SCREENING: PRESERVING THE SENSITIVE INFORMATION DURING DATA MINING

S.Rakesh Subramanian
Sri Sairam Engineering College, West Tambaram, Chennai-600 005
rak.subramanian@gmail.com

Abstract— Over recent past, several techniques for preserving privacy in data mining has been devised. But each one of them added an additional overhead. Generalization has suffered from considerable loss of information whereas bucketization and randomization has suffered from the problem of membership disclosure. Eventually, Slicing approach for privacy preservation has averted the failures of its ancestors. In this paper, we introduce an extended slicing approach called screening that provides better data utility in addition to data privacy. Here, we overlap slicing that satisfies k-anonymity requirement by adding sensitive attribute to each column of slicing in order to enhance the data mining task. We show how attribute and membership disclosure protection can be implemented in this technique of privacy preservation. The implication of this paper can be seen useful when sensitive information stands the risk of getting exposed during mining. In such situations screening will reduce attribute disclosure problem such as revealing the personal information while mining the organization's management database. Thus, not only the volume of data being sent is reduced but also privacy is preserved.

Index Terms— bucketization, data mining, data privacy, data security, generalization, privacy preservation, screening, slicing.

1 INTRODUCTION

Data mining successfully extracts knowledge to support a variety of domains marketing, weather, forecasting, medical diagnosis, and national security—but it is still a challenge to mine certain kinds of data without violating the data owners' privacy. Fig. 1.a shows the actions taken in data mining process. As data mining becomes more pervasive, such concerns are increasing. Online data collection systems are an example of new applications that threaten individual privacy. Already companies are sharing data mining models to obtain a richer set of data about mutual customers and their buying habits. Therefore, there is a need for preserving privacy in mined data. The key directions in the field of privacy-preserving data mining are as follows:

Privacy-Preserving Data Publishing: These techniques tend to study different transformation methods associated with privacy. These techniques include methods such as randomization, k-anonymity, and l-diversity. Another related issue is how the perturbed data can be used in conjunction with classical data mining methods such as association rule mining. Other related problems include that of determining privacy-preserving methods to keep the underlying data useful (utility-based methods), or the problem of studying the different definitions of privacy, and how they compare in terms of effectiveness in different scenarios.

Changing the results of Data Mining Applications to preserve privacy:
In many cases, the results of data mining applications such as association rule or classification rule mining can compromise the privacy of the data. This has spawned a field of privacy in which the results of data mining algorithms such as association rule mining are modified in order to preserve the privacy of the data.

Cryptographic Methods for Distributed Privacy: In many cases, the data may be distributed across multiple sites, and the owners of the data across these different sites may wish to compute a common function. In such cases, a variety of cryptographic protocols may be used in order to

communicate among the different sites, so that secure function computation is possible without revealing sensitive information.

In both bucketization and generalization techniques, attributes are partitioned into three categories: 1) some attributes are identifiers that can uniquely identify an individual, such as Name or Social Security Number; 2) some attributes are Quasi Identifiers (QI), which the adversary may already know (possibly from other publicly available databases) and which, when taken together, can potentially identify an individual, e.g., Birthdate, Sex, and Zipcode; 3) some attributes are Sensitive Attributes (SAs), which are unknown to the adversary and are considered sensitive, such as Disease and Salary.

In both generalization and bucketization, one first removes identifiers from the data and then partitions tuples into buckets. The two techniques differ in the next step. Generalization 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 SA values in each bucket. The anonymized data consist of a set of buckets with permuted sensitive attribute values.

