IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

A NOVEL APPROACH FOR PRIVACY PRESERVING IN DATA PUBLISHING USING SLICING

Ms.C.Ajitha[1], Mrs. K.Kirubanantha valli[2].,
Assistant Professor , Assistant Professor
Computer Science and Engineering
Unnamalai Institute of Technology
Tamilnadu, India.

Abstract: *Several agencies, institutions, bureaus, organizations make (sensitive) data involving people publicly available. However the sensitivity of the real world datasets needs more data privacy, confidentiality, authenticity. It is obvious that the user's data privacy is being violated by the intruders. So, in order to secure the user's sensitive data during releases from the intruders, a technique called Slicing is proposed. Several anonymization techniques, such as generalization and bucketization, have been designed for privacy preserving micro data publishing. We show that slicing preserves better data utility than generalization and can be used for membership disclosure protection and it can also handle high-dimensional data. Our workload experiments confirm that slicing preserves better utility than generalization and is more effective than bucketization in workloads involving the sensitive attribute.*

Keywords*- Privacy-preserving release of transaction data, anonymity, sequential background knowledge*

I. INTRODUCTION

Government agencies and other organizations frequently need to publish micro data, e.g., medical data or census data, for research and other purposes. normally, such data are stored in a table, and each record (row) corresponds to one individual. Each record has a number of attributes, which can be divided into the following three categories: 1) Attributes that clearly identify individuals. These are known as explicit identifiers and include, e.g., Social Security Number. 2) Attributes whose values when taken together, can potentially identify an individual. These are known as quasi-identifiers, and may include, e.g., Zip code, Birth-date, and Gender. 3) Attributes that are considered sensitive, such as Disease and Salary.

When releasing micro data, it is necessary to prevent the sensitive information of the individuals from being disclosed. Two types of information disclosure have been used in this system. Identity disclosure and attribute disclosure. Identity disclosure occurs while an individual is linked to a particular record in the released table. Attribute disclosure occurs while new information about some individuals is revealed, i.e., the released data make it possible to deduce the characteristics of an individual more accurately than it would be possible before the data release.

While the released table gives useful information to researchers, it presents disclosure risk to the individuals whose data are in the table. Therefore, our objective is to limit the disclosure risk to an acceptable level while maximizing the benefit. This is achieved by anonymzing the data before release. The first step of anonymization is to remove explicit identifiers. However, this is not enough, as an adversary may already know the quasi-identifier values of some individuals in the table. This knowledge can be either from personal knowledge (e.g., knowing a particular individual in person), or from other publicly available databases (e.g., a voter registration list) that include both explicit identifiers and quasi-identifiers. A common anonymization approach is generalization, which replaces quasi identifier values with values that are less-specific but semantically consistent. As a result, more records will have the same set of quasi-identifier values. We define an equivalence class of an anonymized table to be a set of records that have the same values for the quasi-identifiers. To efficiently limit disclosure, we need to measure the disclosure risk of an anonymized table.

II. RELATED WORKS

C. Dwork [5] proposed a concrete interactive privacy mechanism achieving differential privacy. To achieve differential privacy by the addition of random noise whose magnitude is chosen as a function of the largest change a single participant could have on the output to the query function.

K. LeFevre, D. DeWitt, and R. Ramakrishnan [1], introduced a multidimensional recoding model for k-anonymity. Although optimal multidimensional partitioning is NP-hard, here we provide a simple and efficient greedy approximation algorithm for several general purpose quality metrics.

T. Li, N. Li, and J. Zhang [3] proposed a general framework for modeling the adversary's background knowledge using kernel estimation methods. This framework subsumes different types of knowledge (e.g., negative association rules) that can be mined from the data. Under this framework, we reason about privacy using Bayesian inference techniques and propose the skyline (B, t)-

privacy model, which allows the data publisher to enforce privacy requirements to protect the data against adversaries with different levels of background knowledge.

