Volume 2, Issue 6, June 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Extending K-Anonymity to Privacy Preserving Data Mining Using Association Rule Hiding Algorithm

Dr.R. Sugumar[1], Dr.A. Rengarajan [2], M.Vijayanand[3]
[1]Associate Professor / CSE, Veltech Multi Tech SRS Engineering College, Chennai, India
[2]Associate Professor / IT, Veltech Multi Tech SRS Engineering College, Chennai, India
[3]Professor / CSE, Aksheya College of Engineering, Chennai , India
sugu16@gmail.com

Abstract - Privacy Preserving Data Mining is a research area concerned with the privacy driven from personally identifiable information when considered for data mining. k-anonymity is one of the most classic models, which prevents joining attacks by generalizing or suppressing portions of the released micro data so that no individual can be uniquely distinguished from a group of size k. This paper focuses on how to extend k-anonymity to privacy preserving data mining using association rule hiding algorithm. Association rule hiding algorithm refers to the process of modifying the original database in such a way that certain sensitive association rules disappear without seriously affecting the data and the non sensitive rules.

Keywords: K-anonymity, Privacy Preserving Data Mining, Association Rule Hiding, Generalization.

I. Introduction

The problem of privacy-preserving data mining has turn into more significant in recent years because of the growing capability to accumulate private data about users, and the ever increasing sophistication of data mining algorithms to influence this information. A number of techniques such as statistical disclosure control, distributed data privacy, randomization and k-anonymity, etc., have been recommended in recent years in order to execute data mining operations in a privacy preserving way. The goal of privacy preserving data mining is to developed algorithms [3], [4] for modifying the original data in some way, so that the private data and private knowledge remain private even after the mining process. There have been two types of privacy concerning data mining. The first type of privacy, called output privacy, is that is minimally altered so that the mining result will preserve certain privacy. The second type of privacy, input privacy, is that the data is manipulated so that mining result is not affected or minimally affected.

k-anonymity is one of the most classic models, which prevents joining attacks by generalizing or suppressing portions of the released micro data so that no individual can be uniquely distinguished from a group of size k. k-Anonymity attributes are suppressed or generalized until each row is identical with at least k-1 other rows.

One way to enable effective data mining while preserving privacy is to anonymize the data set that includes private information about subjects before being released for data mining. Anonymization method aims at making the individual record be indistinguishable among a group records by using techniques of generalization and suppression. The representative anonymization method is k-anonymity. k-anonymity is one of the most classic models, which prevents joining attacks by generalizing or suppressing portions of the released micro data so that no individual can be uniquely distinguished from a group of size k. k-Anonymity attributes are suppressed or generalized until each row is identical with at least k-1 other rows. The larger the value of k, the better the privacy is protected. k-anonymity can ensure that individuals cannot be uniquely identified by linking attacks Determining the association rules hiding is at the heart of data mining. It detects hidden linkages of otherwise seemingly unrelated data. These linkages are rules. Those that exceed a certain threshold are deemed interesting. Interesting rules allow actions to be taken based upon data pattern. They can also help making and justifying decisions.

k-Anonymity attributes are suppressed or generalized until each row is identical with at least k-1 other rows. At this point the database is said to be k-anonymous [1].

i. Suppression – can replace individual attributes with a *
ii. Generalization – replace individual attributes with a broader category

II. K-Anonymity using Generalization

The generalization hierarchy transforms the k-anonymity problem into a partitioning problem. Specifically, this approach consists of the following two steps. The first step is to find a partitioning of the dimensional space,

where n is the number of attributes in the quasi identifier, such that each partition contains at least k records. Then the records in each partition are generalized so that they all share the same quasi-identifier value. The Generalization method substitutes the values of a given attribute with more general values. Generalization can be applied at the following levels.

Attribute Generalization (AG): generalization is performed at the level of column; a generalization step generalizes all the values in the column.

Cell Generalization (CG): generalization is performed on single cells; as a result a generalized table may contain, for a specific column, values at different generalization levels.

There are two types of generalization exist namely domain generalization hierarchy and value generalization hierarchy. The domain generalization hierarchy of a domain topic is a lattice, where each vertex represents a generalized table that is obtained by generalizing the involved attributes according to the Corresponding domain tuple and by suppressing a certain number of tuples to fulfill the k-anonymity constraint.