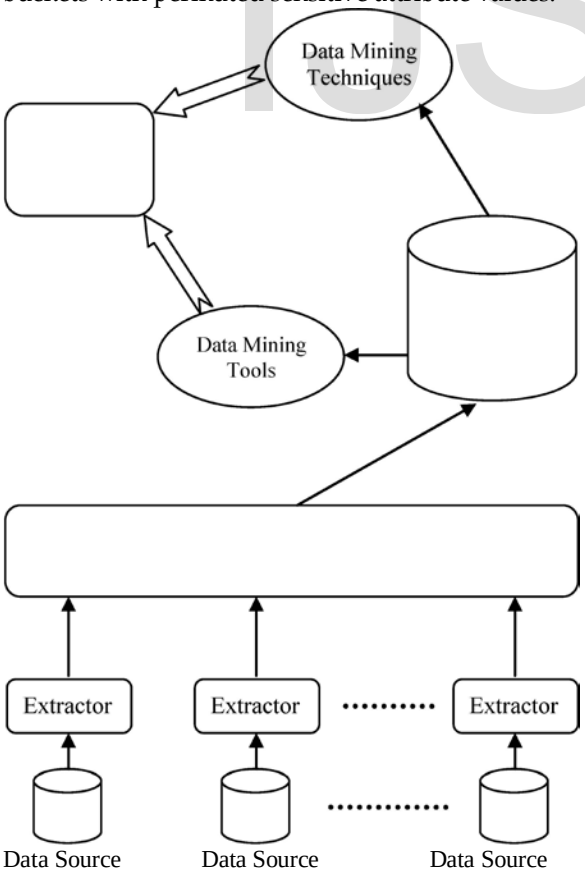

Fig.1.a Mining Process

2 Privacy Preservation Techniques

2.1 k-Anonymity

The technique k-anonymity is proposed for protecting information leakage while publishing sensitive data. If a table satisfies kanonymity for some value k, then anyone who knows only the quasi-identifier values of one individual cannot identify the record corresponding to that individual with confidence greater than 11k. While k-anonymity protects against identity disclosure, it does not provide sufficient protection against attribute disclosure.

2.2 I-Diversity

To address these limitations of k-anonymity, a stronger notion of privacy called I - diversity has introduced. An equivalence class is said to have 1-diversity if there are at least I "well-represented" values for the sensitive attribute. A table is said to have I-diversity if every equivalence class of the table has I-diversity. The term "well represented" has number of interpretations in this principle: Distinct I-diversity, Probabilistic I-diversity, Entropy I-diversity and Recursive (c,l)diversity. The following are observed from the 1- diversity approach:

- When the overall distribution is skewed, satisfying that I-diversity does not prevent attribute disclosure - known as Skewness Attack
- When the sensitive attribute values in an equivalence class are distinct but semantically similar, an adversary can learn important information- known as Similarity Attack.

3 Need For Screening

Generalization for k-anonymity losses considerable amount of information, especially for high-dimensional data. This is because of three reasons:

First, generalization for k-anonymity suffers from the curse of dimensionality. 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. However, in high dimensional data, most data points have similar distances with each other, forcing a great amount of generalization to satisfy k-anonymity even for relatively small k's. Second, in order to perform 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/set is equally possible, as no other distribution assumption can be justified. This significantly reduces the data utility of the generalized data. Third, because each attribute is generalized separately, correlations between different attributes are lost. In order to study attribute correlations on the generalized table, the data analyst has to assume that every possible combination of attribute values is equally possible. This is an inherent problem of generalization that prevents effective analysis of attribute correlations.

Eventhough bucketization has better data utility, 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. From a recent data, 87 percent of the individuals in the United States can be uniquely identified using only three attributes (Birthdate, 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 data sets, 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.

Slicing approach has partitioned data both horizontally and vertically. In slicing process, the sensitive attribute is preserved by breaking correlations between uncorrelated attributes. This has also preserved the data utility during mining process to a certain level. Hence, in this paper we introduce an approach that takes the slicing to next level by replicating the sensitive attribute in the vertically partitioned microdata.

4 SCREENING

We give an example to formulate screening. The table contains information about a group of people with their *salary* being the sensitive attribute. The microdata table is partitioned both vertically and horizontally into buckets. The Quasi identifiers are generalized. The correlation between both (*Age,Sex,Salary*) and (Zipcode,Occupation,Salary) are preserved(SA being equally distributed). The key intuition that screening provides privacy protection is that the screening process ensures that for any tuple, there are generally multiple matching buckets. Given a tuple $t(v_1,v_2\ldots,v_c)$ where c is the number of columns and v_i is the value for the ith column, a bucket is a matching bucket for t if and only if for each $i(1 \le i \le c)$, v_i appears atleast once in the i'th column of the bucket. Any bucket that contains the original tuple is a matching bucket. At the same time, a matching bucket can be due to containing other tuples each of which contains some but not all v_i's. Table1 shows the original microdata and the screened table.

We now discuss the implementation of the screening: First, the correlated attributes are grouped i.e (*Age,Sex,Salary*) and (Zipcode,Occupation,Salary). Then, we apply generalization to quasi attributes(QI) *Sex and ZipCode i.e* For Sex we define M and F as * and for ZipCode we reduce the full code by applying *. The QI *Age* is generalized by specifying the upper and lower limits of ages for each bucket. Screening then partitions attributes into columns. This vertically partitions the table. In our table, the correlated attributes comprises a column.

