Association Rule Hiding by Heuristic Approach to Reduce Side Effects & Hide Multiple R.H.S. Items

Komal Shah
U and P.U. Patel Department of Computer Engineering,
Chandubhai S Patel Institute of Technology,
Charotar University of Science and Technology,
Changa, Gujarat, India

Amit Thakkar
Department of Information Technology,
Chandubhai S Patel Institute of Technology,
Charotar University of Science and Technology,
Changa, Gujarat, India

Amit Ganatra
U and P.U. Patel Department of Computer Engineering,
Chandubhai S Patel Institute of Technology,
Charotar University of Science and Technology,
Changa, Gujarat, India

ABSTRACT
Association rule mining is a powerful model of data mining used for finding hidden patterns in large databases. One of the great challenges of data mining is to protect the confidentiality of sensitive patterns when releasing database to third parties. Association rule hiding algorithms sanitize database such that certain sensitive association rules cannot be discovered through association rule mining techniques. In this study, we propose two algorithms, ADSRRC (Advanced Decrease Support of R.H.S. items of Rule Cluster) and RRLR (Remove and Reinsert L.H.S. of Rule), for hiding sensitive association rules. Both algorithms are developed to overcome limitations of existing rule hiding algorithm DSRRC (Decrease Support of R.H.S. items of Rule Cluster). Algorithm ADSRRC overcomes limitation of multiple sorting in database as well as it selects transaction to be modified based on different criteria than DSRRC algorithm. Algorithm RRLR overcomes limitation of hiding rules having multiple R.H.S. items. Experimental results show that both proposed algorithms outperform DSRRC in terms of side effects generated and data quality in most cases.

General Terms
Database, Data mining, Security.

Keywords
Association Rule Hiding, Data Mining, Privacy Preservation Data Mining.

1. INTRODUCTION
Data mining technology aims to find useful patterns from large amount of data. These patterns represent knowledge and are expressed in decision trees, clusters or association rules. The knowledge discovered by various data mining techniques may contain private information about individual or business. Revelation of any private information may cause threat to security. For example, in medical database, it is useful to share information about diseases but at the same time it is required to preserve patient's identity. Here individual privacy must be maintained. Another example is market basket database which is used to analyze customer's purchasing behavior represented in terms of association rules. In market basket database, instead of data related to individuals, the sensitive information or knowledge derived from data is required to be protected.

Privacy preservation data mining (PPDM) considers problem of maintaining privacy in data mining. PPDM algorithms are developed for modifying the original data in such that sensitive data and knowledge remains unrevealed even after the mining process. Association rule hiding is one of the privacy preservation techniques to hide sensitive association rules. All association rule hiding algorithm aims to minimally modify the original database such that no sensitive association rule is derived from it.

In this paper we have proposed two association rule hiding algorithms, ADSRRC (Advanced Decrease Support of R.H.S. items of Rule Cluster) and RRLR (Remove and Reinsert L.H.S. of Rule), based on heuristic approach. Both algorithms are based on algorithm DSRRC proposed in [12]. Algorithm DSRRC depends on ordering of transactions for removing items from database. Also it requires sorting of database each time item is removed from database. Algorithm ADSRRC is proposed to overcome these limitations. Algorithm DSRRC cannot hide rule having multiple R.H.S. items. To overcome this limitation algorithm RRLR is proposed.

2. THEORETICAL BACKGROUND AND RELATED WORK
Privacy is defined as "The right of an entity to be secure from unauthorized disclosure of sensible information that are contained in an electronic repository or that can be derived as aggregate and complex information from data stored in an electronic repository"[2]. PPDM is to conduct data mining operations under the condition of preserving data privacy [3]. PPDM investigates the side effects of data mining methods that originate from the penetration into the privacy of individuals and organizations [4]. The aim of PPDM algorithms is to extract relevant knowledge from large amounts of data while protecting at the same time sensitive information [1].

The aim of association rule hiding algorithms is to properly sanitize the original data so that any association rule mining algorithms that may be applied to the sanitized version of the data (i) will be incapable to uncover the sensitive rules under certain parameter settings, and (ii) will be able to mine all the non-sensitive rules that appeared in the original dataset (under the same or higher parameter settings) and no new rules generated. There are mainly 3 approaches for Association Rule Hiding (i) Heuristic Approach (ii) Border Based Approach (iii) Exact Approach [1]. In following, overview of these approaches is given in brief.

