Available online at www.pelagiaresearchlibrary.com

Pelagia Research Library

Advances in Applied Science Research, 2011, 2 (3):302-313

ISSN: 0976-8610
CODEN (USA): AASRFC

A Rough Set Based Efficient *l*-diversity Algorithm

B. K. Tripathy[1], G. K. Panda[2*] and K. Kumaran[3]

[1]SCSE, VIT University, Vellore, Tamilnadu, India
[2]Department of CSE&IT, M.I.T.S., Rayagada, Odisha, India
[3]SITE, VIT University, Vellore, Tamilnadu, India

ABSTRACT

Most of the organizations publish micro data for a variety of purposes including demographic and public health research. To protect the anonymity of the entities, data holders often remove or encrypt explicit identifiers. But, released information often contains quasi identifiers, which leak valuable information. Samarati and Sweeney introduced the concept of k-anonymity to handle this problem and several algorithms have been introduced by different authors in recent times. Lin et al put forth a new clustering-based method known as OKA for k-anonymization. But, k-anonymity can create groups that leak information due to homogeneity attack. This problem is tackled by the notion of l- diversity introduced by Machanavajjhala et al. Recently, the OKA algorithm is improved by Tripathy et al by making some modifications in the adjustment stage and introducing distinct l-diversity into it. But, in most of the modern databases impreciseness has become a common characteristic, which is not handled by any of the above algorithms. The primary purpose of this paper is to use MMeR, an algorithm introduced by Tripathy et al, in developing a suitable anonymisation algorithm which is applicable to any database having precise or imprecise heterogeneous data and satisfies both k-anonymity as well as l-diversity properties.

Keywords: OKA, MMeR, data privacy, k-anonymity, rough sets.

INTRODUCTION

Most of the organizations publish micro data (i.e. data published in its raw, non-aggregated form) for a variety of purposes including demographic and public health research. To protect the anonymity of the entities, called respondents, data holders often remove or encrypt explicit identifiers such as names, addresses, and phone numbers. De-identifying data, however, provides no guarantee of anonymity. Released information often contains other data, such as race, birth date, sex and ZIP code, which can be linked to publicly available information to re identify the data respondents, thus leaking information that was not intended for disclosure. Such types of attributes are called quasi identifiers. The large amount of information easily accessible today, together with the increased computational power available to the attackers, makes linking attacks a serious problem (Samarati et al [10]). To avoid such attacks while preserving the integrity of the released data, Samarati and Sweeney [11] proposed the concept of k-anonymity. In later years it was further expanded by Sweeney [13] to the context of table releases. A k-anonymized dataset has the property that each record is indistinguishable from at least k-1 other records within the dataset. The larger the value of k the greater the implied privacy, since no individual can be identified with probability higher than 1/k through the linking attacks alone. While *k*-anonymity protects against identity disclosure, it does not provide sufficient protection against attribute. In order to obtain k-anonymity, several algorithms have been implemented in recent times [Agrawal and Bayardo [1], Byun et al [2], Chiu and Tsai [3], Lin et al [6], Loukides and Shao [7], Machanavajjhala et al [8],Samarati et al [10],Sweeny [13].

There are different attacks on k-anonymity which can compromise on the k-anonymity dataset. Background knowledge and homogeneity attacks are the two attacks where a k anonymous table may disclose the sensitive information. Thus k-anonymity can create groups that leak information due to lack of diversity in the sensitive attribute which leads to homogeneity attack. So, k-anonymity does not protect against attacks based on prior knowledge of the adversary which results in background knowledge attack. This aroused the importance of a greater privacy preserving notion and thus l- diversity (Machanavajjhala [8]) was proposed. In fact l-diversity provides privacy even when the data publisher does not know what kind of knowledge the adversary possesses. The main idea behind *l*-diversity is the requirement that the values of the sensitive attributes are well represented in each group.

