Privacy Preserved Data Publishing Techniques for Tabular Data

Keerthy C.
College of Engineering
Trivandrum

Sabitha S.
College of Engineering
Trivandrum

ABSTRACT

Almost all countries have imposed strict laws on the disclosure of Personally Identifiable Information(PII). However PII need to be published for many purposes like research. In such cases, we apply different types of methods like anonymization, encryption etc. This paper discuss about the different methods of anonymization of tabular microdata. The most popular method of data anonymization of tabular data is k-anonymity. However, it suffers from many attacks and hence l-diversity was proposed. The l-diversity anonymization also possessed various limitations and hence t-closeness was proposed. This paper summarize these anonymization techniques and their limitations.

Keywords

data anonymization, k-anonymity, l-diversity, t-closeness

1. INTRODUCTION

PII(Personally Identifiable Information) is any information which, by itself, or when combined with additional information, enables identification of an individual. This information if made publicly available, possess the threat of loss of reputation of the individual. Health-care data, criminal justice investigation reports, financial institution's data, web surfing behavior, location based services etc are examples of personally identifiable information, which posses such threats.

In many countries, there are strict laws[12] regarding the disclosure of PII. These laws clearly mentions that, data should be collected with the consent of the data subject and should be used by the person who has collected it, and also should not be used for any other purpose than for what it is collected for.

In case of health-care data, there are strict laws like HIPAA(Health Insurance Portability and Accountability Act), GINA(Genetic Information Non-discrimination Act) etc, regarding its disclosure. This data however needs be published for purposes like research. One way of publishing such data is to encrypt the data before release. Such practices makes the data useless and tedious for the purposes like research. An alternate solution is to remove direct identifiers like name, phone numbers, email addresses, social security numbers etc.. However, Latanya Sweeney[6] found out that 87% of the US population can be uniquely identified using the triplet {gender, zip, date of birth}.

Data anonymization is one of the efficient solutions, which is proposed for this problem. Data anonymization, itself is a wide area, consisting of network anonymization, dealing with anonymous logging into the network, anonymous blogging, anonymous web browsing etc.. and also social network anonymization[11], which is concerned with anonymization of social networks, file sharing etc.. In this paper, we discuss about different techniques of anonymization of tabular micro data, consisting of tables describing entities.

The attributes in a tabular dataset, which is prone to privacy attack can be classified into three, which are

a. *identifiers* : Those attributes, which can directly link to an individual.
Examples : names, mailing address, social security numbers etc.

b. *quasi-identifiers* : Those non-sensitive attributes, which when combined with publicly available information like voters list, could help to identify an individual.
Examples : zipcode, social security numbers etc.

c. *sensitive attributes* : Those attributes, which should be protected, that is, the set of attributes a person doesn't like to be linked with.
Examples : disease, account balance, sexual orientation etc.

The anonymization of tabular-data uses methods like generalization and suppression for the quasi-identifier attributes and it removes the sensitive attributes to group the datasets into equivalence classes. The quasi-identifiers will be generalized based on the value-generalization hierarchies. The generalization hierarchy for the quasi-identifier attribute age is shown in the figure 1.

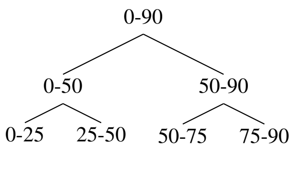

Fig. 1: Domain Generalization Hierarchy for age attribute

The organization of the paper is as follows.This paper first discuss k-anonymity, its different implementations and limitations. Then *l*-diversity, which was proposed as a solution to different problems of k-anonymity, along with its different instantiations, implementations and limitations are discussed. Then finally another anonymization technique, t-closeness and its limitations are discussed.

