Slicing Technique For Privacy Preserving Data Publishing

D. Mohanapriya [#1], Dr. T.Meyyappan M.Sc., MBA. M.Phil., Ph.d., [*2]

[#] *Department of Computer Science and Engineering, Alagappa University, Karaikudi, Tamilnadu,India.*

***Abstract*— Privacy-preserving data mining is the area of data mining that used to safeguard sensitive information from unsanctioned disclosure .The problem of privacy-preserving data mining has become more important in recent years because of the increasing ability to store personal data about users. A number of techniques such as randomization and k-anonymity ,bucketization,generlization have been proposed in recent years in order to perform privacy-preserving data mining. For high-dimension data by using generalization significant amount of information is lost according to recent works. Whereas the Bucketization technique does not forbid membership and does not applicable to the data that does not have a clear distinction between sensitive attributes and quasi-identifying attributes Thus, this paper shows a solution to preserve privacy of high dimensional data.**

Keywords - randomization,kanonymity,generlisation bucketization

.

I INTRODUCTION

Privacy-preserving publishing of micro data has been reviewed rigorously in modern years. Micro data have archives each of which contains information about an individual entity, such as a person, a household, or an organization. Multiple micro data anonymization methods have been suggested. The renowned ones are generalization for k-anonymity and bucketization for ℓ-diversity. In two approaches, attributes are divided into three categories:

(1) some attributes are identifiers that can indistinctively identify an individual, such as Name or Social Security Number;
(2) some attributes are Quasi-Identifiers (QI), which the challenger can possibly identify an individual, like Date of Birth, Sex, and Pin code
(3) some attributes are Sensitive Attributes (SAs), which are not known to the challenger and are sensitive, such as Salary as well as Disease.

In both generalization as well as bucketization, one first eliminates identifiers from the data and then divides records into buckets. The two methods vary in the following step. Generalization changes the QI-values in each bucket into less precise but constant values so that records in the same bucket cannot be differentiated by their QI values. In bucketization, one divides the SAs from the QIs by arbitrarily permuting the SA values in each bucket. The anonymized data involves of a set of buckets with permuted subtle attribute values.

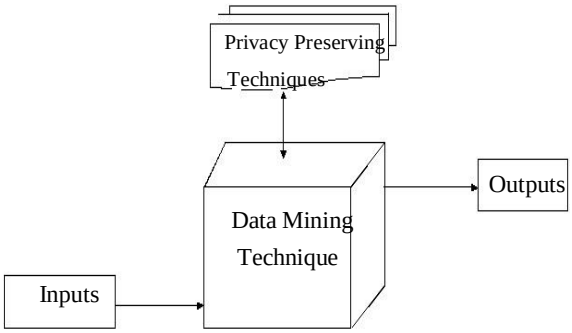

Figure 1 Architecture of Privacy Preserving in Data Mining

II EXISTING METHODS

A. Anatomy: Simple and Effective Privacy Preservation

A general method for preserving privacy is Generalization; it loses significant data in the micro data, and thus, stops efficient data analysis. This builds an anatomy, a creative method which conserves privacy as well as correlation in the micro data, and thus, incapacitates the shortcomings of generalization.Wide experiments confirm that anatomy allows researchers to deduce, from the printed tables, more precise information about the not known micro data, with a mean fault below 10%. As other vital information anatomized tables can be calculate in I/O cost undeviating to the database cardinality.

B. Methods

Various anonymization methods, like generalization and binning, have been intended for privacy preserving microdata publishing. Modern studies have indicated that generalization drops significant quantity of data, particularly for data with greater dimensions. Binning, instead, doesn't stop membership revelation and doesn't concern for information that do not have a flawless parting among quasi-identifying attributes and sensitive attributes.privacy is the claim of individuals to control when, how and to what extent information about them is communicated to others. A privacy protection principle enables users to specify the level of privacy protection against a certain type of

privacy risk. In privacy preserving , k-anonymity and l-diversity are well known principles.