Screening also partition tuples into buckets. Each bucket contains a subset of tuples. This horizontally partitions the table. Within each bucket, values in each column are randomly permutated to break the linking between different columns. Our table is partitioned into two buckets with four tuples each. For example, in the first bucket { (26-32,*,10000), (26-32,*,12000), (26-32 ,*,8000), (26-32,*,20000)} are randomly permuted and {((402**,{nurse,engineer},12000), (402**,{nurse,police},8000), (403**,{police,engineer},20000), (401**,{police,engineer},10000)} are randomly permuted.

TABLE 1a
An Original Microdata table and its Screened version

Name	Sex	ZipCode	Age	Occupation	Salary
Alice	F	40178	26	nurse	10000
Betty	F	40277	30	nurse	12000
Carl	M	40276	32	police	8000
Diana	F	40175	51	cook	9000
Ella	F	40385	28	engineer	20000
Finch	M	40485	43	engineer	23000
Gavin	M	40286	50	clerk	8000
Helvin	M	40267	48	clerk	11000

(1a)

TABLE 1b
Horizontally partitioned one attribute per-column

Sex	ZipCode	Age	Occupation	Salary
F	40178	26	nurse	10000
F	40277	30	nurse	12000
M	40276	32	police	8000
F	40175	51	cook	9000
F	40385	28	engineer	20000
M	40485	43	engineer	23000
M	40286	50	clerk	8000
M	40267	48	clerk	11000

(1b)

TABLE 1c
Screening Output data

(Age,Sex,Salary)	(ZipCode,Occupation,Salary)
(26-32,*,10000)	(402**,{nurse,engineer},12000)
(26-32,*,12000)	(402**,{nurse,police},8000)
(26-32,*,8000)	(403**,{police,engineer},20000)
(26-32,*,20000)	(401**,{police,engineer},10000)
(43-51,*,9000)	(402**,{cook,clerk},11000)
(43-51,*,23000)	(404**,{clerk,engineer},23000)
(43-51,*,8000)	(402**,{clerk,engineer},8000)
(43-51,*,11000)	(401**,{engineer,cook},9000)

4.1 Formalization of Screening

Let T be the microdata table to be published. T contains d attributes: A={A_1,A_2,…,A_d} and their attribute domains are {D[A_1],D[A_2],…D[A_d]}. A tuple t ∈ T can be represented as t={t[A_1],t[A_2],..t[A_d]} where t[A_i] (1≤i≤d) is the A_i value of t.

Definition 1 (Attribute Partition and Columns). An attribute partition consists of several subsets of A, such that each attribute belongs to exactly one subset. Each subset of attributes is called a column. Specifically, let there be c columns C_1, C_2,…,C_c, then $\bigcup_{i=1}^{c} C_i = A$ and for any i≤ i_1≠ i_2≤c, C_{i1} ∩ C_{i2}=∅.

For simplicity of discussion, we consider only one sensitive attribute S. If the data contain multiple sensitive attributes, one can either consider them separately or consider their joint distribution. Exactly one of the c columns contains S. Without loss of generality, let the column that contains S be the last column C_c. This column is also called the sensitive column. All other columns C_1, C_2,…,C_{c-1} contain only QI attributes.

Definition 2 (Tuple Partition and Buckets). A tuple partition consists of several subsets of T, such that each tuple belongs to exactly one subset. Each subset of tuples is called a bucket. Specifically, let there be b buckets B1,B2,…,Bb, then and for any $1 \le i1 \ne i2 \le b$, $\bigcup_{i=1}^{c} B_i = T$ and for any i≤ i1 ≠ i2 ≤c ,Bi1 ∩ Bi2=∅.

Definition 3 (Screening). Given a microdata table T, a slicing of T is given by an attribute partition and a tuple partition.

For example, consider Table1. In Table 1b, the attribute partition is {{Age}, {Sex},{Salary}, {Zipcode}, {Occupation}} and the tuple partition is {{t_1,t_2,t_3,t_4},{t_5,t_6,t_7,t_8}}. In Table 1c, the attribute partition is {{Age, Sex}, {Zipcode, Disease}} and the tuple partition is {{t_1,t_2,t_3,t_4},{t_5,t_6,t_7,t_8}}. In Table 1c, the attributes *Sex* and *Age* are column generalized & the attribute *Occupation* is bucketized.

Definition 4 (Column Generalization). Given a microdata table T and a column Ci={Ai1,Ai2,…,Aij} where Ai1,Ai2,…Aij are attributes, a column generalization for Ci is defined as a set of non overlapping j-dimensional regions that completely cover D[Ai1]×D[Ai2]×…×D[Aij]. A column generalization maps each value of Ci to the region in which the value is contained. Column generalization ensures that one column satisfies the k-anonymity requirement. It is a multidimensional encoding and it is the main step of screening. Specifically, a general screening algorithm consists of the following three phases: attribute partition, column generalization, and tuple partition. Because each column contains much fewer attributes than the whole table, attribute partition enables slicing to handle high-dimensional data.