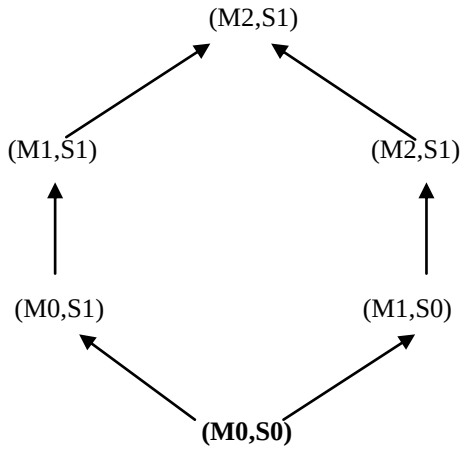

Figure 1: Generalization Hierarchies for Quasi-Identifier

Fig.1 illustrates an example of domain generalization hierarchy obtained by considering marital status and sex its quasi-identifying attributes i.e. by considering the domain tuple (Ma, So). Each path in the hierarchy corresponds to a generalization strategy according to which the original private table PT can be generalized

A. K-Anonymity Using Suppression

K-anonymity together with its enforcement via generalization and suppression has been therefore proposed as an approach to protect respondents' identities while releasing truthful information. A method adopted in [2] to be applied in to obtain k-anonymity is tuple suppression. The idea behind the introduction of suppression is that this additional method can reduce the amount of generalization necessary to satisfy the k-anonymity constraint. Suppression is therefore used to moderate the generalization process when a limited number of outliers (i.e., tuples with less than k occurrences) would force a great amount of generalization.

TABLE I. MICRODATA OF PATIENT DATABASE

Patient Id	Zipcode	Age	Nationality	Disease
1	13053	28	Russian	Heart Disease
2	13068	29	American	Heart Disease
3	13068	21	Japanese	Viral Infection
4	13053	23	American	Viral Infection
5	14853	50	Indian	Cancer
6	14853	55	Russian	Heart Disease
7	14850	47	American	Viral Infection
8	14850	49	American	Viral Infection
9	13053	31	American	Cancer
10	13053	37	Indian	Cancer

The table I shows the details of a patient database with the attributes of patient id, zip code, sex, nationality and disease. When suppression is applied to this table either one or more attributes can be hidden. In the given example nationality of all the persons are hidden and zip code is partially hidden.

Hence, when any unauthorized user is trying to access the suppressed data shown in table II, they would not be able to identify the person since nationality of a person has completely hidden. The table II has implemented generalization too. The age of a patient has made generalized to certain range such as < 30 rather than 28 or 29 so that an adversary would not be able to identify the age of a patient.

TABLE II. K- Anonymity Generalized and Suppressed Data of Patient Database

Patient id	Zipcode	Age	Nationality	Disease
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

III. Problem Description

The goal of data mining is to extract hidden or useful unknown interesting rules or patterns from databases. However, the objective of privacy preserving data mining is to hide certain confidential data so that they cannot be discovered through data mining techniques. In this paper, it is assumed that only sensitive items are given and propose one algorithm to modify data in database so that sensitive items cannot be deduced through association rules mining algorithms. More specifically, given a transaction database D, a minimum support, a minimum confidence and a set of items H to be hidden, the objective is to modify the database D such that no association rules containing H on the right hand side or left hand side will be discovered.

Association rule hiding refers to the process of modifying the original database in such a way that certain sensitive association rules disappear without seriously affecting the data and the non sensitive rules. Association rule mining is defined as: Let be a set of n binary attributes called items. Let be a set of transactions called the database. Each transaction in D has a unique transaction ID and contains a subset of the items in I. A rule is defined as an implication of the form X→Y where and the sets of items (for short item sets) X and Y are called antecedent (left-hand-side or LHS) and consequent (right hand side or RHS) of the rule respectively. For example T = {T1, T2, T3, T4, T5}. I= {crème, sugar, coffee, beer, bread, chips, cheese, milk, oranges, apples, eggs}.

Support measure of X is denoted as Support(X).

$$Support(X) = (Support\ count(X)/n) * 100$$

The confidence of a rule is defined

$$\text{conf}(X \Rightarrow Y) = \text{supp}(X \cup Y)/\text{supp}(X)$$

A. Association Rule Hiding Framework

The Fig. 2 shows the framework of the approach that consists of six processes. Initially, indexing is performed in the database. Then association rules are mined from the database. Sensitive items are identified to find the sensitive rules. Then the sensitive rules are generated. Clustering is performed on the sensitive rules to group the similar items. The rule hiding process is performed and the transactions are updated in the transaction table and finally it is updated in the original database. The main challenge of rule hiding is how to select the items and transactions to modify. The proposed framework hides the sensitive rules.

