Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 5, May 2013, pg.396 – 401

RESEARCH ARTICLE

A STUDY OF PRIVACY-PRESERVING WHILE SHARING SENSITIVE INFORMATION

K.V. Lakshmi Praveena[1]

[1]Student of M. Tech II year, CSE, NBKRIST, Vidyanagar, SPSR Nellore District, India

Abstract— Modern society is increasingly dependent on (and fearful of) massive amounts and availability of electronic information. There are numerous everyday scenarios where sensitive data must be sometimes reluctantly or suspiciously shared between entities without mutual trust. This prompts the need for mechanisms to enable limited (privacy-preserving) information sharing. A typical scenario involves two parties: one seeks information from the other that is either motivated, or compelled to share only the requested information. This paper highlights two main technical challenges:

(1) How to enable this type of sharing such that parties learn no information beyond what they are entitled to, and

(2) How to do so efficiently, in real-world practical terms.

In this paper, it is shown that slicing preserves better data utility than generalization and can be used for membership disclosure protection. Another important advantage of slicing is that it can handle high-dimensional data. In this paper, it is shown how slicing can be used for attribute disclosure protection, Membership disclosure protection and develop an efficient algorithm for computing the sliced data that obey the L-diversity requirement.

Key Terms: - Privacy preservation; Micro data; sensitive information; data anonymization; Bucketization; Generalization; Slicing

I. INTRODUCTION

Data mining on databases is quite useful. However, sharing data related to individuals to public may compromise individual privacy. For example, a hospital may release patient diagnosis records so that data analysts and researchers can study the characteristics of various diseases. The raw data, also called *micro data* contain the identities of individuals such as names or keys which should not be released in order to protect individual privacy. However, if an adversary has access to the publicly available voter registration list1, s/he can discover a large portion of patients' identities by joining the published table and the voter registration list on some attributes such as Age, Sex and Zipcode, which are called *quasi-identifier attributes* (*QI*). Some Attributes are sensitive attributes (SAs), which are unknown to the adversary and are considered sensitive, like Disease and Salary.

Sharing Sensitive Information with Privacy

The notion of privacy is commonly described as the ability of an individual or a group to seclude information about them, and thereby reveal it selectively. In many nations, laws or constitutions protect privacy as a fundamental individual right. The availability of information about an individual may result in having power over that individual, hence, generating concerns on potential misuse by governments, corporations, or other individuals. In recent years, advances in computer and communication technologies have significantly amplified privacy risks. Nowadays, data is routinely exchanged electronically and collected by third parties. Privacy concerns are no longer limited to the anonymity and untraceability of digital activities. The disclosure of private

information yields an increasing number of legal, monetary, practical, or even emotional, privacy issues. However, the need for controlled (privacy-preserving) sharing of sensitive information occurs in many realistic scenarios, ranging from national security to individual privacy protection.

Three types of information disclosures have been identified namely the identity disclosure, attribute disclosure and membership disclosure. When an individual is linked to a particular record in the released table then identity disclosure exists. Attribute disclosure occurs when a new information about some individual is revealed (the released data makes it possible to infer the characteristics of an individual more accurately than it would be possible before the data release).Identity disclosure leads to attribute disclosure. If there is identity disclosure an individual is re-identified and the corresponding sensitive values are revealed. It is recognized that even disclosure of false attribute information may cause harm. An observer may incorrectly perceive an individual's sensitive attribute from the released table. The observer takes a particular value and behaves accordingly based on the perception. This can harm the individual attribute even if the perception is incorrect.

Examples:

Aviation Safety: The U.S. Department of Homeland Security (DHS) needs to check whether any passengers on any flight to/from the United States must be denied boarding or disembarkation, based on several secret lists, e.g., the Terror Watch List (TWL). Today, airlines submit their passenger manifests to the DHS, along with a large amount of sensitive information, including credit card numbers. Besides its obvious privacy implications, this modus operandi poses liability issues with regard to mostly innocent passengers' data Ideally, the DHS would obtain information pertaining only to passengers on one of its watch-lists, without disclosing any information to the airlines.

Healthcare: A health insurance company needs to retrieve information about a client from other insurance carriers or hospitals. Clearly, the latter cannot provide any information on other patients while the former cannot disclose the identity of the target client.

These examples motivate the need for privacy-preserving sharing of sensitive information and pose two main technical challenges: (1) how to enable this type of sharing such that parties learn no information beyond what they are entitled to, and (2) how to do so efficiently, in real-world practical terms.