2.1 Heuristic approach

This approach involves efficient, fast and scalable algorithms that selectively sanitize a set of transactions from the original database to hide the sensitive association rules [1]. Various heuristic algorithms are based on mainly two techniques: Data distortion technique and blocking technique.

Data distortion is done by the alteration of an attribute value by a new value. It changes 1's to 0's or vice versa in selected transactions to increase or decrease support or confidence of sensitive rule. Heuristic algorithms cannot give an optimal solution because of undesirable side effects to nonsensitive rules, e.g. lost rules and new rule.

Algorithms proposed using heuristic approach can be divided into rule hiding and itemset hiding algorithms [6]. Five algorithms are proposed to hide sensitive information in database [6], among them three are rule hiding algorithms and two are itemset hiding algorithms. Later on itemset hiding algorithms to automatically hide sensitive rules without pre mining and selection of rules are proposed in [7] and [8]. These algorithms increase the support of L.H.S. of the rule or decrease the support of the R.H.S. of the rule by inserting and removing sensitive items from selected transactions respectively. An item set hiding algorithm which uses pattern-inversion tree is proposed in [9] to store related information so that only one scan of database is required. In [10] four heuristic algorithms are proposed which selects the sensitive transactions to sanitize based on degree of conflict and then removes items from selected transactions based on certain criteria like remove all items except item with highest frequency, remove item having smallest support, remove item with the maximum support or removes group of items sharing shame patterns. A rule hiding algorithm is proposed in [11], which correlates sensitive association rules and transactions by using a graph to effectively select the proper item for modification. Later on in [12], a rule hiding algorithm named DSRRC (Decrease Support of R.H.S. item of Rule Clusters) is proposed, which clusters the sensitive association rules based on R.H.S. of rules and hides as many as possible rules at a time by modifying fewer transactions. Because of less modification in database it helps maintaining data quality, but this algorithm cannot hide rules having multiple R.H.S. items and also it exhibits undesirable side effects.

Blocking is the replacement of an existing value with a "?". It inserts unknown values in the data to fuzzify the rules. In some applications where publishing wrong data is not acceptable, then unknown values may be inserted to blur the rules. In [13] two algorithms are built based on blocking for rule hiding. The first one focuses on hiding the rules by reducing the minimum support of the itemsets that generated these rules (i.e., generating itemsets). The second one focuses on reducing the minimum confidence of the rules.

2.2 Border based approach

This approach hides sensitive association rule by modifying the borders in the lattice of the frequent and the infrequent itemsets of the original database. These algorithms use the theory of borders presented in [14]. The first frequent itemset hiding methodology that is based on the notion of the border is proposed in [15]. It maintains the quality of database by greedily selecting the modifications with minimal side effect.

2.3 Exact approach

This approach contains nonheuristic algorithms which formulates the hiding process as a constraints satisfaction problem or an optimization problem which is solved by integer programming. These algorithms can provide optimal hiding solution with ideally no side effects. An exact algorithm for association rule hiding is proposed in [16] which tries to minimize the distance between the original database and its sanitized version. In [17] proposed an exact border based approach to achieve optimal solution as compared to previous approaches.

The rest of this paper is organized as follows. In section 3, we discuss association rule mining strategy and problem of association rule hiding. We also discuss limitations of algorithm DSRRC. In section 4, a detailed description of proposed ADSRRC and example demonstrating ADSRRC is given. Section 5 contains detailed description of RRLR algorithm and example demonstrating this algorithm. In section 6 we analyze and discuss the performance results of proposed algorithms. Finally in section 7, we conclude by defining some future enhancements.

3. PROBLEM DEFINITION

Let $I = \{i_1, \ldots, i_n\}$ be a set of items. Let D be a set of transactions or database. Each transaction $t \in D$ is an item set such that t is a proper subset of I. A transaction t supports X, a set of items in I, if X is a proper subset of t. Assume that the items in a transaction or an item set are sorted in lexicographic order.

An association rule is an implication of the form $X \rightarrow Y$, where X and Y are subsets of I and $X \cap Y = \emptyset$. The support of rule $X \rightarrow Y$ can be computed by the following equation: $Support(X \rightarrow Y) = |X \rightarrow Y| / |D|$, where $|X \rightarrow Y|$ denotes the number of transactions in the database that contains the itemset XY, and |D| denotes the number of the transactions in the database D. The confidence of rule is calculated by following equation: $Confidence(X \rightarrow Y) = |X \rightarrow Y| / |X|$, where |X| is number of transactions in database D that contains itemset X. A rule $X \rightarrow Y$ is strong if $support(X \rightarrow Y) \geq$ min_support and $confidence(X \rightarrow Y) \geq$ min_confidence, where min_support and min_confidence are two given minimum thresholds. Association rule mining algorithms scan the database of transactions and calculate the support and confidence of the rules and retrieve only those rules having support and confidence higher than the user specified minimum support and confidence threshold.

