.

Table 3: Bucketization

Age	Sex	Zipcode	Disease
22	M	47906	flu
22	F	47906	dyspepsia
33	F	47905	bronchitis
52	F	47905	flu
54	M	47302	gastritis
60	M	47302	flu
60	M	47304	dyspepsia
64	F	47304	dyspepsia

III. NEED OF SLICING

Generalization meyhod transforms the QI-values in each bucket into “less specific but semantically consistent” values so that tuples in the same bucket cannot be distinguished by their QI values. In bucketization, one separates the SAs from the QIs by randomly permuting the sensitive attributes values in each bucket. The anonymized data consists of a set of buckets with the permuted sensitive attribute values. It has been shown[5][6] that generalization for k-anonymity losses considerable amount of information, especially for high-dimensional data. This is due to the following three reasons: First, generalization for k-anonymity suffers from the curse of the dimensionality. Second, in order to perform the data analysis or data mining tasks on the generalized table, the data analyst has to make the uniform distribution assumption that every value in a generalized interval or set is equally possible, as no other distribution assumption can be justified. Third, because each attribute is generalized separately, then correlations between different attributes are lost.

While bucketization method [3][4] has better data utility than generalization, it has several limitations. First, bucketization method does not prevent membership disclosure [7]. Because bucketization method publishes the QI values in their original forms, an adversary can find out whether an individual has a record in the published data or not. As shown in [2], 87 percent of the individuals in the United States can be uniquely identified using only three attributes (Birthdate, Sex, and Zipcode). Second, bucketization method requires a clear separation between QIs and SAs. Third, by separating the sensitive attribute from the QI attributes, bucketization method breaks the attribute correlations between the QIs and the SAs.

Table 4: Slicing

(Age,Sex)	(Zipcode,Disease)
(22,M)	(47905,flu)
(22,F)	(47906,dysp.)
(33,F)	(47905,bron.)
(52,F)	(47906,flu)
(54,M)	(47304,gast.)
(60,M)	(47302,flu)
(60,M)	(47302,dysp.)
(64,F)	(47304,dysp.)

Slicing is the process of partitioning the dataset both vertically and horizontally. Vertical partitioning is done by grouping attributes into various columns based on the correlations among the attributes. Each column contains a subset of the attributes that are highly correlated. Horizontal partitioning is done by grouping the tuples into buckets. Finally, within each bucket, the values in each column are randomly permutated or sorted to break the linking between different columns. The basic idea of slicing method is to 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 bucketization methods.

IV. SLICING ALGORITHM

The proposed algorithm consists of three phases: attribute partitioning, column generalization, and tuple partitioning.

A. Attribute Partitioning

Attribute portioning phase partitions attributes so that highly-correlated attributes are in the same column. This is good for both utility as well as 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 that of the association of highly-correlated attributes because the association of uncorrelated attributes values is much less frequent and thus is more identifiable. Therefore, it is better to break the associations between the uncorrelated attributes, in order to protect privacy. In this phase, we first compute the correlations between a pairs of attributes and then cluster attributes based on their correlations.

B. Column Generalization

In the column generalization phase, tuples are generalized to satisfy some minimal frequency requirement. We want to point out that column generalization is not an indispensable phase in slicing algorithm. Bucketization method provides the same level of privacy protection as generalization, with respect to attribute disclosure. Although column generalization is not a required step, it can be useful in several aspects. First, column generalization phase may be required for identity/membership disclosure protection. Second, when column generalization phase is applied, to achieve the same level of privacy against attribute disclosure, bucket sizes can be smaller.

C. Tuple Partitioning

In the tuple partitioning phase, tuples are partitioned into various buckets. We modify theMondrian [8] algorithm for tuple partition phase. Unlike Mondrian k-anonymity, no generalization method is applied to the tuples; we use Mondrian for the purpose of partitioning the tuples into buckets. Algorithm 1 gives the description of 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 (line 1). In each iteration (line 2 to line 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 this algorithm puts the two buckets at the end of the queue Q (for more splits, line 6). Otherwise, we cannot split the bucket anymore and then the algorithm puts the bucket into SB (line 7). When "Q" becomes empty, we have computed the sliced table. The set of sliced buckets is "SB" (line 8).