Definition 5 (Matching Buckets). Let be the { C_1, C_2,…,C_c } c columns of a screened table. Let t be a tuple, and t[C_i] be the C_i value of t. Let B be a bucket in the screened table, and B[C_i] be the multiset of C_i values in B. We say that B is a matching bucket of t iff for all 1 ≤i ≤c, t[C_i]∈B[C_i].

5 SCREENING ALGORITHMS

We now present an efficient slicing algorithm to achieve l-diverse screening. Given a microdata table T and three parameters c,l and k, the algorithm computes the sliced table that consists of c columns and satisfies the privacy requirement of l-diversity and k-anonymity. Our algorithm consists of three phases: attribute partitioning, column generalization, and tuple partitioning. We now
describe the three phases.

5.1 Attribute Partitioning

Screening partitions attributes so that highly correlated attributes are in the same column. This is good for both utility and 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 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.

5.2 Column Generalization

By generalizing attribute values into "less-specific but semantically consistent values," generalization offers some protection against membership disclosure. Thus column generalization is 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. 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 allow better data utility. Therefore,
there is a trade-off between column generalization and tuple partitioning. Screening resolves this trade-off by generalizing the attributes in each bucket separately. In our table, the *Age* attribute of Table1c is generalized separately for buckets B_1 and B_2.

Algorithm tuple-partition(T, ℓ)
1. $Q = \{T\}$; $SB = \emptyset$.
2. while Q is not empty
3. remove the first bucket B from Q; $Q = Q - \{B\}$.
4. split B into two buckets B_1 and B_2, as in Mondrian.
5. if **diversity-check**$(T, Q \cup \{B_1, B_2\} \cup SB, \ell)$
6. $Q = Q \cup \{B_1, B_2\}$.
7. else $SB = SB \cup \{B\}$.
8. return SB.

Fig.1 Tuple Partition Algorithm

Algorithm diversity-check(T, T^*, ℓ)
1. for each tuple $t \in T$, $L[t] = \emptyset$.
2. for each bucket B in T^*
3. record $f(v)$ for each column value v in bucket B.
4. for each tuple $t \in T$
5. calculate $p(t, B)$ and find $D(t, B)$.
6. $L[t] = L[t] \cup \{\langle p(t, B), D(t, B)\rangle\}$.
7. for each tuple $t \in T$
8. calculate $p(t, s)$ for each s based on $L[t]$.
9. if $p(t, s) \geq 1/\ell$, return false.
10. return true.

Fig.2 Diversity Check Algorithm

5.3 Tuple Partitioning

In the tuple partitioning phase, tuples are partitioned into buckets.

Fig. 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 screened buckets SB. Initially, Q contains only one bucket which includes all
tuples and SB is empty (line 1). In each iteration (lines 2 to 7), the algorithm removes a bucket from Q and splits the bucket into two buckets
If the screened 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, we cannot split the bucket anymore and the algorithm puts the bucket into SB (line 7). When Q becomes empty, we have computed the sliced table. The
set of screened buckets is SB (line 8).
The main part of the tuple-partition algorithm is to check whether a screened 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 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 (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 candidate sensitive values D(t,B) which are added to the list L[t] (line 6). At the end of line 6, 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. Specifically

$$p(t,s) = \sum_{e \in L[t]} e.p(t,B) * e.D(t,B)[s]$$

The sliced table is l-diverse iff for all sensitive value s, p(t,s)≤1/l.

6 Applications of Privacy-Preserving Data Mining

The problem of privacy-preserving data mining has numerous applications in homeland security, medical database mining, and customer transaction analysis. Some of these applications such as those involving bio-terrorism and medical database mining may intersect in scope. In this section, we will discuss a number of different applications of privacy-preserving data mining methods.

6.1 Medical Databases: The Scrub and Datafly Systems

The scrub system was designed for de-identification of clinical notes and letters which typically occurs in the form of textual data. Clinical notes and letters are typically in the form of text which contain references to patients, family members, addresses, phone numbers or providers. Traditional techniques simply use a global search and replace procedure in order to provide privacy. The Scrub system uses numerous detection algorithms which compete in parallel to determine when a block of text corresponds to a name, address or a phone number. The Scrub System uses local knowledge sources which compete with one another based on the certainty of their findings. Such a system is able to remove more than 99% of the identifying information from the data.
The Datafly System was one of the earliest practical applications of privacy-preserving transformations. This system was designed to prevent identification of the subjects of medical records which may be stored in multidimensional format. The multi-dimensional information may include directly identifying information such as the social security number, or indirectly identifying
information such as age, sex or zip-code. The system was designed in response to the concern that the process of removing only directly identifying attributes such as social security numbers was not sufficient to guarantee privacy.

