Anonymization Techniques for Privacy Preservation of published Data

Vidya Ingle

Abstract— **In today's world data is collected for various purposes. The collected data includes personal details or some other confidential information. Database is a collection of data that can be accessed, updated and it enables user to retrieve data. Providing security to these databases is big issue now days. Today online information is stored for almost everything like-Online admission, online banking, online shopping, E-Health service and social networking etc. This information can be used by insurance companies, drug manufacturing companies and various other marketing agencies without consent of the data provider. To maintain confidentiality, unauthorized third parties must be prevented from accessing and viewing data. It is also essential to maintain database integrity while data is transferring from source to destination. Suppose the data owner wants to share the data with researchers or analysts, how can a data owner technically insure that the individuals who are the subjects of the data cannot be re- identified while the data remain practically useful?**
Various anonymization techniques, like suppression, generalization and bucketization are designed for preserving privacy of published data. Generalization loses significant amount of information especially for large volume of data and bucketization is not sufficient prevent membership disclosure and cannot be used for data where quasi-identifying attributes and sensitive attributes cannot be clearly separated. . Slicing is new privacy preserving technique which provides improved data utility than generalization and is more efficient in workload pertaining to the sensitive attribute.

Keywords— **Anonymization Techniques, Privacy Preservation, NER, PubMed, Biomedical texts**

I. INTRODUCTION

The privacy-preserving is the process of protecting distributed information from the attackers. A number of privacy notions for protecting data publishing have been proposed.
There are three types of attributes in an original microdata table: Identifiers 1. Attributes that are uniquely identified e.g. Social Security Number;
2. Quasi-identifiers (QI): attributes which the adversary may already know and which, when taken together, can identify an individual e.g., Birth date, Sex, and Zip code; and 3. Sensitive attributes (SAs): some attributes are unknown to the adversary and are considered sensitive e.g. Disease and Salary [1].
Data anonymization is a process of removing personal identifiers to protect private information.

Manuscript received 20 July, 2015
Vidya Ingle, Department of Information Technology, Mumbai University/ Pillai's Institute of information Technology, New Panvel/ Navi Mumbai,India, (e-mail: inglevidya@rediffmail.com).

Data anonymization enables transmit information between two organizations by converting text data into non human readable form [1].This technique protects privacy of original data by transformation. There are several techniques are present for privacy preserving data publishing named as K-anonymity, perturbation, swapping etc. Basically these techniques are used to convert data in such a format that unauthorized person was not indented to trace the original data as well as associated user also can find the limited data for his use.
Data anonymization enables transferring information between two organizations by converting text data into non human readable form .This technique protects privacy of original data by modification.

I. K-ANONYMITY

When some attributes of database are suppressed or generalized such that each row is identical with at least k-1 other rows in that database is said to be K-anonymous database. K-Anonymity is useful to prevent exact database linkages when multiple version of same database is released or an individual has record in multiple public databases. K-anonymity is achieved with two techniques: generalization and suppression. The protection k-anonymity provides is easy and simple to understand. Three kinds of information disclosure have been mentioned in literature as:
I. Identity disclosure II. Attribute disclosure III. Membership disclosure
K-anonymity protects against identity disclosure but cannot provide a safety against attribute disclosure. Generalization and suppression are the most common methods used for deidentification of the data in k-anonymity based algorithms [2]. Suppression is a process of replacing occurrences of certain value with a special value "?" or "*" signifying that any value can be placed as a substitute. Suppression radically diminishes the quality of the data if not properly used. Thus most k-anonymity- related research have focused on generalization.
A new anonymization technique called slicing is used, which partitions the data both horizontally and vertically. Slicing conserve enhanced data usefulness than generalization and can be used for membership disclosure protection.
Attacks on k-anonymity

1) Homogeneity Attack: This attach happens when an attacker can reveal sensitive information on base of some known information. When there are same values for sensitive attribute in a set of *k* records, nevertheless the data has been *k*-anonymized; it is easy to identify sensitive value for particular individual [4].