The main approached of association rule hiding algorithms to hide some generated association rules, by increase or decrease the support or the confidence of the rules. The association rule items whether in Left Hand Side (LHS) or Right Hand Side (RHS) of the generated rule, that cannot be deduced through association rule mining algorithm.

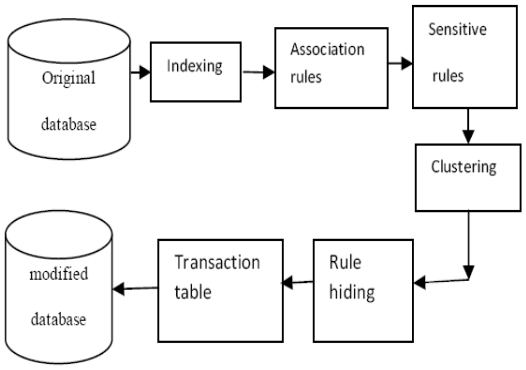

Figure 2: Framework of Association Rule Hiding

IV. Proposed Algorithm

In order to preserve the privacy of the client in data mining process, a variety of techniques based on k-anonymity of data records have been proposed recently. The proposed association rule hiding (ARH) algorithm has reduced information loss by means of hiding those transactions that supports the specific sensitive rule. Then, this algorithm assigns priority for each transaction and sorts the transactions in ascending order according to the priority value of each rule. Once if a transaction has a high priority value such as 1, 2, 3 and so on could be there at top of the table. The existing k-anonymity method either suppresses or generalizes the records to protect the data from unauthorized user. During this process for each 10% of data are suppressed, there could be 0.5% of data loss has happened. Whereas, in the proposed algorithm hiding those transactions that supports the specific sensitive rule rather than hiding all the transactions in the k-anonymity method.

In the proposed algorithm, a sensitive knowledge which can be mined from the database is hidden while non-sensitive knowledge can still be mined. The rule hiding research focuses on association rule hiding and frequent item set hiding. It refers to the process of modifying the original database in such a way that certain sensitive association rules or frequent item sets disappear without seriously affecting the data and non-sensitive rules or item sets. The most wide ranging survey about association rule hiding divides association rule hiding methods as heuristic, border based and exact approaches.

This algorithm requires two database scans. At first scan the inverted file index is created and at second scan items are deleted from selected transactions. The idea behind this approach is that sometimes replacing false values may have bad consequences. The aim of the algorithm is to hide given sensitive rules by replacing unknown values and minimize side effects on non-sensitive rules.

The classical association rule hiding operates on a set of transactions, each composed of a set of items, and produce association rules of the form X → Y, where X and Y are sets of items. Intuitively, rule X → Y expresses the fact that transactions that contain items X tend to also contain items Y. Each rule has a support and a confidence, in the form of percentage. The support expresses the percentage of transactions that contain both X and Y , while the confidence expresses the percentage of transactions, among those containing X, that also contain Y. Since the goal is to find common patterns, typically only those rules that have support and confidence greater than some predefined thresholds are considered of interest.

TABLE III. ASSOCIATION RULE HIDING ALGORITHM

```
INPUT: Original Database D, minimum support minsup,
List of  sensitive itemsets Ls
OUTPUT: Sanitized Database Ds, Frequent itemsets FIs of
Ds
BEGIN
P1 Read D and find frequent items // first scan of database
P2 Read D and build MFI (Maximum Frequent Itemset),
STE (Set of Frequent Set) and TList (Transaction List)
P3 Modify MFI //speeding-up the frequent pattern search
1       FOR every itemset in Ls
2       Calculate support of the sensitive itemset Is
2       Number of iterations:=(Support of Is –minsup) *
        number of  transactions in TList +1
4       FOR 1 to Number of iterations
5       Select pattern from MFI (shortest or longest one)
6       Select transaction from TList
7       Select item to distort (most frequent MaxFI or least
        frequent MinFI in Is)
8       Distort item in D
9       Update MFI
10      Update STE
11      Update TList
12      END
13      END
14      Find frequent itemsets FIs using up to date MFI
15      Return Ds, FIs
END
```

The Association Rules can be implemented in data mining using association rule mining and association rule hiding. Association rule mining is defined as: Let be a set of n binary attributes called items. Let be a set of transactions called the database. Each transaction in D has a unique transaction ID and contains a subset of the items in I. A rule is defined as an implication of the form X→Y where and .The sets of items (for short item sets) X and Y are called antecedent (left-hand-side or LHS) and consequent (right-hand side or RHS) of the rule respectively. Association rule hiding refers to the process of modifying the original database in such a way that certain sensitive association rules disappear without seriously affecting the data and the non sensitive rules.