II. PROBLEM DEFINITION

Generally in privacy preservation there is a loss of security. Privacy protection is impossible due to the presence of Adversary's background knowledge in real life application .The current practice in data sharing relies mainly on policies and guidelines as to what types of data can be shared and on agreements on the use of shared data. The approach alone may lead to excessive data distortion or insufficient protection. Privacy-preserving data sharing provides methods and tools for sharing useful information while preserving data privacy. Many algorithms like generalization and bucketization have tried to preserve privacy however they exhibit attribute disclosure. So to overcome this problem an algorithm called Slicing is used.

III. EXISTING SYSTEM

Several anonymization techniques, such as generalization and bucketization, have been designed for privacy preserving micro data publishing. Recent work has shown that generalization loses considerable amount of information, especially for high dimensional data. Bucketization, on the other hand, 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.

IV. PROPOSED SYSTEM

In this paper, a new data anonymization technique called slicing introduced. Slicing partitions the data set both vertically and horizontally. Slicing vertically partition the attributes by grouping attributes into columns based on the correlations among the attributes. Each attribute column contains a subset of attributes that are highly correlated. Horizontal partition is achieved by grouping tuples into buckets. Attributes finally within each bucket, values in each column are randomly permutated (or sorted) to break the linking between different columns.

Objective

The basic idea of slicing is to break the association cross columns but preserve the association within each column. So that dimensionality of the data is reduced and preserves better utility than generalization and bucketization. Slicing groups highly correlated attributes together to preserve the utility and to preserve the correlations between such attributes. Slicing breaks the associations between ncorrelated attributes to protect privacy.

Slicing implementation

Table 1 shows an example microdata table and its anonymized versions using various anonymization techniques. The original table is shown in Table 1(a). The three QI attributes are {Age, Sex, Zip code}, and the sensitive attribute SA is Disease. A generalized table that satisfies 3-anonymity is shown in Table 1(b), a bucketized table that satisfies 3- diversity is shown in Table 1(c), and sliced table is shown in Table 1(d). First the attributes are partitioned into columns. The column contains subset of attributes to vertically partition the table. Example, the sliced table in Table 1(d) contains 2 columns: the first column contains {Age, Sex} and the second column contains {Zipcode, Disease}. Slicing partition the tuples into buckets. Each bucket contains a subset of tuples to horizontally partition the table. Example, sliced table in Table 1(d) contain 2 buckets, each containing 3 tuples. Within each bucket values in each column are randomly permutated to break the linking between different columns. Example in the first bucket of the sliced table shown in Table 1(d), the values {(25, M), (32, F),(40, F)} are randomly permutated and the values {(600016, ulcer), (6000116, cholera), (47905, cancer)} are randomly permutated so that the linking between the two columns within one bucket is hidden.

Table 1

a) Original Table

Age	Sex	Zip code	Disease
25	M	600016	Ulcer
32	F	600016	cholera
40	F	600017	Cancer
49	M	600108	cholera
57	M	600108	Flu
64	F	600093	Cancer

b) Generalized Table

Age	Sex	Zip code	Disease
[25-40]	*	60001*	ulcer
[25-40]	*	60001*	Cholera
[25-40]	*	60001*	Cancer
[45-64]	*	60010*	cholera
[45-64]	*	60010*	flu
[45-64]	*	60009*	cancer

c) Bucketized Table

Age	Sex	Zip code	Disease
25	M	600016	Ulcer
32	F	600016	cholera
40	F	600017	Cancer
49	M	600108	cholera
57	M	600108	Flu
64	F	600093	Cancer

d) Sliced Table

(Age, Sex)	(zip Code, Disease)
(22,M)	(600016,cholera)
(32,F)	(600016,cancer)
(40,F)	(600017,ulcer)
(57,M)	(600093,cancer)
(64,M)	(600108,flu)
(69,M)	(600108,cholera)

Attribute Disclosure Protection

Based on the privacy requirement of ℓ-diversity slicing prevent attribute disclosure. An example is given illustrating how slicing satisfies ℓ-diversity where the sensitive attribute is "Disease". The sliced table shown in Table 1(d) satisfies 2-diversity.Consider tuple t1 with QI values (22, M, 60016). 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 1(d), it is known that t1 must be in the first bucket B1 because there are no matches of (22, M) in bucket B2. Therefore, one can conclude that t1 cannot be in bucket B2 and t1must be in bucket B1.Then, by examining the Zip code attribute of the second column (Zip code, Disease) in bucket B1. It is also known that the column value for t1 must be either (600016, cancer) or (600016, cholera) because they are the only values that match t1's zip code 600017. Note that the other two column values have zip code 600016. 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 1(d).