Algorithm 1: Tuple-partition(T, ℓ**)**
1. Q = {T}; SB = ∅.
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 **diversity-check**(T, Q ∪ {B1,B2} ∪ SB, ℓ)
6. Q = Q ∪ {B1,B2}.
7. else SB = SB ∪ {B}.
8. return SB.

The main part of algorithm 1 is to check whether a sliced table satisfies ℓ-diversity (line 5). Algorithm 2 gives a brief description of the diversity-check algorithm.

Algorithm 2: Diversity-check(T, T^*, ℓ**)**
1. for each tuple t ∈ T, L[t] = ∅.
2. for each bucket B in T^*
3. record f(v) for each column value v in bucket B.
4. for each tuple t ∈ T
5. calculate p(t,B) and find D(t,B).
6. L[t] = L[t] ∪ {<p(t,B),D(t,B)>}.
7. for each tuple t ∈ T
8. calculate p(t, s) for each s based on L[t].
9. if p(t, s) ≥ 1/ℓ, return false.
10. return true.

Algorithm 2 first takes one scan of each bucket B (line 2 to line 3) to record the frequency f(v) of each column value v in bucket B. Then this algorithm takes one scan of each tuple t in the table T (line 4 to line 6) to find out all tuples that match B and record their matching probability p(t,B) and the distribution of the candidate sensitive values D(t,B), which are added to the list L[t] (line 6). The sliced table is ℓ-diverse iff for all the sensitive value s, $p(t, s) \leq 1/\ell$ (line 7 to line 10).

V. CONCLUSION

Data often contains personally identifiable information and therefore releasing such data may result in various privacy breaches. Several anonymization methods, like Generalization and Bucketization are designed for privacy preserving microdata publishing. Generalization loses considerable amount of information mainly for high dimensional data. Bucketization does not prevent membership disclosure and does not apply for data that do not have a clear separation between QI- attributes and SAs. In this paper we show how slicing method can be used for attribute disclosure protection. Slicing preserves better utility than generalization method and is more effective than bucketization method in workloads involving the sensitive attribute. It also demonstrates that how overlapping slicing is used to prevent the membership disclosure.

REFERENCES

[1] P. Samarati and L. Sweeney, "Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression," 1998. Technical Report, SRI-CSL-98-04, SRI International.
[2] L. Sweeney, "k-Anonymity: A model for protecting privacy," International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, vol. 10, no. 5, pp. 557– 570, 2002.
[3] X. Xiao and Y. Tao, "Anatomy: simple and effective privacy preservation," in Proceedings of the International Conference on Very Large Data Bases (VLDB), pp. 139–150, 2006.
[4] N. Koudas, D. Srivastava, T. Yu, and Q. Zhang, "Aggregate query answering on anonymized tables," in Proceedings of the International Conference on Data Engineering (ICDE), pp. 116–125, 2007.
[5] D. Kifer and J. Gehrke, "Injecting utility into anonymized datasets," in Proceedings of the ACM SIGMOD International Conference onManagement of Data (SIGMOD), pp. 217–228, 2006.
[6] C. Aggarwal, "On k-anonymity and the curse of dimensionality," in Proceedings of the International Conference on Very Large Data Bases (VLDB), pp. 901–909, 2005.
[7] M. E. Nergiz, M. Atzori, and C. Clifton, "Hiding the presence of individuals from shared databases," in Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD), pp. 665–676, 2007.
[8] K. LeFevre, D. DeWitt, and R. Ramakrishnan, "Mondrian multidimensional k- anonymity," in Proceedings of the International Conference on Data Engineering (ICDE), p. 25, 2006.