There are two strategies for transaction selection. First one is to find shortest pattern in Maximum Frequent Itemset (MFI) that includes the sensitive itemset and select the last transaction from Transaction List (TList). Second strategy is to find the longest pattern and select the transaction from TList. Most heuristic approaches use transaction length to decide transaction to modify. However, compact matrix structure of proposed approach has more valuable information as patterns of frequent items so length of the pattern is used instead of length of the transaction. This approach also eliminates the need for database access in choosing decision.

The Table III shows an Association Rule Hiding Algorithm. In this, for every itemset in sensitive itemsets list Ls, hiding process is run. Support value for sensitive itemset is calculated using MFI and Set of Frequent Set (STE). If the support of the itemset is above minsup then the number of iterations to hide itemset is calculated. This number indicates number of distortions to be done on the dataset to reduce the support of the sensitive itemset is below minsup. Folowing this, at each iteration transaction to be modified is selected.

A. Simulations and Performance Evaluation

The major objective of this paper is to investigate the performance of ARH has evaluated in terms of data quality, efficiency, and scalability. To accurately evaluate this approach, the system also compared the implementation with k- anonymity method.

In this section, two synthetic databases are used to see effect of different database size. The algorithms are executed on databases i) to see effect of increasing number of sensitive itemsets, ii) to see effect of increasing support of sensitive itemset and decreased data loss .The effects observed are number of lost itemsets as side effect, time cost for hiding process and number of items distorted for hiding itemsets. During evaluations, it is ensured that the system state is similar for all test runs and results are checked for consistency.

B. Simulation Environment

The proposed system has performed all experiments on a PC with Association Rule Mining And Deduction Analysis (ARMADA) tool (James Malone et al. 2008). Synthetic databases used in the evaluations are generated using ARMADA tool.

Two different databases are used for evaluations in the number of transactions since the algorithm needs to compare the effects of the size of the database on hiding process. One database has 500 transactions while number

of items are 50 and average length of transaction is 5. Other database has 1000 transactions while number of items are 50 and average length of transactions is 5. Minimum support is defined as 3.5% for all evaluations and if no hiding is applied then 293 frequent item sets from 500 transactions database and 586 frequent item sets from 1000 transactions database can be found.

C. Increasing Number of Sensitive Itemsets

For both databases five of length three itemsets which are closest to 3.0% support are selected as sensitive itemsets. These itemsets are given in Table IV. Selected itemsets are mutual exclusive to ensure that one is not affected by hiding process of previous itemsets. The aim of this research is to understand the effect of increasing the number of sensitive itemsets on itemset hiding and the effect of data loss. For each run next item set in the Table IV added to the sensitive itemsets given to program. At first run item set number 1 is given as sensitive, at second run itemset number 1 and itemset number 2 are given as sensitive and so on.

TABLE IV. SUPPORT OF 500 AND 1000 TRANSACTION DATABASE

Itemset number	Itemsets for 500 database	Support level (%)	Itemsets for 1000 database	Support level (%)
1	16	2.96	31	3.00
2	7	3.06	13	3.01
3	6	2.92	14	2.93
4	13	3.08	27	3.09
5	20	2.94	41	2.95

The side effect which is the number of lost itemsets for increasing number of sensitive itemsets is given in the Table IV for 500 transaction database. The Fig. 3 shows the side effects while increasing number of sensitive item sets for 500 transaction database. The Fig.4 shows the side effects while increasing number of sensitive item sets for 1000 transaction database. It is clear that the number of lost itemset has decreased by 8.5% for ARH due to the increase in MFI compared to the existing system.

Time cost for item set hiding is given in Fig. 5 and Fig. 6 for 500 transaction database and 1000 transaction database respectively. Selecting shortest pattern seems as a better method no matter maximum or minimum frequent item is selected for distortion. The database size is doubled and time needed for hiding item sets is increased more than 100%.