2. K-ANONYMITY

DEFINITION 1. *Let* $R_T(A_1,...,A_n)$ *be a table and* QI_{R_T} *be the quasi-identifier associated with it.* R_T *is said to satisfy k-anonymity if and only if each sequence of values in* $R_T[QI_{R_T}]$ *appears with at least 'k' occurrences in* $R_T[QI_{R_T}]$ *[1].*

EXAMPLE 2.1. *The table referenced in table 1, is a 3-anonymized dataset, where the quasi-identifiers are 'marital status', 'date of birth' and 'zip code', and the sensitive attribute is 'Crime'. In this table, we can see that there are atleast 3 tuples with same set of quasi-identifiers, that is* $\{1,3,6,9\}$,$\{2,5,8\}$,$\{4,7,10\}$ *are having the same values for the quasi-identifiers.*

	Identifier	Quasi-Identifier			Sensitive
ID	Name	Marital Status	DOB	Zip	Crime
1	****	Separated	1991	20001	Murder
2	****	Single	20**	19552	Theft
3	****	Separated	1991	20001	Traffic
4	****	Married	198*	36363	Assault
5	****	Single	20**	19552	Murder
6	****	Separated	1991	20001	Piracy
7	****	Married	198*	36363	Indecency
8	****	Single	20**	19552	Theft
9	****	Separated	1991	20001	Piracy
10	****	Married	198*	36363	Assault

Table 1. : k-anonymity with k=3

The following sections will discuss various k-anonymization algorithms.

2.1 Datafly [1]

The core Datafly algorithm works as follows :

1. Construct a frequency list, *freq* containing distinct sequence of values from the quasi-identifier set, along with their occurrences. Each frequency list may represent one or more tuples in the table.
2. Then the quasi-identifier attribute with the largest number of distinct values in *freq* is generalized until there are k or lesser tuples having distinct sequences in *freq*.
3. Suppress any sequence of *freq* having less than k-occurances.
4. Apply complementary suppression so that the number of suppressed tuples satisfies k-anonymity.

The problem with Datafly algorithm is that it produces generalizations that satisfy k-anonymity, but however does not guarantee k-minimal distortions, that is, generalizations with minimal information loss.

2.2 Samarati's Algorithm [2]

Samarati's algorithm proposes k-anonymity with minimal distortion of data. The algorithm considers only quasi identifiers.

Race	Zip
White	94123
Black	94124
Asian	94122
White	94122
Asian	94124
**	***
**	***
**	***
**	***

Table 2. : A Sample QID table

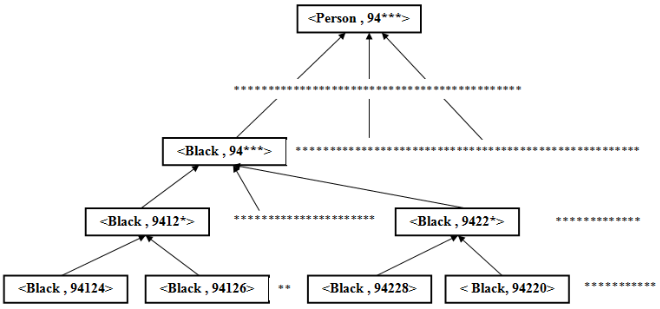

Fig. 2: lattice for generalization

1 It checks the middle height in the area of search to find atleast one node that satisfies k-anonymity with minimum suppression(the minimum suppression will be already set).
2 If not the minimum, the upper area will be set as the new area for search.
3 Else, if minimum, then lower area will be set as the new area for search.
4 If the area of search consists of more than one level in the lattice, repeat step 1
5 Otherwise, a solution is residing on this level.

The figure 2 represents a domain hierarchy for table 2. In the figure 2 suppose that <Black, 94***> itself satisfies k-anonymity with minimal suppression, with respect to some value of k, then there is no need searching above that height for a solution, we can rather look down the path to find if there exist a solution which can produce k-anonymity with lesser suppression. The problem with this algorithm is that, this algorithm is NP-Hard.

2.3 k-optimize [3]

Bayardo and Agrawal [3] proposed an algorithm called *'k-optimize'* which obtains good solutions for k-anonymization with a reduced computational time compared to [1, 2]. This approach, for an attribute *A* with an ordered domain *D*, partitions the attribute domain into intervals so that each value in the domain appears in some interval *I*.

The order of the quasi-identifiers are then decided by assigning an integer, called index within each interval of the quasi-identifier attributes. This index assignment reflects the total order relationship over intervals in the domains and among quasi-identifier attributes. For each attributes, the union of the individual index values represents a generalization. The least value in an attribute domain can be omitted since it will certainly appear in the generalizations for that domain.

k-Optimize then builds a set enumeration tree of the set *I* of index values. The root node of the tree is always empty set. The children of a node say *n*, will enumerate those sets that can be formed by appending a single element of *I* to *n*, with a restriction that, this single element must be following every element already in *n* according to the total order previously defined. The tree represents the existence of a unique path between the root and each node. The visit of the set enumeration tree using a standard traversal strategy is equivalent to the evaluation of each possible solution to the k-anonymity problem. At each node *n* in the tree the cost of the generalization strategy represented by *n* is computed and compared against the best cost found until that point; if lower it becomes the new best cost.