1) K-anonymity

While liberating micro data for exploration tenacities, one desires to bound expose risks to a satisfactory level while exploiting data utility. To limit exposure risk, Sweeney presented the *k*-anonymityprivacy requisite, which needs each record in an anonymzed table to be inseparable with at least *k*-1 other tuples within the dataset, pertaining to a set of quasi-identifier attributes. To attain the *k*-anonymity requisite they used generalization and suppression together for data anonymization

The k-anonymity model requires that within any equivalence class of the micro-data there are at least k records. The protection k-anonymity provides is simple and easy to understand. Kanonymity[7] cannot provide a safeguard against attribute disclosure in all cases. Homogeneity attack and the Background knowledge attack are identified when using K-anonymity.

2) Attacks on k-anonmity

Homogeneity Attack:

When the non sensitive information of an individual is known to the attacker then sensitive information may be revealed based on the known information. It occurs if there is no diversity in the sensitive attributes for a particular block. This method of getting sensitive information is also known as positive disclosure. This suggests that in addition to k-anonymity, the sanitized table should also ensure "diversity" – all tuples that share the same values of their quasi-identifiers should have diverse values for their sensitive attributes.

*Background Knowledge Attack***:**

If the user has some extra demographic information which can be linked to the released data which helps in neglecting some of the sensitive attributes, then some sensitive information about an individual might be revealed. This method of revealing information is also known as negative disclosure.

To protect the identities of individuals whose records are in the data to be released, Samarati and Sweeney (1998) proposed the k-anonymity principle. A dataset satisfies k-anonymity if every individual's record is indistinguishable from at least k-i other records on quasi- identifier, i.e., attributes that can be used to link with external data, e.g., Age, Sex and Zipcode. For example, data in Table observes 4-anonymity by generalizing attributes Age and Zipcode, where records are partitioned into two indistinguishable groups. The first indistinguishable group consists of records 2, 3, 11 and 12 and the second is made of records 1 and 9 .

Table 1 original table –patient data

	Non sensitive			sensitive
	ZIP	**Sex**	**Age**	**condition**
1	13053	M	28	Heart disease
2	13068	M	29	Heart disease
3	13068	M	21	Viral infection
4	13053	F	23	Viral infection
5	14853	M	50	Cancer
6	14853	F	55	Heart disease
7	14850	M	47	Viral infection
8	14850	M	49	Viral infection
9	13053	F	31	cancer
10	13053	F	37	cancer
11	13068	M	36	cancer
12	13068	F	35	cancer

Table 2 4 ananymous patient table

	Non sensitive			sensitive
	ZIP	**Sex**	**Age**	**condition**
1	130**	*	<30	Heart disease
2	130**	*	<30	Heart disease
3	130**	*	<30	Viral infection
4	130**	*	<30	Viral infection
5	1485 *	*	>40	cancer
6	1485*	*	>40	Heart disease
7	1485*	*	>40	Viral infection
8	1485*	*	>40	Viral infection
9	130**	*	3*	Cancer
10	130**	*	3*	Cancer
11	130**	*	3*	Cancer
12	130**	*	3*	Cancer

3) L-Diversity

From the limitation of k-anonymity l-diversity can be introduced. L-diversity tries to put constraints on minimum number of distinct values seen within an equivalence class for any sensitive attribute.An equivalence class has l-diversity if there is l or more well-represented values for the sensitive attribute.

Table 3 3-diverse table

	Non-Sensitive Data			Sensitive Data
#	ZIP	Age	sex	Condition
1	1305*	<= 40	*	Heart Disease
2	1305*	<= 40	*	Viral Infection
3	1305*	<= 40	*	Cancer
4	1305*	<= 40	*	Cancer
5	1485*	>= 40	*	Cancer
6	1485*	>= 40	*	Heart Disease
7	1485*	>= 40	*	Viral Infection
8	1485*	>= 40	*	Viral Infection
9	1306*	<= 40	*	Heart Disease
10	1306*	<= 40	*	Viral Infection
11	1306*	<= 40	*	Cancer
12	1306*	<= 40	*	Cancer

4)Limitation of L-diversity

While the 'diversity principle represents an important step beyond k-anonymity in protecting against attribute disclosure, it has several shortcomings.'l - Diversity may be difficult to achieve and may not provide sufficient privacy protection.