Association rule hiding algorithms prevents the sensitive rules from being disclosed. The problem can be stated as follows: "Given a transactional database D, minimum confidence, minimum support and a set R of rules mined from database D. A subset R_H of R is denoted as set of sensitive association rules which are to be hidden. The objective is to transform D into a database D' in such a way that no association rule in R_H will be mined and all non sensitive rules in R could still be mined from D'. The problem for finding an optimal sanitization to a database against association rule analysis has been proven to be NP-Hard [5].

For example, a sample transactional database D is shown in Table I. TID shows unique transaction number. Suppose MST and MCT are selected 3 and 75% respectively. The association rules satisfying MST and MCT, generated by apriori algorithm are $b \rightarrow a$, $a \rightarrow d$, $d \rightarrow a$, $b \rightarrow d$, $c \rightarrow d$, $d \rightarrow c$, $e \rightarrow c$, $e \rightarrow d$, $ab \rightarrow d$, $bd \rightarrow a$, $ac \rightarrow d$, $ce \rightarrow d$, $de \rightarrow c$, and $e \rightarrow dc$. Suppose the rules $b \rightarrow a$, $b \rightarrow d$ and $c \rightarrow d$ specified as sensitive and should be hidden in sanitized database.

In order to hide an association rule, we can either decrease its support or its confidence to be smaller than pre-specified

minimum support and minimum confidence threshold. To decrease the support of rule X→Y, we can decrease the support of corresponding large itemset XY. To decrease the confidence of rule X→Y, we can either increase support of X (i.e. L.H.S.) in transactions not supporting Y or we can decrease the support of Y (i.e. R.H.S.) in transactions supporting X and Y both. In algorithm DSRRC, authors decrease confidence of rule by removing R.H.S. of rule from selected transactions. To remove R.H.S. authors change value of an itemset from '1' to '0'. They also use following concept of sensitivities specified in [12].

1) Item Sensitivity is the frequency of data item exists in the number of the sensitive association rule containing this item. It is used to measure rule sensitivity.

2) *Rule Sensitivity* is the sum of the sensitivities of all items containing that association rule.

3) Cluster Sensitivity is the sum of the sensitivities of all association rules in cluster. Cluster sensitivity defines the rule cluster which is most affecting to the privacy.

4) Sensitive Transaction is the transaction in given database which contains sensitive item.

5) Transaction sensitivity is the sum of sensitivities of sensitive items contained in the transaction.

They clusters the sensitive rules based on R.H.S. So in this example two clusters are made as shown in Table II. It also shows sensitivities of each item in cluster, sensitivity of each cluster and sensitivities of each transaction with respect to both clusters. C1 and C2 represent sensitivities of transaction with respect to cluster 1and 2 respectively. They first sort clusters in decreasing order of sensitivities and choose highest sensitive cluster first. Then transactions are sorted in decreasing order of their sensitivities for that cluster. Then highest sensitive transaction is selected for modification and removes the R.H.S. item of cluster from that transaction. Each time an item is modified, sensitivities are updated and transactions are sorted in decreasing order of their sensitivities. Thus for large database, it will affect the running time of algorithm. This process continues until all sensitive rules in each cluster are hidden. Final sanitized database is shown in Table III. In final sanitized database, all sensitive rules are successfully hidden (i.e. 0% Hiding Failure), almost 36% of the rules are missing (i.e. Misses Cost) and no artifactual rules generated and only one transaction is modified.