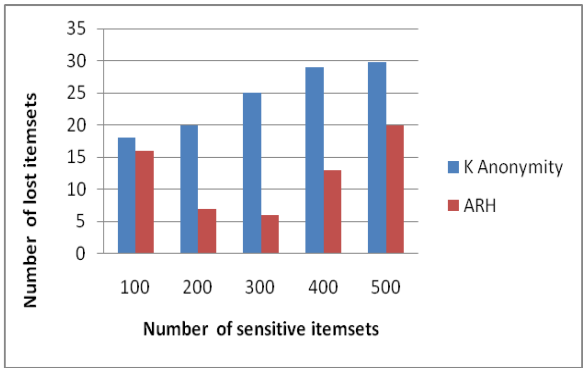

Figure 3: Performance comparison of ARH with K-Anonymity for lost itemsets for 500 transaction database

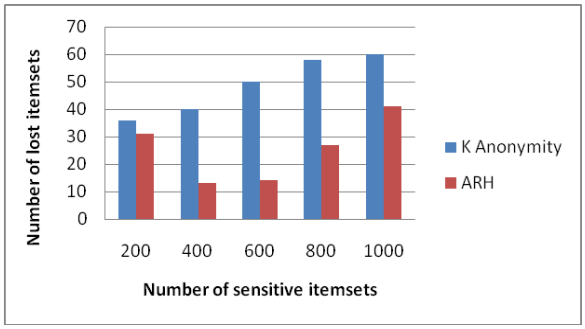

Figure 4: Performance comparison of ARH with K-Anonymity for lost itemsets for 1000 transaction database

The reason behind this is the cost of travelling on matrix to select pattern. It is clear that matrix size is bigger for 1000 transaction database compared to 500 transaction databases.

TABLE V. TIME TO HIDE ITEMSETS FOR 500 TRANSACTION DATABASE

Item set	Time to Hide K- Anonymity	Time to Hide ARH (ms)	% of Time to hide improved
100	13	9	30
200	22	14	36
300	35	21	40
400	43	25	41
500	55	31	43

The Performance comparison of ARH with K-Anonymity for time to hide itemset for 500 transaction database is shown in Fig. 3. The time to hide itemsets of ARH has reduced for 100, 200, 300, 400, 500 transactions as 30%, 36%, 40%, 41% and 43% due to the increase in minimum support. It is clear that the proposed ARH has improved the performance compared to k-anonymity method.

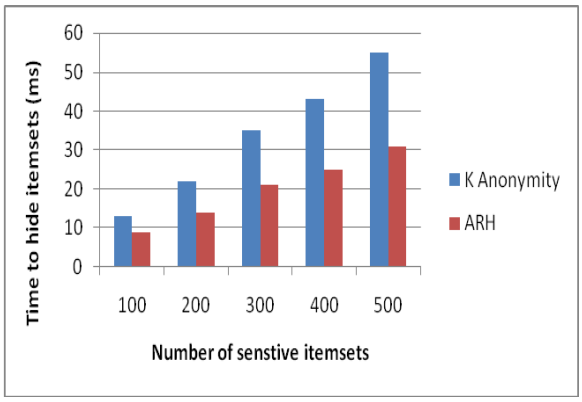

Figure 5: Performance comparison of ARH with K-Anonymity for time to hide itemset for 500 transaction database

Time cost for item set hiding is shown in Table VI for 1000 transaction database. Time to hide itemsets of ARH has reduced for 200, 400, 600, 800, 1000 transactions as 30%, 34%, 41%, 43% and 47% due to the increase in minimum support.

TABLE VI. TIME TO HIDE ITEMSETS FOR 1000 TRANSACTION DATABASE

Item set	Time to Hide K- Anonymity (ms)	Time to Hide ARH (ms)	% of Time to hide improved
200	27	19	30
400	44	29	34
600	71	42	41
800	85	49	43
1000	112	61	47

Time cost for item set hiding is shown in Fig. 6 for 1000 transaction database. Time to hide itemsets for ARH has reduced for 200, 400, 600, 800, 1000 transactions as 30%, 34%, 41%, 43% and 47% due to the increase in minimum support.