Suppose that the original data have only one sensitive attribute: the test result for a particular virus. It takes two values: positive and negative. Further, suppose that there are 10,000 records, with 99 percent of them being negative, and only 1 percent being positive. Then, the two values have very different degrees of sensitivity. One would not mind being known to be tested negative, because then one is the same as 99 percent of the population, but one would not want to be known/considered to be tested positive. In this case, 2-diversity does not provide sufficient privacy protection for an equivalence class that contains only records that are negative.

5) Attacks on l-diversity

Skewness attack:

When the overall distribution is skewed, satisfying that diversity does not prevent attribute disclosure. Suppose that one equivalence class has an equal number of positive records and negative records. It satisfies distinct 2-diversity, entropy 2-diversity, and any recursive (c,2)-diversity requirement that can be imposed.

C.Anonymization Techniques

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 consists of a set of buckets with permuted sensitive attribute values.

1) Generalization

Generalization replaces a value with a "less-specific but semantically consistent" value.Three types of encoding schemes have been proposed for generalization:

- Global Recording,
- Regional Recording
- Local Recording.

Global recoding has the property that multiple occurrences of the same value are always replaced by the same generalized value. Regional record is also called multi-dimensional recoding (the Mondrian algorithm) which partitions the domain space into non- intersect regions and data points in the same region are represented by the region theyare in. Local recoding does not have the above constraints and allows different occurrences of the same value to be generalized differently.

Generalization consists of substituting attribute values with semantically consistent but less precise values. For example, the month of birth can be replaced by the year of birth which occurs in more records, so that the identification of a specific ndividual is more difficult. Generalization maintains the correctness of the data at the record level but results in less specific information that may affect the accuracy of machine learning algorithms applied on the k-anonymous dataset.

Comparison with Generalization

There are multiple kinds of recodings for generalization. Local recoding conserves the most information. Local recoding firstly clusters records into buckets and for every individual bucket, one changes all values of one attribute with a generalized value. This recoding is considered local since the same attribute value might be indiscriminate contrarily when they emerge in different buckets.

Drawback

(1) It fails on high-dimensional data due to the curse of dimensionality
(2) It causes too much information loss due to the uniform-distribution assumption

2) Bucketization

Bucketization[14,15] 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 permuted sensitive attribute values. In particular, bucketization has been used for anonymizing high dimensional data. However, their approach assumes a clear separation between QIs and SAs.

Comparison with Bucketization:

To contrast slicing with bucketization, we should consider that bucketization can be regarded as a particular case of slicing, where there are precisely two columns: one column consists of only the SA, and another consists of all the QIs. The benefits of slicing over bucketization can be implied as follows. Mainly, by dividing attributes into multiple columns, slicing can be used to avoid membership exposure.

Table 4 - original table

Age	Sex	Zipcode	Disease
21	M	46805	Sinus
21	F	46805	Cancer
32	F	46804	Bronchitis
51	F	46804	Sinus
53	M	46201	Gastritis
59	M	46201	Sinus
59	M	46203	Cancer
63	F	46203	Cancer

Table 5 - Generalized Table

Age	Sex	Zipcode	Disease
[20-51]	*	4680*	Cancer
[20-51]	*	4680*	Sinus
[20-51]	*	4680*	Sinus
[20-51]	*	4680*	Bronchitis
[53-63]	*	4620*	Sinus
[53-63]	*	4620*	Cancer
[53-63]	*	4620*	Cancer
[53-63]	*	4620*	Gastritis

Table 6 - Bucketizable Table

Age	Sex	Zipcode	Disease
21	M	46805	Cancer
21	F	46805	Sinus
32	F	46804	Sinus
51	F	46804	Bronchitis
53	M	46201	Sinus
59	M	46201	Cancer
59	M	46203	Cancer
63	F	46203	Gastritis

III PROPOSED WORK

In this paper, we present a novel technique called **slicing** for privacy-preserving data publishing. Our contributions include the following.