In Lin et al [6], a new clustering-based method known as OKA (One pass k- means algorithm) is proposed for k-anonymization. This method has a time complexity of O(n^2/k), where n is the number of records. This algorithm has advantages over some of the preceding algorithms proposed by Byun et al [2], Loukides and Shao [7] and Chiu and Tsai [3]. The OKA algorithm has two phases. It first clusters the data tuples and then in the adjustment stage makes up the sizes of the clusters to have a minimum of k elements each.

Recently, this algorithm has been improved by Tripathy et al [17] by making some modifications and introducing distinct l-diversity into it. However, none of the above algorithms take care of impreciseness in data. But, in most of the modern databases impreciseness has become a common characteristic. The basic purpose of this paper is to develop suitable algorithms such that any database having precise or imprecise heterogeneous data can be anonymized before its publication while managing both k-anonymity as well as l-diversity property.

MATERIALS AND METHODS

The concept of Rough set introduced by Pawlak [9], has been a wonderful model to capture impreciseness in data. Using the rough set concept, a very efficient clustering algorithm called MMeR (Min Mean Roughness), was introduced by Tripathy and Prakash Kumar [15, 16] which takes care of heterogeneous data that is both numeric and categorical data can be handled simultaneously. It has been established that this clustering algorithm is the most efficient among all the clustering algorithms developed so far.

So, we tried to use this clustering algorithm instead of the clustering stage algorithm of OKA. It improves the performance of the OKA algorithm and also, impreciseness in data could be handled. Also, this approach transforms the algorithm into the best among all the k-means algorithms.

At least three directions of improvement have been mentioned by Lin et al [6]. Out of these, one proposal is to improve the adjustment stage. We have achieved this through the following steps:

I. In the OKA algorithm, excess records from the clusters having more than k records are taken out, basing upon their distance from the centroid and are collected. These records are added to the clusters having less than k records to make up their size to k. If any additional record still remains unassigned then these are added to their nearest clusters. This adds to the complexity of the algorithm. However, it is clear that these records are closer to their parent clusters. So, we kept the cluster identity along with every record taken out and return the excess records to their parent clusters. This change makes the adjustment stage more efficient.

II. We tried to handle the small cluster problem in the following manner. In fact, we propose for merging of the records in clusters of size less than k/2 to the clusters of size lying between k/2 and k. While doing this we find the nearest cluster among those are suitable. This is done before step I above so that the number of record transfers and distance comparisons become less. The modification has two advantages. First, we don't have to transfer too many records to make up the size of the small clusters. Next, the sizes of the clusters having cardinality lying between k/2 and k have been improved so that the number of transfers of records from clusters having size greater than k is very much reduced.

Finally, we find that the distinct l-diversity algorithm proposed in Tripathy et al [17] has some problems in it. The clusters having diversity less than 'l' are taken one at a time and are compared with those having diversity at least 'l'. If some records are found in the later clusters, which have sensitive attribute values which are not there in the first cluster then two such clusters are swapped. This process may lead to the following problems:

I. If the number of such records is only one in the second cluster then the swapping may reduce its diversity.

II. Also, the swapping cannot be carried out with any tuple of the first cluster. If we change a record with count of sensitive attribute value 'one' then the diversity of the first cluster shall not increase.

So, we modified the third phase algorithm in Tripathy et al [17], in order to rectify these two drawbacks. To achieve this, we go ahead with the swapping only when the multiplicity is a minimum of two for both the records. Also, we find that some of the clusters which do not have the required l-diversity at the end of the swapping of records between the set of clusters which satisfy l-diversity and those which do not can be adjusted among themselves to achieve the required diversity. So, we go for one pass for such adjustments. In case, this is also not sufficient we go for merging of these clusters with the nearest cluster satisfying l-diversity.

DEFINITIONS AND NOTATIONS

The notion of rough sets as a model to capture impreciseness in data was introduced by Pawlak [9]. Since its inception many fruitful applications have been found in various fields. The basic assumption in rough set theory is that human knowledge depends upon their capability to classify objects. As classification of universes and equivalence relations are interchangeable notions, for mathematical reasons equivalence relations are used to define rough sets. A rough set is represented by a pair of crisp sets, called the lower approximation, which comprises of elements belonging to it and upper approximation, which comprises of elements possibly in the set with respect to the available information.