The following algorithm 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. In the starting Q contains only one bucket which includes all tuples and SB is empty (line 1).

Algorithm for 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} ∪ B, ℓ)
6. Q = Q ∪ {B1,B2}.
7. else SB = SB ∪ {B}.
8. return SB.

In each iteration (lines 2 to7), 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 (for more splits, line 6). Otherwise, it cannot split the bucket anymore and the algorithm puts the bucket into SB (line 7).When Q becomes empty, a sliced table is computed. The set of sliced buckets is SB (line 8). The main part of the tuple-partition algorithm is to check whether a sliced table satisfies l-diversity (line 5). Fig. 2 gives a description of the 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 (lines 2 to 3) 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 (lines 4 to 6) 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] (line 6). At the end of line 6, it Obtains 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) .

$$P(t,s)=\Sigma\, e\in L\,[t]e\,.p\,(t\,,\,B)\ast e\,.D\,(t\,,B\,)[S\,]$$

Algorithm 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] ∪ {hp(t,B),D(t,B)i}.
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.

l-Diverse Slicing

In the above example, tuple t1 has only one matching bucket. In general, a tuple t can have multiple matching buckets. It is now extended the above analysis to the general case and introduce the notion of l-diverse slicing. Consider an adversary who knows all the QI values of t and attempts to infer t's sensitive value from the sliced table. She or he first needs to determine which buckets t may reside in, i.e., the set of matching buckets of t. Tuple t can be in any one of its matching buckets. Let p(t,B) be the probability that t is in bucket B . For example, in the above example, p(t1,B1)=1 and p(t1,B1)=0. In the second step, the adversary computes p(t,s), the probability that t takes a sensitive value s. The probability for p(t,s) is calculated using the law of total probability. Let p(s|t,B) be the probability that t takes sensitive value s given that t is in bucket B according to the law of total probability, the probability p(t,s) is

$$P(t, s) = \sum_{B} P(t, B) P(s|t, B)$$

MEMBERSHIP DISCLOSURE PROTECTION

Let us first examine how an adversary can infer membership information from bucketization. Because bucketization releases the QI values in their original form and most individuals can be uniquely identified using the QI values, the adversary can simply determine the membership of an individual in the original data by examining the frequency of the QI values in the bucketized data. Specifically, if the frequency is 0, the adversary knows for sure that the individual is not in the data. If the frequency is greater than 0, the adversary knows with high confidence that the individual is in the data, because this matching tuple must belong to that individual as almost no other individual has the same QI values.The above reasoning suggests that in order to protect membership information, it is required that, in the anonymized data, a tuple in the original data should have a similar frequency as a tuple that is not in the original data. Otherwise, by examining their frequencies in the anonymized data, the adversary can differentiate tuples in the original data from tuples not in the original data. It is shown how slicing 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 Ds be the sliced data. Given Ds and a tuple t, the goal of membership disclosure is to determine whether $t \in D$ or $t \in D$. In order to distinguish tuples in D from tuples in D, their differences are examined. If $t \in D$, t must have at least one matching buckets in Ds.

To protect membership information, it must ensure that at least some tuples in D should also have matching buckets. Otherwise, the adversary can differentiate between $t \in D$ and $t \in D$ by examining the number of matching buckets. We can call a tuple an original tuple if it is in D. We can also call a tuple a fake tuple if it is in D and it matches at least one bucket in the sliced data. Therefore, two measures for membership disclosure protection have been considered. 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.

V. CONCLUSION

Slicing overcomes the limitations of generalization and bucketization and preserves better utility while protecting against privacy threats. It is illustrated how to use slicing to prevent attribute disclosure and membership disclosure. Protection against membership disclosure also helps to protect against identity disclosure and attribute 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.

REFERENCES

[1] C. Aggarwal. On k-anonymity and the curse of dimensionality. In VLDB, pages 901–909, 2005.
[2] A. Asuncion and D. Newman. UCI machine learningrepository, 2007.
[3] A. Inan, M. Kantarcioglu, and E. Bertino. Using anonymized data for classification. In ICDE, 2009.

[4] X. Xiao and Y. Tao. Anatomy: simple and effective privacy preservation. In VLDB, pages 139–150, 2006.
[5] "Slicing: A New Approach to Privacy Preserving Data Publishing" by Tiancheng Li, Ninghui Li, Jian Zhang, Ian Molloy 2009