First, we introduce slicing as a new technique for privacy preserving data publishing.Slicing has several advantages when compared with generalization and bucketization. It preserves better data utility than generalization. It preserves more attribute correlations with the SAs than bucketization. It can also handle high-dimensional data and data without a clear separation of QIs and SAs.

Second, we show that slicing can be effectively used for preventing attribute disclosure, based on the privacy requirement of *l*-diversity. We introduce a notion called *l*-diverse slicing, which ensures that the adversary cannot learn the sensitive value of *any* individual with a probability greater than 1/*l*.

Third, we develop an efficient algorithm for computing the sliced table that satisfies *l*diversity. Our algorithm partitions attributes into columns, applies column generalization, and partitions tuples into buckets. Attributes that are highly correlated are in the same column; this preserves the correlations between such attributes. The associations between uncorrelated attributes are broken; the provides better privacy as the associations between such attributes are less- frequent and potentially identifying.

Fourth, we describe the intuition behind membership disclosure and explain how slicing prevents membership disclosure. A bucket of size k can potentially match *kc* tuples where *c* is the number of columns. Because only k of the *kc* tuples are actually in the original data, the existence of the other *kc* − k tuples hides the membership information of tuples in the original data.

Slicing partitions the dataset both vertically and horizontally. Vertical partitioning is done by grouping attributes into

columns based on the correlations among the attributes. Each column contains a subset of attributes that are highly correlated. Horizontal partitioning is done by grouping tuples into buckets.

Finally, within each bucket, values in each column are randomly permutated (or sorted) to break the linking between different columns. The basic idea of slicing 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.

Slicing 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 uncorrelated attributes, which areinfrequent and thus identifying. Note that when the dataset contains QIs and one SA, bucketization has to break their correlation; slicing, on the other hand, can group some QI attributes with the SA, preserving attribute correlations with the sensitive attribute.

Slicing retains improved data utility than generalization and can be recycled for membership exposure shield. Additional important benefit of slicing is that it can manage data with greater dimension. We depict how slicing can be recycled for attribute exposure protection and build an effective algorithm for calculating the sliced data that comply with the ℓ-diversity requisite. Slicing conserves enhanced utility than generalization and is more efficient than binning in assignments comprising the sensitive attribute. Slicing can be used to stop membership exposure.

A) Slicing Algorithms:

An effective slicing algorithm to obtain ℓ-diverse slicing is offered. For a given a micro data table T and two factors c and ℓ, the algorithm calculates the sliced table that involves of c columns and gratifies the privacy requisite of ℓ-diversity. Our algorithm involves of three steps: attribute partitioning column generalization and tuple partitioning. The three phases are

1)Attribute Partitioning:

Our algorithm divides attributes such that largely related attributes are in the same column. This is better for utility as well as privacy.
ith respect to data utility, clustering highly related attributes conserves the relations among those attributes. With respect to privacy, the association of not related attributes shows more identification risks than that of the association of high related attributes since the association of unrelated attribute values is very less common and therefore more identifiable. Thus, it is good to split the associations among uncorrelated attributes to guard privacy. In this step, we first calculate the relations among pairs of attributes and then group attributes on the basis of their correlations.

2) Column Generalization

Records are generalized to gratify certain minimum frequency requisite. We want to emphasize that column generalization is not a vital step in our algorithm.

3) Tuple Partitioning

In the tuple partitioning steps, records are divided into buckets. We change Mondrian algorithm for tuple partition. Not like Mondrian k-anonymity, no other generalization can be related to the records; we make use of the Mondrian for the reason of dividing tuples into buckets.

4) Membership Disclosure Protection

Let us first inspect how a challenger can conclude membership data from binning. Since binning liberates the QI values in their real form and more individuals can be solely determined using the QI values, the challenger can easily settle the membership of single individual in the real data by inspecting the regularity of the QI values in the binned information. Precisely, if the regularity is 0, the challenger knows for certain that the individual is not in information. If the regularity is higher than 0, the challenger knows with good assurance that the individual is in the information, since this similar records must fit to that unique as nearly no further individual has the identical values of QI.