D.J. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J.Y.Halpern, [4] proposed a polynomial time algorithm to measure the amount of disclosure of sensitive information in the worst case, given that the attacker has atmost k pieces of information in this language. We also provide a method to efficiently sanitize the data so that the amount of disclosure in the worst case is less than a specified threshold.

J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A.W.-C. Fu [6] study the problem of utility-based anonymization. First, we propose a simple framework to specify utility of attributes. The framework covers both numeric and categorical data. Second, we develop two simple yet efficient heuristic local recoding methods for utility-based anonymization. Our extensive performance study using both real data sets and synthetic data sets shows that our methods outperform the state-of-the-art multidimensional global recoding methods in both discernability and query answering accuracy.

T. Li and N. Li [2] proposed the Injector framework for data anonymization. Injector first mines negative association rules from the data using data mining techniques and then uses them in the data anonymization process. Also develop an efficient anonymization algorithm to incorporate these negative association rules.

III. SLICING ALGORITHM

The Slicing algorithm proposed here is to achieve l-diverse slicing. Our algorithm consists of three phases: attribute partitioning, column generalization, and tuple partitioning.

3.1 Attribute Partitioning

Here we partition the attributes, so that highly correlated attributes are in the same column. This is good for both utility and privacy. In terms of data utility, group ing highly-correlated attributes 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 association of uncorrelated attribute values is much less frequent and thus more identifiable. Therefore, it is better to break the associations between uncorrelated attributes, in order to protect privacy. In this phase, we first compute the correlations between pairs of attributes and then cluster attributes based on their correlations.

3.2 Column Generalization

Initially column generalization may be required for identity/membership disclosure protection. If a column value is unique in a column (i.e.,the column value appears only once in the column), a tuple with this unique column value can only have one matching bucket. Second, when column generalization is applied, to achieve the same level of privacy against attribute disclosure, bucket sizes can be smaller. While column generalization may result in information loss, smaller bucket-sizes allows better data utility.

3.3 Tuple Partitioning

In the tuple partitioning phase, tuples are partit ioned into buckets. The tuple-partition 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 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.

The objective of the tuple-partition algorithm is to check whether a sliced table satisfies ℓ-diversity.The diversity check algorithm maintains a list of statistics L[t] about t's matching buckets. Each element in the list L[t] contains statistics about one matching bucket B: the matching probability p(t,B) and the distribution of candidate sensitive values D(t,B).

The algorithm first takes one scan of each bucket B 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 candidate sensitive values D(t,B), which are added to the list L[t].Then 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, s) values based on the law of total probability. The sliced table is ℓ-diverse if for all sensitive value s, $p(t, s) \leq 1/\ell$.

```
Algorithm tuple-partition(T, ℓ)
1. Q = {T}; SB = Φ.
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 U {B1,B2} U SB, ℓ)
6. Q = Q U {B1,B2}.
7. else SB = SB U {B}.
8. return SB.
```

```
Algorithm diversity-check (T,T*, ℓ)
1. for each tuple t ϵ T, L[t] = Φ.
2. for each bucket B in T*
3. record f (v) for each column value v in bucket B.
4. for each tuple t ϵ T
5. calculate p(t,B) and find D(t,B).
6. L[t] = L[t] U {{p(t,B),D(t,B)}}.
7. for each tuple t ϵT
8. Calculate p(t, s) for each s based on L[t].
9. if p(t, s) ≥ 1/ℓ, return false.
10. return true.
```