2.4 Incognito [4]

Incognito algorithm [4] takes advantage of a bottom-up aggregation along dimensional hierarchies and a priori aggregate computation. The key idea of Incognito is that if a set of quasi-identifiers are k-anonymous, then any subset of this quasi-identifiers will also be k-anonymous. That means, if a dataset is k-anonymous with respect to $<$Marital Status , Zipcode$>$ then, the dataset will be k-anonymous with respect to each of marital status and zipcode, if taken alone.

Incognito uses bottom-up BFS(breadth first search) approach on the DGH(domain generalization hierarchy). The algorithm generates all possible minimal k-anonymous tables for a given private table. It first checks single-attribute subsets of the quasi-identifier. Then it iterates, checking k-anonymity with respect to large subsets in the increasing order.

For each combination, Incognito then checks for k-anonymity constraint in a bottom-up approach. When a generalization satisfies k-anonymity, all its direct generalizations should also satisfy k-anonymity. So they are no longer considered. Thus at iteration *i*, Incognito will not consider all the combinations of *i* attributes together, but only considers the generalizations that satisfied the k-anonymity constraint at iteration $i-1$.

2.5 Multi-Dimensional k-anonymity [5]

The anonymization algorithms discussed so far, works on the partitioning of the data into equivalence classes, and then they apply generalization. These partitions can be either single-dimensional or multi-dimensional. k-anonymity partitions, most commonly are single-dimensional(like [3]).

Problems with single-dimensional partitions are :

a Single-dimensional partitions are not flexible, thus there is a reduction in quality of anonymity.
b It does not perform better in preventing privacy attacks.
c When the data scattering is higher or the k values are smaller, single-dimensional partitioning often leads to one attribute partitioning.
d Single-dimensional partition methods are so data sensitive because, the results are largely influenced by a small change in data.

In Mondrian [5] k-anonymity algorithm, a multi-dimensional k-anonymity algorithm is proposed. It can do the partition on multiple attributes at the same time. Multi-dimensional partitioning problem is NP-hard and hence they propose a greedy approximation algorithm, for both categorical and numerical datasets. The time complexity of the new greedy algorithm proposed is *O(n log n)*, where n represents the number of tuples in the table which needs anonymization. The algorithm produces high-quality datasets than with single dimensional partitioning.

3. ATTACKS AGAINST K-ANONYMITY

Even though k-anonymity algorithms are efficient is producing better anonymized results, they are still subjected to many types of attacks. Few of them are discussed below :

3.1 Unsorted matching attack against k-anonymity

The unsorted matching attack [6] is an attack based on the ordering of the tuples in the relational table. For example in table 3, the position of the data tuples in the original data set and those in the anonymized data set remains the same. It gives a chance for linking attack. However, this problem can be solved by randomly sorting the elements in the anonymized dataset.

Race	Crime	Race	Crime
White	Murder	Person	Murder
Black	Theft	Person	Theft
Asian	Traffic	Person	Traffic
White	Assault	Person	Assault

Table 3. : Original and anonymized datasets showing unsorted attacks

3.2 Complementary release attack against k-anonymity

In the example table 3, all the attributes were quasi-identifiers. Usually the data set released will be such that, quasi-identifiers will be a subset of data released. So there are chances of joining this data with other external information. Therefore, while releasing a dataset, we need to consider the all the previous release of the datasets inorder to avoid the linking attacks.[6].

3.3 Temporal attack against k-anonymity

The process like tuple addition, deletion etc are dynamic, that means that new data will be added and removed over time. Therefore, the release of generalized data over time is subject to a temporal inference attack. At time *t=0*, let table T_0 be the original table and the k-anonymity solution based on T_0, will be the table RT_0. At time *t*, suppose that additional tuples were added to the table T_0, and it becomes T_t and RT_t be the k-anonymized table with respect to time *t*.