5)Sliced Data

Another important advantage of slicing is its ability to handle high-dimensional data. By partitioning attributes into columns, slicing reduces the dimensionality of the data. Each column of the table can be viewed as a sub-table with a lower dimensionality. Slicing is also different from the approach of publishing multiple independent sub-tables in that these sub-tables are linked by the buckets in slicing.

Table 7- Sliced Table

Age,sex	Zipcode,disease
(21,M) (21,F) (32,F) (51,F)	(46804,Sinus) (46805, Cancer.) (46804, Bronchitis.) (46805,sinus)
(53,F) (59,M) (59,M) (63,M)	46804,Sinus) (46805, Cancer.) (46804, Bronchitis.) (46805,sinus)

IV CONCLUSION

A new method slicing method for privacy-preserving micro data publishing has been proposed . Slicing incapacitates the boundaries of generalization as well as binning and conserves improved service while safeguarding against security dangers. We show how to practice slicing to avoid attribute exposure and membership disclosure. This work persuades numerous ways for future study. By dividing attributes into columns, we secure privacy by breaching the involvement of not correlated attributes and conserve information utility by conserving the relationship between highly correlated attributes. For instance, slicing could be used for anonymizing transaction databases, which had been reviewed lately. Lastly, while a number of anonymization methods have been developed, it rests an open hindrance on how to use anonymized information. In our trials, we arbitrarily produce the correlations between column values of a bucket at the cost of loss of data utility.

Our experiments show that slicing preserves better data utility than generalization and is more effective than bucketization in workloads involving the sensitive attribute. The general methodology proposed by this work is that before anonymizing the data, one can analyze the data characteristics and use these characteristics in data anonymization. The rationale is that one can design better data anonymization techniques when we have shown the data better. We show that attribute correlations can be used for privacy attacks.

V SIMULATION WORKS/RESULTS

We have simulated our system in Dot NET. We implemented and tested with a system configuration on Intel Dual Core processor, Windows XP and using Visual Studio 2008 (C#.net). We have used the following modules in our implementation part. The details of each module for this system are as follows:

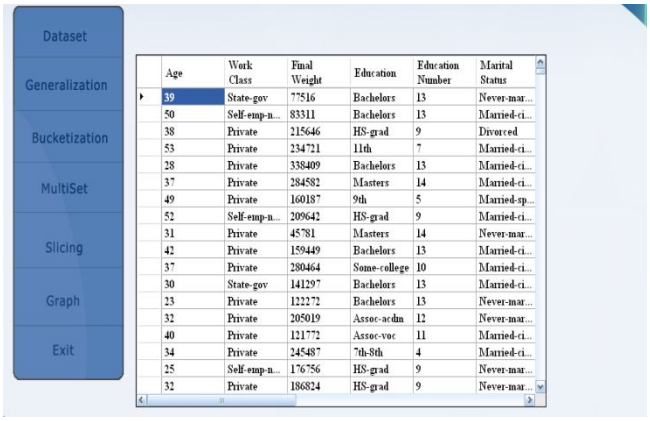

Fig 3 : Load the dataset

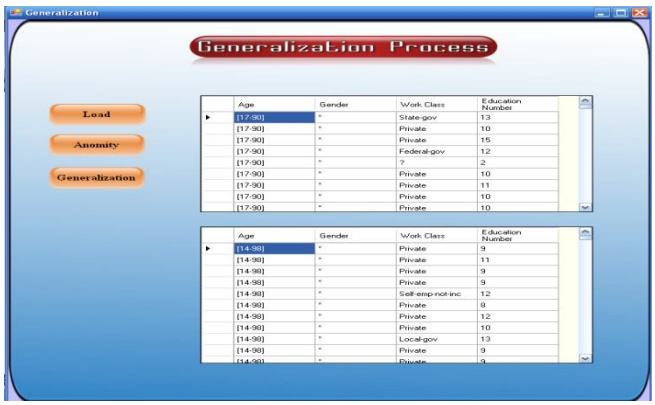

Fig 4: Execution of Generalization Process

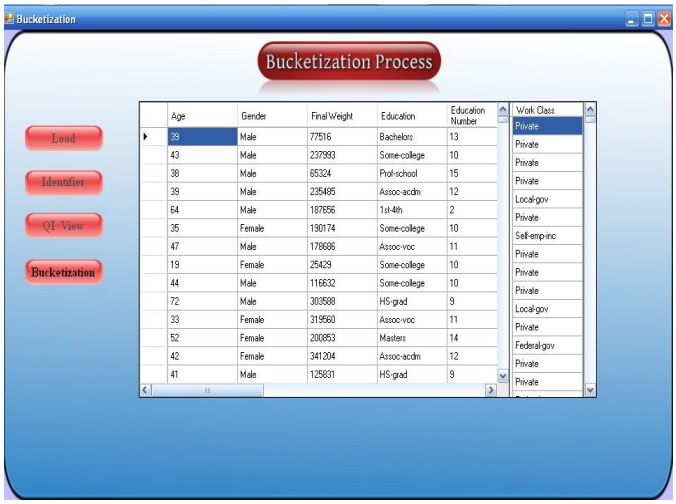

Fig 5: Execution of Bucketization Process

Fig 6 : Resultant of Overlapping Slicing

V REFERENCES

[1] E. Bertino, D. Lin, W. Jiang (2008). A Survey of Quantification of Privacy. In: Privacy-Preserving Data Mining. Springer US, Vol 34, pp. 183-205.

[2] R. J. Bayardo, R. Agrawal (2005). Data privacy through optimal k-anonymization. In: Proc. of the 21st International Conference on Data Engineering, IEEE Computer Society, pp. 217-228.

[3] K. Liu, H. Kargupta, J. Ryan (2006). Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. IEEE Transactions on Knowledge and Data Engineering, Vol 18(1), pp. 92–106

[4] P. Samarati (2001). Protecting respondents‟ identities in microdata release. IEEE Transactions on Knowledge and Data Engineering, VOl 13(6), pp. 1010–1027

[5**]** L. Sweeney (2002). Achieving k-anonymity privacy protection using generalization and suppression. International Journal of Uncertainty, Fuzziness and Knowledge Based Systems, Vol 10(5), pp. 571–588

[6] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati (2007). k-Anonymity. In: Secure Data Management in Decentralized Systems. Springer US, Vol 33, pp. 323-353.

[7] P. Samarati and L. Sweeney, “Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression”, In Technical Report SRI-CSL-98-04, SRI Computer Science Laboratory, 1998

[8] L. Sweeney, “k-anonymity: a model for protecting privacy”, International Journal on Uncertainty, Fuzziness and Knowledge based Systems, 2002, pp. 557-570.

[9] Latanya Sweeney “Achieving *k*-anonymity Privacy Protection Using Generalization and Suppression”,May 2002, International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10 (5), 2002; 571-588

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

[11] A. Machanavajjhala, J. Gehrke ,D. Kifer, M. Venkitasubramaniam (2007). ℓ-Diversity: Privacy Beyond k-Anonymity. ACM Transactions on Knowledge Discovery from Data, Vol 1(1), Article: 3.

[12**]** Li N., Li T., Venkatasubramanian S: *t*-Closeness: Orivacy beyond *k*-anonymity and *l*-diversity. *ICDE Conference*, 2007. 13. N. Zhang, “Privacy-Preserving Data Mining”, Texas A&M University, pp.19-25, 2006.

[13] Tiancheng Li, Ninghui Li, Senior Member, IEEE, Jia Zhang, Member, IEEE, and Ian Molloy “Slicing: A New Approach for Privacy Preserving Data Publishing” *Proc. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 24, NO. 3, MARCH 2012.*.

[14] 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.*

[15**]** A. Meyerson and R. Williams. "On the complexity of optimal k-anonymity", In Proceedings of PODS’04, pages 223–228, New York,NY, USA, 2004. ACM.