3.1 Basic Rough Sets

Let U be a universe of discourse and A be a set of attributes. With every attribute $a \in A$ we associate a set V_a of its values, called the domain of a. Any subset B of A determines a binary relation I(B) on U, which will be called an indiscernibility relation and is defined as follows:

x I(B) y if and only if a(x) = a(y) for every $a \in A$, where a(x) denotes the value of attribute a for element x.

It is clear that I(B) is an equivalence relation. The family of all equivalence classes of I(B), that is partition determined by B, will be denoted by U/I(B), or simply U/B; an equivalence class of I(B), that is block of the partition U/B, containing x will be denoted by B(x).

If (x, y) belongs to I(B) we will say that x and y are B-indiscernible. Equivalence classes of the relation I(B) (or blocks of the partition U/B) are referred to as B-elementary sets. In the rough set approach the elementary sets are the basic building blocks (concepts) of our knowledge about reality. The indiscernibility relation will be used next to define approximations, basic concepts of rough set theory.

Definition 1

The approximations can be defined as follows:

$$\underline{X}_B = \{x \in U : B(x) \subseteq X\},$$

$$\overline{X}_B = \{x \in U : B(x) \cap X \neq \phi\},$$

assigning to every subset X of the universe U two sets $\underline{X}_B$ *and* $\overline{X}_B$ called the *B-lower* and the *B-upper approximation of X*, respectively. The set $BN_B(X) = \overline{X}_B - \underline{X}_B$ is referred to as the *B-boundary region* of X.

Definition 2

If the boundary region of X is the empty set, that is $BN_B(X)=\phi$, then the set X is *crisp (exact) with respect to B*; in the opposite case, that is if $BN_R(X)\neq\phi$, the set X is to as *rough (inexact) with respect to B.*

We denote the equivalence class of xi in the relation I (B) by$[x_i]_{I(B)}$, which is also known as elementary set in B.

Definition 3

The ratio of the cardinality of the lower approximation and the cardinality of the upper approximation is defined as the *accuracy of approximation*, which is a measure of roughness. It is presented as

$$R_B(X)=1-(|\underline{X_B}|/|\overline{X_B}|)$$

Definition 4

Given $a_i \in A$, X is a subset of objects having one specific value α of attribute a_j, $\underline{X_{a_j}(a_i=\alpha)}$ and $\overline{X_{a_j}(a_i=\alpha)}$ refer to the *lower* and *upper approximation* with respect to {a_j}, then $R_{a_j}(X)$ is defined as the *roughness of X with respect to {* a_j *}*, that is

$$R_{a_j}(X/a_i=\alpha)=1-\frac{\left|\underline{X_{a_j}(a_i=\alpha)}\right|}{\left|\overline{X_{a_j}(a_i=\alpha)}\right|},\quad \text{where } a_i, a_j \in A \text{ and } a_i \neq a_j.$$

The *mean roughness on attribute* a_i *with respect to {* a_j *}* is defined as

$$\text{Rough}_{a_j}(a_i)=\frac{R_{a_j}(X|a_i=\alpha_1)+\ldots R_{a_i}(X|a_i=\alpha_{|V(a_i)|})}{|V(a_i)|},$$

where $a_i, a_j \in A$ and $a_i \neq a_j$.