6.2 Bioterrorism Applications

In typical bioterrorism applications, we would like to analyze medical data for privacy-preserving data mining purposes. Often a biological agent such as anthrax produces symptoms which are similar to other common respiratory diseases such as the cough, cold and the flu. In the absence of prior knowledge of such an attack, health care providers may diagnose a patient affected by an anthrax attack of have symptoms from one of the more common respiratory diseases. The key is to quickly identify a true anthrax attack from a normal outbreak of a common respiratory disease, In many cases, an unusual number of such cases in a given locality may indicate a bio-terrorism attack. Therefore, in order to identify such attacks it is necessary to track incidences of these common diseases as well. Therefore, the corresponding data would need to be reported to public health agencies. However, the common respiratory diseases are not reportable diseases by law. Privacy preservation technique like screening
allows only limited access to the data.

6.3 Homeland Security Applications

A number of applications for homeland security are inherently intrusive because
of the very nature of surveillance. Some examples of such applications are as follows:

CredentialValidation Problem: In this problem, we are trying to match the subject of the credential to the person presenting the credential. For example, the theft of social security numbers presents a serious threat to homeland security.

Identity Theft: A related technology is to use a more *active* approach to avoid identity theft. The *identity angel* system, crawls through cyberspace, and determines people who are at risk from identity
theft. This information can be used to notify appropriate parties.

Web Camera Surveillance: One possible method for surveillance is with the use of publicly available webcams, which can be used to detect unusual activity. The approach can be made more privacy-sensitive by extracting only facial count information from the images and using these in order to detect unusual activity.

Video-Surveillance: In the context of sharing video-surveillance data, a major threat is the use of facial recognition software, which can match the facial images in videos to the facial images in a driver license database. While a straightforward solution is to completely black out each face, the
result is of limited new, since all facial information has been wiped out. A more balanced approach is to use selective downgrading of the facial information, so that it scientifically limits the ability of facial recognition software to reliably identify faces, while maintaining facial details in images.

TheWatch List Problem: The motivation behind this problem is that the government typically has a list of known terrorists or suspected entities which it wishes to track from the population. The aim is to view
transactional data such as store purchases, hospital admissions, airplane manifests, hotel registrations or school attendance records in order to identify or track these entities. This is a difficult problem because the transactional data is private, and the privacy of subjects who do not appear
in the watch list need to be protected. Therefore, the transactional behavior of non-suspicious subjects may not be identified or revealed. Furthermore, the problem is even more difficult if we assume that the watch list cannot be revealed to the data holders

7 Conclusion

Publishing data about individuals for mining without revealing sensitive information is an important problem. Anonymization and Bucketization techniques are insufficient to protect privacy issues like Homogeneity attack, Skewness Attack etc. This paper presents a new approach called screening which preserves the privacy of sensitive data by generalizing the sliced table. By allowing a column to contain both some QI attributes and the sensitive attribute, attribute correlation between the sensitive attribute and the QI attributes are
preserved. Thus Screening overcomes the discrepancies between generalization and slicing models and acts as a shield for confidential data. Improving the data utility by providing such privacy is left for future work.

8 References

[1] Data Mining: Concepts and Techniques *Jiawei Han and Micheline Kamber*, Morgan Kaufmann

[2] Aristides Gionis and Tamir Tassa, k-Anonymization with Minimal Loss of Information, 2009

[3] X. Xiao and Y. Tao, "Anatomy: Simple and Effective Privacy Preservation," Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 139-150, 2006.

[4] C. Dwork, "Differential Privacy: A Survey of Results," Proc. Fifth Int'l Conf. Theory and Applications of Models of Computation (TAMC),

[5] pp. 1-19, 2008.

[6] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu, "Approximation Algorithms for k-Anonymity," Proc. 10th Int'l Conf. Database Theory (ICDT), 2005.

[7] N. Li, T. Li, and S. Venkatasubramanian, "t-Closeness: Privacy Beyond k-Anonymity and '-Diversity," Proc. IEEE 23rd Int'l Conf.

[8] Data Eng. (ICDE), pp. 106-115, 2007.

[9] Tiancheng Li, Ninghui Li, Senior Member, IEEE, Jian Zhang, Member, IEEE, and Ian Molloy, Slicing: A New Approach for Privacy Preserving Data Publishing,2012