Since there is no connection between the releases RT_t and RT_0, linking of the tables RT_0 and RT_t may reveal sensitive information which may compromise k-anonymity. To avoid this problem, either all of the attributes of RT_0 should be considered as quasi identifier for subsequent releases, or subsequent releases themselves would be based on RT_0.[6]

3.4 Homogeneity Attack

k-anonymity does not provide anonymization to all groups of people. For example, consider the table in table 4. The table is k-anonymized with k=3. However, consider the last set of people where the marital status is *Single*. Even though the data set is having 3 tuples for the same set of quasi-identifiers, all those tuples are having the same value for the sensitive attributes. This makes the sensitive attribute of the person, which he doesn't wish to publish, easily identified, even though the data set is anonymized. This is because k-anonymity doesn't deal with diversity of the sensitive attributes.[7].

Quasi-Identifier			*Sensitive*
Marital Status	DOB	Zip	Crime
Separated	1991	20001	Murder
Separated	1991	20001	Theft
Separated	1991	20001	Traffic
Separated	1991	20001	Assault
Married	198*	36363	Murder
Married	198*	36363	Piracy
Married	198*	36363	Indecency
Single	*20***	*19552*	*Theft*
Single	*20***	*19552*	*Theft*
Single	*20***	*19552*	*Theft*
Single	*20***	*19552*	*Theft*

Table 4. : Homogeneity attack

3.5 Background Knowledge Attack

The attacks based on background knowledge cannot be protected by k-anonymity. For example, suppose that Jay knew that one of his colleagues was caught by police for misbehaving in public. When he looks the published dataset in table 4, he could find that it is reported that only one person has been caught for the crime indecency, and also the person is living in area with zipcode 36363. From this data, he could easily identify that the person is Jack, since he is the only person working with Jay who is living at zipcode 36363. Again, this attack is also because of the fact that k-anonymity does not deal with diversity of sensitive attributes.[7].

4. L-DIVERSITY

l-diversity [7] was proposed as a solution for background knowledge attack and homogeneity attack of k-anonymity.

DEFINITION 2. *An equivalence class of data attributes is said to be l-diverse, if it contains atleast l distinct values for the sensitive attributes. A table is said to have 'l-diversity' if every equivalence class of the table has 'l-diversity'.[9]*

The different instantiations of *l*-diversity are the following:

4.1 Distinct l-diversity

Let q^* be the set of tuples whose nonsensitive attribute values are generalized. Then this block is said to be distinct *l*-diverse if contains at least *l* "well-represented" values for the sensitive attribute S. A table is *l*-diverse if every block is *l*-diverse. The problem with the distinct *l*-diversity is that it does not prevent the probabilistic inference attacks.[9] This means that, if an equivalence class is having one value of a sensitive attribute appearing more frequently than the other attributes, then an attacker can affirm that this is the senstive value of the person with a higher probability. For example,in one equivalent class, there are ten tuples. In the senstive attribute "Crime", among the ten values, one of them is "Murder", another one value is "Piracy" and the remaining eight are "Theft". This satisfies 3-diversity. However, the attacker can still affirm that the target person's disease is "Theft" with the accuracy of 70%.

4.2 Entropy l-Diversity

For entropy *l*-diversity, every equivalence class should have sensitive attribute values distributed evenly. A table is Entropy *l*-Diverse [7] if for every q^*-block

$$-\sum_{s\in S} p_{(q^*,s)} log(p_{(q^*,s')}) \geq log(l) \tag{1}$$

where

$$p_{(q^*,s)} = \frac{n_{(q^*,s)}}{\sum_{s'\in S} n_{(q^*,s')}} \tag{2}$$

is the fraction of tuples in the q^*-block with sensitive attribute value equal to s.[7]

This simply means that the entropy of the distribution of sensitive values in each equivalence class should be at least log(*l*). However this can be restrictive because some values may be common and some may be low. This leads to the less conservative notion of *l*-diversity, recursive(c,l) diversity.

4.3 Recursive (c, l)-Diversity

In a given q^*-block, let r_i denote the number of times the i^{th} most frequent sensitive value appears in that q^*-block. Given a constant c, the q^*-block satisfies recursive (*c,l*)-diversity if $r_1 <$ c(r_l + r_{l+1} + ... + r_m). A table T^* satisfies recursive (c, *l*)-diversity if every q^*-block satisfies recursive *l*-diversity.[7].