Definition 5

Given n attributes, MR, min-roughness of attribute $a_i(a_i \in A)$ refers to the minimum of the mean roughness, that is,

$$\text{MR}(a_i)=\text{Min}(\text{Rough}_{a_1}(a_i),\ldots..\text{Rough}_{a_j}(a_i)\ldots.),\ \text{where } a_i, a_j \in A, a_i \neq a_j, a_i \neq a_j, 1\leq i, j \leq n.$$

And we define

$$\text{MMR}=\text{Min}(\text{MR}(a_1),\ldots\ldots\text{MR}(a_i),\ldots.), \text{where } a_i \in A, \text{i goes from 1 to cardinality(A)}.$$

Definition 6

Given n attributes, and each attribute $(a_i \in A)$ can generate equivalence classes like objects obtained from a_i = a. Mean roughness for an equivalence class (MeR) is defined as the summation of roughness values of each equivalence class of certain attribute with respect to other attributes

$$\mathrm{MeR}(a_i = a) = (\sum_{j=1}^{n} R_{a_j}(X/a_i = a))/(n-1)$$

And MMeR is defined as

$$\mathrm{MMeR} = \mathrm{Min}(\mathrm{MeR}(a_i = \alpha_1), \ldots\ldots\ldots \mathrm{MeR}(a_i = \alpha_{k_i})),$$

i=1, 2…n; k_i is the number of equivalence classes in the domain of a_i.

3.2 Information loss

The notion of information loss is used to quantify the amount of information that is lost due to k-anonymisation. We follow the notions used in Byun et al [2] for this section.

Let T denote a set of records, which is described by m numeric quasi-identifiers $N_1, N_2, \ldots N_m$ and q categorical quasi-identifiers $C_1, C_2, \ldots C_q$. Let P = $\{P_1, P_2, \ldots P_p\}$ be a partition of T. Each categorical attribute C_i is associated with a taxonomy tree T_{C_i} that is used to generalize the values of this attribute.

Consider a set $P \subseteq T$ of records. Let $\hat{N}_i(P), \check{N}_i(P)$ *and* $\overline{N_i}(P)$ denote the maximum, minimum and average values of the records in P with respect to the numeric attribute N_i. Also, let $C_i(P)$ denote the set of values of records in P with respect to the categorical attribute C_i, and let $T_{C_i}(P)$ denote the maximal sub tree of T_{C_i} rooted at the lowest common ancestor of the values of $C_i(P)$. Then, the diversity of P, denoted by D(P), is defined as,

$$D(P) = \sum_{i \in [1,..m]} (\hat{N}_i(P) - \check{N}_i(P)) / (\hat{N}_i(T) - \check{N}_i(T)) + \sum_{i \in [1,...q]} H(T_{C_i}(P)) / H(T_{C_i})$$

Where H(T) represents the height of the tree T.

Let r’ and r* be two records, then the distance between r’ and r* is defined as the diversity of the set {r’, r*}.

The centroid of P is a record whose value of attributes is at minimum distance from all other attribute values in P. To anonymise the records in P means to generalize these records to the same values with respect to each quasi- identifier. The amount of information loss occurred by such a process, denoted as L(P), is defined as

$L(P) = |P| \times D(P)$, where $|P|$ represents the number of records in P.

THE ALGORITHMS

In this section we shall describe the algorithms used to achieve l-diversity in this paper.

4.1. The Clustering Algorithm (MMeR)

We are using the unaltered algorithm MMeR for clustering of heterogeneous data as in Tripathy and Prakash Kumar [15, 16]. The algorithm is as follows:

```
Procedure MMeR (U, k)
Begin
        Set current number of cluster CNC = 1
        Set ParentNode = U
Loop1:
        If CNC < k and CNC $\neq$ 1 then
            ParentNode = ProcParentNode (CNC)
        End if
        // Clustering the ParentNode
        For each $a_i \in A$ (i = 1 to n, where n is the number of attributes in A)
             Determine $[x_m]_{I(a_i)}$ (m = 1 to number of objects)
             For each $a_j \in A$(j = 1 to n, where n is the number of attributes in A, j $\neq$ i)
                 Calculate $Rough_{a_j}(a_i)$
             Next
             Mean-Roughness ( $a_i$ ) = Mean ( $Rough_{a_j}(a_i)$ )
        Next
        Set Min–Mean-Roughness =Min (Mean-Roughness ( $a_i$ )), i = 1,. . .,n
        Determine splitting attribute $a_i$ corresponding to the Min–Mean-Roughness
        Do binary split on the splitting attribute $a_i$
          CNC = the number of leaf nodes
          Go to Loop1:
End

ProcParentNode (CNC)
Begin
        Set i = 1
        Do until i < CNC
        If  Avg-distance of cluster i is calculated
            Goto label
        else
            n = Count (Set of Elements in Cluster i).
          Avg-distance (i)= $2*(\sum_{j=1}^{n-1}\sum_{k=j+1}^{n}$ (Hamming distance between objects $a_j$ and $a_k$))/(n*(n-1))
         label :
            increment i
        Loop
        Determine Max (Avg-distance(i))
        Return (Set of Elements in cluster i) corresponding to Max (Avg-distance (i))
End
```