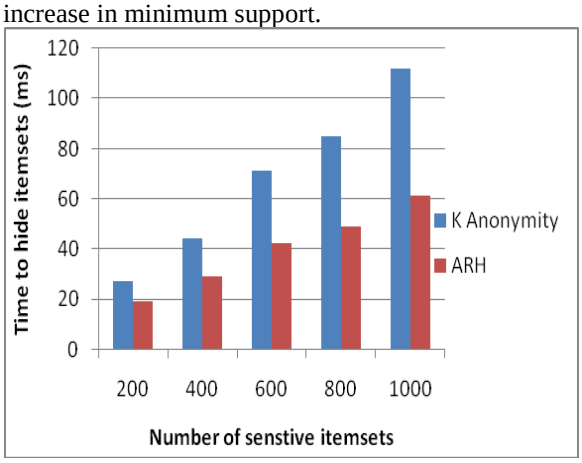

Figure 6: Performance comparison of ARH with K-Anonymity for time to hide itemset for 1000 transaction database

V. CONCLUSION

The proposed ARH algorithm is developed for hiding those transactions which support the specific sensitive rule rather than suppressing all transactions in the k-anonymity method. The performance of ARH for 500 and 1000 transactions database has compared to the existing method k- anonymity. The ARH method has decreased the lost of data and time to hide item sets compared with the k-anonymity method. The proposed algorithm has reduced the overall 21.35% of item set loss for both 500 and 1000 transaction database compared to k- anonymity.

REFERENCES

[1] Sweeney L., 'Achieving k-anonymity privacy protection using generalization and suppression', International Journal of Uncertainty, Fuzziness and Knowledge Based Systems, Vol.10, pp. 571-588, 2007.

[2] Samarati P , 'Protecting respondents' identities in microdata release', IEEE Transactions on Knowledge and Data Engineering, Vol.13(6), pp. 1010-1027, 2001.

[3] Saygin, Y., V.S. Verykios and A.K. Elmagarmid, 'Privacy preserving association rule mining', Proceedings of the 12th International Workshop on Research Issues in Data Engineering: pp. 151-158, 2002.

[4] Vaidya, J., H. Yu and X. Jiang, 'Privacy preserving SVM classification', Journal Knowledge and Information Systems, pp. 161-178, 2008.

[5] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke and Muthuramakrishnan, 'l-diversity: Privacy beyond k-anonymity', Proceedings of the International Conference on Data Engineering, pp. 24-39, 2007.

[6] Dimitris Sacharidis, Kyriakos Mouratidis and Dimitris Papadias, 'k-Anonymity in the Presence of External Databases', IEEE Transactions on Knowledge and Data Engineering, Vol. 22, pp. 392-404, 2010.

[7] Friedman A., Wolff R. and Schuster A., 'Providing k-Anonymity in Data Mining', International Journal of Very Large Data Bases, Vol. 17, pp. 789-804, 2008.

[8] Jajodia S., Yao C and Wang XS, 'Checking for k-anonymity violation by views', Proceedings of the 31st International Conference on Very Large Data Bases, pp.167-172, 2005.

[9] Mannila H., Toivonen H., and Verkamo A. I., 'Efficient algorithms for discovering association rules', Proceedings of the Workshop on Knowledge Discovery in Databases, pp. 181-92,2006.

[10] Mannila H., Toivonen H., and Verkamo A. I., 'Efficient algorithms for discovering association rules', Proceedings of the Workshop on Knowledge Discovery in Databases, pp. 181-92, 2006.

[11] Nan Zhang, Shengquan Wang and Wei Zhao, 'A New Scheme on Privacy Preserving Association Rule Mining', Proceedings of the 2nd European Conference on Principles of Data Mining and Knowledge Discovery, pp. 484-495, 2008.

[12] Oliveira S. R. M., Zaiane O.R. and Saygin Y., 'Secure association rule sharing, advances in knowledge discovery and data mining', Proceedings of the 8th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, pp.157-168, 2007.

[13] Shan-Tai Chen, Shih-Min Lin, Chi-Yii Tang, and Guei-Yu Lin, 'An Improved Algorithm for Completely Hiding Sensitive Association Rule Sets', Proceedings of the 2nd IEEE International Conference on Computer Science and its Applications, pp.1-6, 2011.

[14] Shaofei Wu and Hui Wang, 'Research On The Privacy Preserving Algorithm Of Association Rule Mining in Centralized Database', Proceedings of the International Symposiums on Information Processing, pp. 131-134, 2008.