Example 1 Table 1 is the original data table, and Table 2 is an anonymized version of it satisfying 3-anonymity. Here the sensitive attribute is Disease. Assume that person A knows that person B is a 28-year old man living in ZIP 41077 and B's record is in the table. From Table 2, person A can infer that person B's record is present in one of the first three records, thus must have Respiratory disease. This is the homogeneity attack.

2) Background Knowledge Attack: This attack may occur when some demographic information can be linked with released data and helps to neglect some of the sensitive values so that definite sensitive value of an attribute can be revealed [4].

For an example, suppose that, person A knows person C's age and zip code, person A can conclude that person C's record is in the last equivalence class in Table 2. Furthermore, suppose that person A knows that person C has very low risk for Respiratory disease. This background knowledge enables person A to conclude that person C most probably has cancer.

Limitations of k-anonymity are: (1) K-anonymity exposes about the presence of a given individual in the database, ie does not protect against membership disclosure

(2) Sensitive attributes are plainly published in K-anonymous database.

(3) K-anonymity suffers from background knowledge and homogeneity attack.

(5) K-anonymity is not suitable for high volumes of data with no loss of usefulness.

Table 1: Original Table

Age	Sex	Zipcode	Disease
28	M	41077	Respiratory Disease
23	F	41002	Respiratory Disease
29	F	41078	Respiratory Disease
41	F	41405	Gastritis
53	M	41409	Respiratory Disease
49	M	41406	Cancer
33	M	41005	Respiratory Disease
35	F	41073	Cancer
34	M	41007	Cancer

Table 2: 3-Anonymous version of Table 1

Age	Sex	Zipcode	Disease
2*	*	410**	Respiratory Disease
2*	*	410**	Respiratory Disease
2*	*	410**	Respiratory Disease
>=40	*	4140*	Gastritis
>=40	*	4140*	Respiratory Disease
>=40	*	4140*	Cancer
3*	*	410**	Respiratory Disease
3*	*	410**	Cancer
3*	*	410**	Cancer

II. L-DIVERSITY

A group of tuple with same quasi identifiers is said to have *l*-diversity if there are at least *l* "well-represented" values for the sensitive attribute for that group. A table is said to have *l*-diversity if every equivalence class of the table has *l*-diversity is an extension to k-anonymity and protects against homogeneity attack [4]. L-diversity is implemented by a technique called bucketization.

Bucketization: In bucketization SAs are separated from the QIs by doing the random permutation on the SA values in each bucket. Thus an anonymized data consist of a set of buckets with a permuted sensitive attribute values. Bucketization does not prevent membership disclosure because it publishes the QI values in their original forms thus it is easy to find out an individual's record in the published data. Bucketization requires the clear separation of SA and QI attributes and it breaks the attribute correlation between them. L-diversity is insufficient to provide attribute disclosure .Two attacks on l-diversity are Skewness Attack: When the overall distribution is skewed, satisfying l-diversity does not prevent attribute disclosure. Similarity Attack: When the sensitive attribute values in an equivalence class are distinct but semantically similar, an adversary can learn important information. This leakage of sensitive information occurs because while l-diversity requirement ensures "diversity" of sensitive values in each group, it does not take into account the semantically closeness of these values [3].

While bucketization has better data utility than generalization, it has several limitations. First, bucketization does not prevent membership disclosure. Because bucketization 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 an example, 87% of the individuals in the United States can be uniquely identified using only three attributes (Birth date, Sex, and Zipcode). A microdata (e.g., census data) usually contains many other attributes besides those three attributes. This means that the membership information of most individuals can be inferred from the bucketized table. Second, bucketization requires a clear separation between QIs and SAs. However, in many datasets, it is unclear which attributes are QIs and which are SAs. Third, by separating the sensitive attribute from the QI attributes, bucketization breaks the attribute correlations between the QIs and the SAs [3].

III. SLICING

Slicing splits attribute uniformly horizontally and vertically. In vertical division more correlated attributed are taken into one group and uncorrelated attributed are grouped independently. Tuple are clustered forming buckets by partitioning horizontally. After grouping tuple values of column are randomly shuffled [1]. Slicing works in three main phases:

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

Functional procedure:-

Step 1: Obtain the data set from the database.