4.2. The Adjustment algorithm

The adjustment stage algorithm proposed by Lin et al [6] for the second stage takes the outputs of the first stage and applies a procedure, using which the clusters having less than k elements are compensated with elements taken from those clusters which have more than k elements. However, after adding suitable number of elements to make the number of elements in all the clusters more than k, the rest of the elements if any are again distributed among all the clusters such that they are placed in the clusters to which they are closet. But, it obviously increases the complexity in terms of processing time. It is clear from the first stage that the elements are clearly closest to the clusters from which they have been chosen. So the algorithm can be modified to take care of the return of the excess elements if any to their parent clusters. We present the slightly modified algorithm as follows:

Input : a partitioning $P = \{P_1 \ldots\ldots\ldots P_K\}$ of T
Output: an adjusted partitioning $P = \{P_1 \ldots\ldots P_K\}$ of T

```
Let S: = Ø;
For each cluster P$\in$ $P$ with |P | < k/2 do
    Do S = $S \cup P$;
While ( $S \neq \phi$ ) do
    Randomly select a record r from S;
    If P contains cluster $P_i$ with $k/2 < |P_i| < k$ do
        Add r to the closest such cluster;
    Else add r to the closest cluster in $P$
End of While
Let R: = Ø;
For each cluster P$\in$ $P$ with |P | > k do
    Sort records in P by distance to centroid of P;
    While (|P | > k) do
        r $\in$ P is the record farthest from centroid of P ;
        Let P: = P \ {r}; R: = $R \cup \{r\}$ and c = Index(P);
    End of While
End of For
While (R ≠ Ø) do
    Randomly select a record r from R;
    Let R: = R \ {r};
    If $P$ contains cluster $P_i$ such that $|P_i| < k$ then
        Add r to its closest cluster P satisfying $|P_i| < k$ ;
    Else
        Add r to its cluster $P_c$;
    End if
End While
```

After the completion of adjustment stage, the following algorithm is to be used to achieve *l*-diversity in the clusters:

4.3. Algorithm for l-diversity

Input : Clusters formed after adjustment stage (m in number)
Output: Clusters satisfying l-diversity

1. Let P be the matrix of frequencies of attribute values, whose columns correspond to the clusters and rows correspond to the different attribute values in the domain of the sensitive attribute. The last row contains the diversity values (d_i) for the clusters (equal to the number of non-zero values in the corresponding column). The entries in P other than those in the last row contain frequencies of attribute values in the clusters.
2. Order the columns in P according to the ascending order of the diversity values.
3. Let q = $\max\{i : d_i < l\}$.
4. For each cluster C_i with $1 \le i \le q$, compare with cluster C_j, j = q+1… m.
5. F = {the sensitive attribute values which are in C_j but not in C_i and have frequency greater than 1}. Find $m_i = \min\{(l - d_i), |F|\}$ of them which are closest to the tuples in C_i.
6. Interchange m_i tuples between C_i (Those tuples with sensitive values > 1) and C_j s.
7. Increment the diversity of C_i by m_i.
8. Continue the process till the diversity of all C_i is ’l’ or no cluster is left in $\{C_j, q+1 \le j \le m\}$ for comparison.
9. Let L = {D1, D2,…Dr}, where each Di has diversity less than ‘l’
10. For j= 1 to r, compare with cluster Cp, p = j+1… r.
11. G = {the sensitive attribute values which are in C_p but not in C_j and have frequency greater than 1}. Find $m_j = \min\{(l - d_j), |G|\}$ of them which are closest to the tuples in C_p.
12. Interchange m_j tuples between C_p (Those tuples with sensitive values > 1) and C_j s.
13. Increment the diversity of C_j by m_j.
14. If diversity of $C_i \neq l$ for some $1 \le i \le q$ then merge it with some cluster with diversity $\ge l$ and closest to C_i.