[15] Shyue Liang, Wang YuHuei, Lee Billis and Jafari S., 'Hiding Sensitive items in Privacy Preserving Association rule Mining', Proceedings of the International Systems, Man and Cybernetics, pp.3239-3244, 2006.

[16] Vassilios S. Verykios, Emmanuel D and Pontikakis, 'Efficient algorithms for distortion and blocking techniques in association rule hiding', Journal Distributed and Parallel Databases, pp.212-236, 2007.

[17] Verykios V. and Gkoulalas Divanis, 'A survey of association rule hiding methods for privacy', Journal of Database and Expert Systems Applications, pp. 267-289, 2008.

[18] Verykios, V.S., Elmagarmid A.K., Bertino E., Saygin Y. and Dasseni E., 'Association rule hiding', IEEE Transactions on Knowledge and Data Engineering, Vol. 16, pp. 434-447, 2004.

[19] Wu Y.H., Chiang C.M. and Chen A.L.C., 'Hiding Sensitive Association Rules with Limited Side Effects', IEEE Transaction on Knowledge Data Engineering, Vol.19, pp.29-42, 2007.

[20] Xiangmin Ren and Jing Yang, 'Research on Privacy Protection Based on K-Anonymity', Proceedings of the International Conference on Biomedical Engineering and Computer Science, pp. 1-5, 2010.

[21] Xiao X and Tao Y, 'Personalized privacy preservation", Proceedings of the ACM Conference on management of Data', ACM Press, New York, pp.785-790, 2006.

[22] Xiaojun Ye, Yawei Zhang and Ming Liu, 'A Personalized (α, k)-Anonymity Mode', Proceedings of the IEEE International Conference on 9th International Conference on Web-Age Information Management, pp.122-131, 2009.

[23] Xiaoming Zhang and Xi Qiao, 'New Approach for Sensitive Association Rule Hiding', Proceedings of the International Workshop on Education Technology and Training, pp.174-181, 2008.

[24] Xinyi Huang and Yang Xiang, 'A Generic Framework for Three-Factor Authentication: Preserving Security and Privacy in Distributed Systems', IEEE Transactions on Parallel and Distributed Systems, Vol. 22, pp.1390-1398, 2011.

[25] Modi C.N., Rao U.P. and Patel D.R., 'Maintaining privacy and data quality in privacy preserving association rule mining', Proceedings of the International Conference on Computing Communication and Networking Technologies, pp. 1-6, 2010.

[26] Pingshui Wang, 'Survey on Privacy Preserving Data Mining', International Journal of Digital Content Technology and its Application, Vol. 4, pp.142-149, 2010.

[27] Poovammal E.and Ponnavaikko, 'An Improved Method for Privacy Preserving Data Mining', Proceedings of the IEEE International Conference on Advance Computing, pp.143-52, 2009.

[28] Samarati P. and Sweeney L., 'Protecting Privacy When Disclosing Information: Kanonymity and its Enforcement Through Generalization and Suppression', Proceedings of the IEEE Symposium on Research in Security and Privacy, pp.54-82, 2009.

R. Sugumar received his Undergraduate Degree in Computer Science and Engineering from Madras University, in 2003 and the Post Graduate degree in Computer Science and Engineering from Dr.M.G.R. Educational and Research Intituite, Chennai in 2007 and Ph.D in Computer Science and Engineering at Bharath University, Chennai during 2011. He has more than 15 publications in National Conferences and international journal proceedings. He has more than 8 years of teaching experience. His areas of interest includes Data Mining, Data Structures, DBMS, Distributed systems and OS.

A.Rengarajan received the Undergraduate Degree in Computer Science and Engineering from Madurai Kamaraj University, in 2000 and the Post Graduate degree in Computer Science and Engineering from Sathyabama University, Chennai in 2005 and Ph.D in Computer Science and Engineering at Bharath University, Chennai during 2011. He has more than 15 publications in National Conferences and international journal proceedings. He has more than 12 years of teaching experience. His areas of interest include Network security, Mobile computing, Data Structures, DBMS, Distributed systems and Operating systems.

M.Vijay Anand has more than 14 years of teaching and research experience. He did his Postgraduate in ME in Computer Science and Engineering at sathyabama university during 2005. He is pursuing Ph.D in Computer Science and Engineering at Anna University, Chennai. He has published more than 10 research papers in various conference and journal proceedings. His areas of interest include Mobile computing, Data mining, DBMS, Computer Networks and Operating systems.