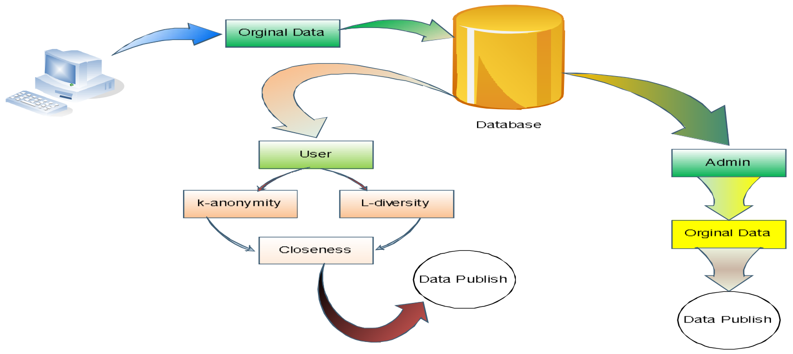

Fig.1 Proposed Architecture

IV. MEMBERSHIP DISCLOSURE PROTECTION

Our Slicing algorithm protects against membership disclosure. Let D be the set of tuples in the original data and let D be the set of tuples that are not in the original data. Let D^s be the sliced data. Given Ds and a tuple t, the goal of membership disclosure is to determine whether t ε D or t ε D. In order to distinguish tuples in D from tuples in D, we examine their differences. If t ε D, t must have at least one matching buckets in D^s. To protect membership information, we must ensure that at least some tuples in D should also have matching buckets. Otherwise, the adversary can differentiate between t ε D and t ε D by examining the number of matching buckets. We call a tuple an original tuple if it is in D. We call a tuple a fake tuple if it is in D and it matches at least one bucket in the sliced data. Therefore, we have considered two measures for membership disclosure protection.

The first measure is the number of fake tuples. When the number of fake tuples is 0 (as in bucketization), the membership information of every tuple can be determined. The second measure is to consider the number of matching buckets for original tuples and that for fake tuples. If they are similar enough, membership information is protected because the adversary cannot distinguish original tuples from fake tuples.

Slicing is an effective technique for membership disclosure protection. A sliced bucket of size k can potentially match k^c tuples. Besides the original k tuples, this bucket can introduce as many as $k^c - k$ tuples in D, which is $k^{c-1} - 1$ times more than the number of original tuples. The existence of such tuples in D hides the membership information of tuples in D, because when the adversary finds a matching bucket, she or he is not certain whether this tuple is in D or not since a large number of tuples in D have matching buckets as well. In our experiments, we empirically evaluate slicing in membership disclosure protection.

V. l -DIVERSE SLICING

Definition (l-diverse slicing). A tuple t satisfies l-diversity if for any sensitive values $p(t,s) \leq 1/l$ A sliced table satisfies 'diversity iff every tuple in it satisfies l-diversity. Our analysis show that from an l-diverse sliced table, an adversary cannot correctly learn the sensitive value of any individual with a probability greater than $1/l$. Note that once we have computed the probability that a tuple takes a sensitive value, we can also use slicing for other privacy measures such as t-closeness

VI. CONCLUSION

While k-anonymity protects not in favor of identity disclosure, it does not provide sufficient protection against attribute disclosure. The notion of 'l-diversity attempts to solve this problem. We have shown that 'l-diversity has a number of limitations and especially presented two attacks on 'l-diversity., we have proposed a novel privacy notion called "closeness." We propose two instantiations: a base model called t-closeness and a more flexible privacy model called closeness. We explain the rationale of the closeness model and show that it achieves a better balance between privacy and utility. However, in t-closeness, removing an outlier may smooth a distribution and bring it closer to the overall distribution. Another possible technique is to generalize a sensitive attribute value, rather than hiding it completely.

VII. REFERENCES

[1] K. LeFevre, D. DeWitt, and R. Ramakrishnan, "Mondrian Multidimensional k-Anonymity," Proc. Int'l Conf. Data Eng. (ICDE), p. 25, 2006.
[2] T. Li and N. Li, "Injector: Mining Background Knowledge for Data Anonymization," Proc. Int'l Conf. Data Eng. (ICDE), 2008.
[3] 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.
[4] 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
[5]C. Dwork, "Differential Privacy," Proc. 33rd Int'l Colloquiumon Automata, Languages and Programming (ICALP '06), pp. 1-12,2006.
[6] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A.W.-C. Fu, "Utility-Based Anonymization Using Local Recoding," Proc. 12th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 785-790, 2006