IMPLICATIONS

The effect and efficiency of these algorithms in anonymisation can be seen from the following example. A complete implementation was done for the three algorithms using JAVA. We illustrate below through an example toy database as how the algorithms run and the results obtained step wise.

5.1. An Example

The following example illustrates the four different stages of the functioning of the algorithm, where Table 1 is the original data table and Table 2, Table 3 and Table 4 are the results after each of the three algorithms being executed. Table 5 is the final table for publication and satisfies 3-annonymity as well as 3-diversity.

Here, we use the following taxonomy trees for the attributes:

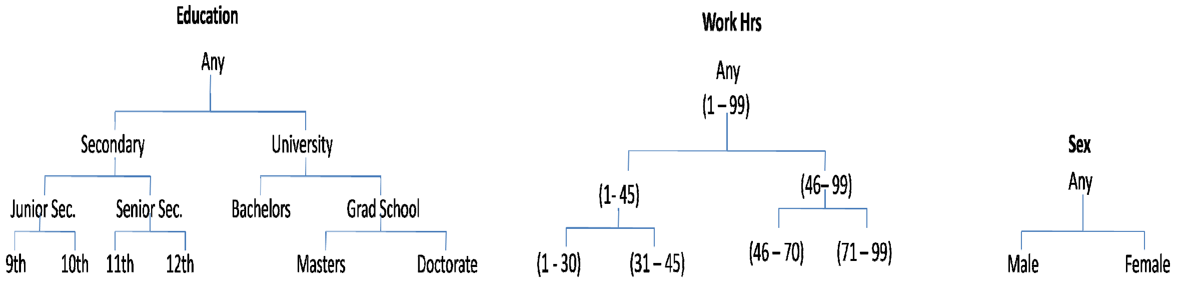

Table 1: Base Table

Education	**Sex**	**Work Hours**	**Disease**
9th	Male	30	Cholera
9th	Male	32	Bronchitis
9th	Male	33	Flu
10th	Female	35	Flu
10th	Female	36	Cholera
11th	Male	37	Bronchitis
11th	Male	37	Flu
12th	Male	38	Cholera
12th	Female	38	Flu
Bachelors	Female	39	Bronchitis
Bachelors	Female	39	Bronchitis
Masters	Female	40	Flu
Masters	Male	41	Cholera
Masters	Male	42	Flu
Masters	Male	44	Cholera
Doctorate	Female	44	Cholera
Doctorate	Female	44	Bronchitis
Doctorate	Female	45	Flu
Doctorate	Female	45	Cholera

CONCLUSION

In this paper, we have developed a three stage algorithm which prepares a data table for publication and has k-anonymity as well as l-diversity. This algorithm is based upon some previous algorithms in this direction due to Lin, Jun-Lin and Wei, Meng-Cheng [6], Tripathy et al [17]. In fact, this algorithm takes care of impreciseness in data tables through the MMeR algorithm developed in Tripathy et al [15, 16]. We have improved the adjustment algorithm and l-diversity algorithm of Tripathy et al [17]. However, as far as l-diversity is concerned, we have taken care of distinct l-diversity only, which is not the best of the three forms of l-diversities introduced by Machanavajjhala et al [18]. So, the third stage of the algorithm can be improved to take care of such type of diversities. Also, extensions can be made to incorporate t-closeness property to make the algorithm most effective towards anonymisation.