4.4 Anatomy[8]

Inorder to overcome the defects of generalization, anatomy, an *l*-diverse algorithm was proposed. This algorithm achieves *l*-diversity by capturing the exact quasi identifier distribution.The anatomy algorithm releases a *quasi-identifier table (QIT)* and a *sensitive table (ST)*, that separate quasi-identifier-values from sensitive values. The algorithm works as follows :

1 Partition the tuples into several quasi-identifier-groups, based on a certain strategy.

2 Then, create the QIT. The Quasi-Identifier table contains the exact value of the quasi-identifier, (not the generalized value), together with a group number in a new column group-id. But this table does not contain the value of the sensitive attribute.

3 Finally produce the ST. This table consist of group-id, sensitive attribute and its count.

Quasi-Identifier Table			
Age	Sex	Zip	Group-Id
23	Male	695100	1
29	Female	394564	1
32	Male	789526	1
33	Male	978578	1
48	Female	679824	1
60	Female	789582	2
65	Male	987485	2
50	Male	789582	2
52	Male	369782	2

Sensitive Table		
Group-id	Crime	Count
1	Murder	1
1	Theft	2
1	Traffic	1
2	Assault	1
2	Piracy	1
2	Indecency	1

Table 5. : Anatomy

Consider the tables in table 5. There are two tables, quasi-identifier table and sensitive table. Every record in the QIT is associated with a group-id, and those groups are detailed in the SID with a count value. For example, consider the record with zipcode 789526. This record identifies that the person belongs to group 1. In the sensitive table, we can find that, three crimes are associated with group 1, where two people committed theft, one murder and one traffic violation. So we can only make an assumption that the person is having one of the three sensitive attributes. Anatomy preserves privacy because the QIT does not indicate the sensitive value of any tuple, which must be randomly guessed from the ST.[8]

5. LIMITATIONS OF L-DIVERSITY

The following limitations of *l*-diversity was discussed in [9].

1 *l*-diversity may be difficult and unnecessary to achieve.

2 *l*-diversity is insufficient to prevent attribute disclosure.

3 *l*-diversity suffers from two types of attacks: Skewness Attack and Similarity Attack

5.1 *l*-diversity may be difficult and unnecessary to achieve

Suppose that for some tabular data, the sensitive values may have only two values- say positive and negative. The value of positive may cover upto, say 90%. Therefore for a dataset of say 100 people, 90 persons will be having the value of sensitive attribute as positive and 10 may be negative. So even if we choose the value of *l* to be 2, there will be a huge data loss inorder to achieve *l*-diversity.

5.2 Skewness attack

Consider that the sensitive value is the result of some tests like HIV. Suppose that have equal number of positive and negative values, that is 50% positive values and 50% negative values. In this case, 2-diversity is easily achieved. However, the problem here is that these sensitive values are having different levels of privacy-risks. Consider that when anonymize using 2-diversity, suppose that there are 49 positive values and 1 negative value. Then a person an equivalence class could be identified with 98% chance of positive, which is a more serious than being identified as 98% negative. This is the skewness attack.

5.3 Similarity attack

This attack occurs due to the similarities of the sensitive values. Suppose that, we have a 2-diverse table as shown in table 6. In this table, from the sensitive attribute of the last equivalence class, we can infer that the person belonging to that class, suffers from a neurological problem. This is because of the fact that, the even though *l*-diversity deals with "distinct" values of the attributes, it does not take into account the semantic closeness of the attributes[9].

Quasi-Identifier			*Sensitive*
Marital Status	DOB	Zip	Disease
Separated	1991	20001	AIDS
Separated	1991	20001	Cancer
Separated	1991	20001	Chicken pox
Separated	1991	20001	Malaria
Married	198*	36363	Alexia
Married	198*	36363	Fatty Lever
Married	198*	36363	Cancer
Single	*20***	*19552*	*Amnesia*
Single	*20***	*19552*	*Autism*
Single	*20***	*19552*	*Cerebral palsy*
Single	*20***	*19552*	*Autism*

Table 6. : Similarity attack

6. T-CLOSENESS

t-closeness is an extension of k-anonymity which tries to solve the attribute disclosure problem.