Step 2: Divide the records into columns and buckets.

Step 3: Swap the sensitive values.

Step 4: Multi set values generated and displayed.

Step 5: Dataset attributes are combined and safe and sound data presented.
When new record is inserted to sliced database tuple is checked if it violets the privacy of database.There is check for l-diversity to protect against homogeneity and background knowledge attack .

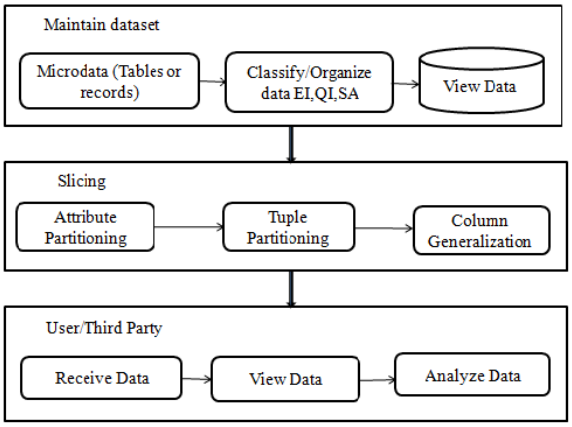

Fig 2: Architecture of the slicing

Architecture of the slicing as shown in figure 2. In first phase maintain dataset in the form of tables and records. Attributes are classified into three main categories i.e. Explicit identifier, quasi identifier and sensitive attributes. In second step slicing is performed on the data set i.e. vertical and horizontal portioning is performed on the dataset.

Table 3:Original Table with diverse attribute values

Age	Sex	Zipcode	Disease
21	M	41006	dyspepsia
21	F	41006	flu
36	F	41005	flu
55	F	41005	bronchitis
56	M	41402	flu
60	M	41402	dyspepsia
60	M	41404	Dyspepsia
66	F	41404	gastritis

Table 4: Generalized Table

Age	Sex	Zipcode	Disease
[22-52]	*	4100*	dyspepsia
[22-52]	*	4100*	flu
[22-52]	*	4100*	flu
[22-52]	*	4100*	bronchitis
[56-66]	*	4140*	flu
[56-66]	*	4140*	dyspepsia
[56-66]	*	4140*	Dyspepsia
[56-66]	*	4140*	gastritis

Table 5:Bucketised table

Age	Sex	Zipcode	Disease
21	M	41006	flu
21	F	41006	dyspepsia
36	F	41005	bronchitis
55	F	41005	flu
56	M	41402	gastritis
60	M	41402	flu
60	M	41404	dyspepsia
66	F	41404	dyspepsia

Table 6:Sliced Table

(Age,Sex)	(Zipcode, Disease)
(21,M)	(41005,flu)
(21,F)	(41006,dyspepsia)
(36,F)	(41005,bronchitis)
(55,F)	(41006,flu)
(56,M)	(41404,gastritis)
(60,M)	(41402,flu)
(60,M)	(41402,dyspepsia)
(66,F)	(41404,dyspepsia)

Comparison with Generalization [1]: Slicing groups correlated attributes in one column and maintain their association. We can see original table and generalized table shown in Table 3 and Table 4. When the values are generalized data usefulness is vanished. But in the sliced table shown in Table 6, correlations between Age and Sex and correlations between Zipcode and Disease are preserved. As slicing groups correlated attributes into one column, slicing decreases the dimensionality of the data. Every column of the table can be seen as a sub-table with a lesser dimensionality [1].
Comparison with Bucketization [1]: To compare slicing with bucketization, remember that bucketization can be viewed as a special case of slicing, where there are exactly two columns: one column contains only the SA, and the other contains all the QIs.Table 5 is bucketized version of Table 3. The advantages of slicing over bucketization can be understood as follows. First, by partitioning attributes into more than two columns, slicing can be used to prevent membership disclosure. Second, unlike bucketization, which requires a clear separation of QI attributes and the sensitive attribute, slicing can be used without such a separation. For dataset such as the census data, one often cannot clearly separate QIs from SAs because there is no single external public database that one can use to determine which attributes the adversary already knows. Slicing can be useful for such data. Finally, by allowing a column to contain both some QI attributes and the sensitive attribute, attribute correlations between the sensitive attribute and the QI attributes are preserved. For example, in Table 6, Zipcode and Disease form one column, enabling inferences about their correlations.
Attribute Disclosure Protection: Let us see how slicing can be used to prevent attribute disclosure, based on the privacy requirement of ℓdiversity and introduce the notion of ℓ-diverse slicing. Take an example illustrating how slicing satisfies ℓ-diversity where the sensitive attribute is