Table 2: After Clustering Stage

Education	Sex	Work hours	Diseases
9th	Male	33	Flu
9th	Male	30	Cholera
9th	Male	32	Bronchitis
11th	Male	37	Flu
10th	Female	35	Flu
10th	Female	36	Cholera
12th	Female	30	Flu
11th	Male	37	Bronchitis
Bachelor	Female	39	Bronchitis
Masters	Male	42	Flu
12th	Male	38	Cholera
Masters	Male	41	Cholera
Masters	Male	44	Cholera
Doctorate	Female	45	Flu
Doctorate	Female	44	Bronchitis
Doctorate	Female	44	Bronchitis
Doctorate	Female	45	Cholera
Masters	Female	40	Flu
Bachelor	Female	39	Bronchitis

REFERENCES

[1] Agrawal, R. and Bayardo, R, In: Data privacy through optimal k-Anonymization, 21st International Conference on Data Engineering, **2005**, 217-218.

[2] Byun, J. W., Kamra, A., Bertino, E. and Li, N., In: Efficient k-anonymization using clustering techniques, Internal Conference on Database Systems for Advanced Applications (DASFAA), **2007**.

[3] Chiu, C. C. and Tsai, C.Y., In: A k-anonymity clustering method for effective data privacy preservation, Third International Conference on Advanced Data Mining and Applications (ADMA), **2007**.

[4] Duncan, G. T. and Lambert, D., *J. Am. Stat. Assoc.*, **1986**, 10–28.

[5] Lambert, D., *J. Official Stat.*, **1993**, 9:313.

[6] Lin, Jun-Lin and Wei, Meng-Cheng, In: An efficient clustering method for k-anonymization, 2008 international workshop on Privacy and anonymity in information society, March 29-29, **2008**, Nantes, France.

[7] Loukides, G. and Shao, J., In: Capturing data usefulness and privacy protection in k-anonymisation, 2007 ACM symposium on Applied Computing, **2007**.

[8] Machanavajjhala, A., Gehrke, J., Kifer, D and Venkitasubramaniam, M., In: *l*-diversity: Privacy beyond *k*-anonymity, *22nd Intl. Conf. Data Engg.. (ICDE)*, **2006**, 24.

[9] Pawlak, Z., Rough sets, *Intl. Journal of information and computer science*, 11, **1982**,341.

[10] Samarati, P., Foresti, S., Vimercati, S. D. C. Di and Ciriani, V., **2007**, k-anonymity,Springer (US) Advances in Information Security.

[11] Samarati, P. and Sweeney, L., In: Generalizing data to provide anonymity when disclosing information, 17th ACM-SIGMOD-SIGACT SIGART Symposium on the Principles of Database Systems, 188, Seattle, **1998**, WA, USA.

[12] Samarati, P., Protecting respondents' identities in micro data release, IEEE Transactions on Knowledge and Data Engineering, 13(6), **2001**,1010-1027.

[13] Sweeney, L., *International Journal on Uncertainty*, Fuzziness and Knowledge-based Systems, 10(5), **2002**, 571-588.

[14] Tan, P.N., Steinbach, M. and Kumar, V., Introduction to Data Mining, Addison-Wesley, Boston, **2005**, 487–559.

[15] Tripathy, B.K. and M S Prakash Kumar Ch., In: MMeR: An algorithm for clustering categorical data using Rough Set Theory, ICADABAI-**2009**, IIM Ahmadabad.

[16] Tripathy, B.K. and M S Prakash Kumar Ch., *International Journal of Rapid Manufacturing*: *special issue on Data Mining*, vol.1, no.2, **2009**,189-207.

[17] Tripathy, B.K., Devineni, H., Jayasri, K.J. and Bhargava, M., In: An Efficient Clustering Algorithm for l-diversity, International conf. on Advances And Emerging Trends in Computing Technologies,21- 24 June **2010**,SRM University,Chennai,67-73.