DEFINITION 3. *An equivalence class is said to have t-closeness if the distance between the distribution of a sensitive attribute in this class and the distribution of the attribute in the whole table is no more than a threshold t. A table is said to have t-closeness if all equivalence classes have t-closeness.[9]*

t-closeness defines privacy as measure of information gain of an observer. This gain is the difference between the prior belief(knowledge before the dataset is released) and the posterior belief(knowledge after the dataset is released). The difference between the posterior and prior belief is large means that the new released datasets contains a lots of new information. So for achieving t-closeness, we use distance measures like Earth mover distance and then we limit the gain between the two releases, so as to limit the information gain between the two release.

6.1 Limitation of t-closeness

The following limitations were cited about *t*-closeness.

1 There is no computational procedure to enforce *t*-closeness followed in.[13]

2 Since each values of the attributes are generalized separately, their dependence and co-relation on each other is lost.[13]

3 Smaller value of *t* results in a larger computational time and lesser utility of the data.[9]

7. CONCLUSION

This paper, discussed about various anonymization methods and different algorithms for anonymization. Different implementations of k-anonymity were discussed, each with their limitations. k-anonymity algorithms have the problems of homogeneity attack and background knowledge attack. These problems were solved by the introduction of *l*-diversity. However, the *l*-diversity have many limitations, including that they could be only used for categorical data. The problems of *l*-diversity like skewness attack, similarity attack were resolved by t-closeness. However, *t*-closeness is impractical to achieve. There are different implementations of k-anonymity that could achieve results in better execution time. Due to these facts, apart from *l*-diversity and *t*-closeness algorithms, k-anonymity algorithms are popular and widely used. However, if practical models which could achieve *l*-diversity and *t*-closeness in better execution time are developed, these methods still prove to be better than k-anonymity.

8. REFERENCES

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

[2] Pierangela Samarati, *Protecting Respondents' Identities in Microdata Release* in IEEE Transactions On Knowledge and Data Engineering, VOL. 13, NO. 6, November/December 2001

[3] Roberto J. Bayardo, Rakesh Agrawal, *Data Privacy Through Optimal k-Anonymization* in Proceedings of the 21st International Conference on Data Engineering 2005, 217-228, Tokyo, Japan

[4] Kristen LeFevre, David J. DeWitt, Raghu Ramakrishnan *Incognito: Efficient full-domain k-anonymity* in Proceedings of the 24th ACM SIGMOD International Conference on Management of Data, pp. 49-60, Baltimore, Maryland, USA.

[5] Kristen LeFevre, David J. DeWitt, Raghu Ramakrishnan, *Mondrian Multidimensional K-Anonymity* in Proceedings of the International Conference on Data Engineering (ICDE'06), Atlanta, GA, USA.

[6] Latanya Sweeney *k-Anonymity: A Model For Protecting Privacy* in International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10 (5), 2002; 557-570.

[7] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, *l-Diversity: Privacy Beyond k-Anonymity* in Proceedings of the International Conference on Data Engineering(ICDE'06), Atlanta, GA, USA.

[8] Xiaokui Xiao, Yufei Tao *Anatomy: Simple and Effective Privacy Preservation* in Proceedings of the International Conference on Very Large Databases.

[9] Ninghui Li, Tiancheng Li, Suresh Venkatasubramanian *t-Closeness: Privacy Beyond k-Anonymity and l-Diversity* in IEEE 23rd International Conference on Data Engineering, 2007.

[10] Ninghui Li, Ninghui Li, Venkatasubramanian, S *Closeness: A New Privacy Measure for Data Publishing* in IEEE Transactions on Knowledge and Data Engineering,Volume:22, Issue: 7 943 - 956, 2009.

[11] Kenneth L. Clarkson, Kun Liu, Evimaria Terzi *Towards Identity Anonymization in Social Networks* in Proceedings of the 2008 ACM SIGMOD International Conference on Data Management.

[12] Data Protection Laws of the World, *https://www.dlapiperdataprotection.com*

[13] S.Deebika, A.Sathyapriya, S.K.Kiruba *Survey Result on Privacy Preserving Techniques in Data Publishing* in International Journal of Latest Trends in Engineering and Technology Volume:3 Issue:2 November 2013.