"Disease".The sliced table shown in Table 6 satisfies 2-diversity. Consider tuple t1 with QI values (21, M, 41006). In order to determine t1's sensitive value, one has to examine t1's matching buckets. By examining the first column (Age, Sex) in Table 6, we know that t1 must be in the first bucket B1 because there are no matches of (21,M) in bucket B2. Therefore, one can conclude that t1 cannot be in bucket B2 and t1 must be in bucket B1. Then, by examining the Zipcode attribute of the second column (Zipcode,Disease) in bucket B1, we know that the column value for t1 must be either (41006,dyspepsia) or (41006,flu) because they are the only values that match t1's zipcode 41006. Note that the other two column values have zipcode 41005. Without additional knowledge, both dyspepsia and flu are equally possible to be the sensitive value of t1. Therefore, the probability of learning the correct sensitive value of t1 is bounded by 0.5. Similarly, we can verify that 2-diversity is satisfied for all other tuples in Table 6 [1].

Membership Disclosure Protection: Bucketization releases the QI values in their original form. We know that most persons can be distinctively identified with the QI values; the presence of an individual in the original data can be confirmed by examining the occurrence of QI values in bucketized data. If QI values does not exist means individual is not in data but if QI values occurs at least once it can be assumed that individual is present in data [1]. Thus to mystify an adversary the occurrence of original tuple must be same as number of occurrences of fake tuples in bucketized data.

Assume D is a set of tuples in the original data and let D_n be the set of tuples that are not in the original data. Let D_s be the sliced data. If we have D_s and a tuple t, membership disclosure can be established by knowing if $t \in D$ or $t \in D_n$. If $t \in D$ there should be at least one matching buckets in D_s. To protect membership information, we must ensure that at least some tuple in D_n should also have matching buckets. Otherwise, the adversaries can differentiate between $t \in D$ and $t \in D_n$ 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_n 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 [1].

IV. CONCLUSION

Anonymization technique is powerful method for privacy preserving of published data. This paper focuses on new anonymization method that is slicing. Slicing overcomes the limitations of generalization and bucketization and preserves better utility while protecting against privacy threats. Slicing preserves better data utility than generalization and is more effective than bucketization in workloads involving the sensitive attribute.

.

REFERENCES

[1] Tiancheng Li,Ninghuli Li,Jian Zhang,Ian Molloy, Slicing: A new approach for privacy preserving data publishing,IEEE transactions on knowledge and data engineering,vol 24,no 3,march 2012.

[2] Alberto Trombetta, Wei Jiang, Elisa Bertino and Lorenzo Bossi , Privacy –Preserving Updates to Anonymous and Confidential Databases, IEEE, 2011.

[3] Ninghui Li, Tiancheng Li,Suresh Venkatasubramanian,t-Closeness: Privacy Beyond k-Anonymity and _-Diversity, IEEE 23rd International Conference on Data Engineering, 2007

[4] Ashwin Machanavajjhala Johannes Gehrke Daniel Kifer Muthuramakrishnan Venkitasubramaniam, ℓ-Diversity: Privacy beyond k-Anonymity, 22'nd Int'l Conf.Data Engineering, 2006.

[5] A. Trombetta, E. Bertino. Private updates to anonymous databases. In Proc. Int'l Conf. on Data Engineering (ICDE), Georgia, US, 2006.

[6] L. Sweeney. K-anonymity: a model for protecting privacy. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10(5), 557–570, 2002.

Vidya Ingle P.G. scholar at Pillai's Institute of Information Tecnology (PIIT), New Panvel,Mumbai University.