Table I. Sample Database

TID	Items
1	a b c d e
2	a c d
3	a b d f g
4	b c d e
5	a b d
6	c d e f h
7	a b c g
8	a c d e
9	a c d h

Table II. Cluster and Transaction Sensitivities

Cluster-1 RHS: d Rules:b→d,c→d Sensitivity :4		Cluster-2 RHS: a Rules: b→a Sensitivity :2	
Item	**Sensitivity**	**Item**	**Sensitivity**
b	1	b	1
c	1	a	1
d	2		

TID	Items	C1	C2
1	a b c d e	4	2
2	a c d	3	-
3	a b d f g	3	2
4	b c d e	4	-
5	a b d	3	2
6	c d e f h	3	-
7	a b c g	-	2
8	a c d e	3	-
9	a c d h	3	-

Table III. Sanitized Database by DSRRC Algorithm

TID	Items
1	b c e
2	a c d
3	a b d f g
4	b c d e
5	a b d
6	c d e f h
7	a b c g
8	a c d e
9	a c d h

Table IV. Modified Database By Interchanging Rows Of Table I

TID	Items
1	b c d e
2	a b d
3	a c d
4	a b c g
5	a c d e
6	a b d f g
7	c d e f h
8	a b c d e
9	a c d h

Table V. Transaction Sensitivities and Final Sanitized Database

TID	Items	C1	C2
1	b c d e	4	-
2	a b d	3	2
3	a c d	3	-
4	a b c g	-	2
5	a c d e	3	-
6	a b d f g	3	2
7	c d e f h	3	-
8	a b c d e	4	2
9	a c d h	3	-

TID	Items
1	b c e
2	b d
3	a c d
4	a b c g
5	a c d e
6	a b d f g
7	c d e f h
8	a b c d e
9	a c d h

It is observed that algorithm DSRRC is sensitive to the order of transaction in the database. For example if we interchange any two or more rows in database then it gives different result on same database. Consider original database shown in Table I and modified database by interchanging rows shown in Table IV. According to DSRRC algorithm two clusters are generated as shown in Table II. Then transactions are indexed

as per sensitivities as shown in Table V. Because this algorithm selects transactions sequentially, for modified database, two transactions are updated. So it is analyzed that DSRRC is dependent on ordering of records in database. Ordering of rows greatly impacts the resultant modified database shown in Table V. In final sanitized database shown in Table V, there is 0% hiding failure and 36% Misses Cost and 27.28% of new rules are created as artifactual patterns. Also two transactions are modified in database. For same database, algorithm DSRRC exhibits different results if we simply interchange some of the transactions. Transaction selection method of DSRRC cannot select same transaction to modify for different instances of same database. Another limitation of algorithm DSRRC is that cannot hide rule having multiple RHS items. It hides only rules having single consequent.

In next sections we are proposing two different algorithms ADSRRC and RRLR. ADSRRC overcomes the problem of transaction dependency by employing a different technique of transaction selection. Algorithm RRLR is extension of DSRRC such that it can hide rules multiple R.H.S. items.

4. PROPOSED ALGORITHM – ADSRRC

ADSRRC is based on concept of sensitivities in given in [12]. Initially association rules are mined from the source database by using association rule mining algorithms e.g. Apriori algorithm. Then sensitive rules are specified from mined rules. Selected rules are clustered based on common R.H.S. item of the rules.

Then transactions are indexed by sensitivities. In DSRRC transaction sensitivity is different for each cluster but ADSRRC calculates transaction sensitivity irrespective of clusters. It means for all clusters transaction sensitivity is same.

After transaction indexing, ADSRRC sorts the transactions based on sensitivity. In DSRRC, each time item is removed, transactions are sorted. But in ADSRRC, first transactions are sorted in decreasing order of their sensitivity. Then transactions having same sensitivity are sorted in decreasing order of their length. When an item is removed from any transaction, sensitivity is not modified. Thus transactions are sorted only two times. Thus we are avoiding here multiple sorting of transactions which will significantly reduce execution time of algorithm for large database.

After sorting process, rule hiding process, starts by selecting highest sensitive transaction for deleting R.H.S. item. If two transactions have same sensitivity then lengthiest transaction is chosen to be modified. This process continues until all the sensitive rules in all clusters are not hidden. Finally modified transactions are updated in original database and produced database is called sanitized database D' which ensures certain privacy for specified rules and maintains data quality.

Algorithm ADSRRC
INPUT: Source database D, Minimum Confidence Threshold (MCT), Minimum support threshold (MST).
OUTPUT: The sanitized database D'.

1. Begin
2. Generate association rules.
3. Selecting the Sensitive rule set RH with single antecedent and consequent e.g. x→y.
4. Clustering-based on common item in R.H.S. of the selected rules.
5. Find sensitivity of each item in each cluster.
6. Find the sensitivity of each rule in each cluster.
7. Find the sensitivity of each cluster
8. Find sensitivity of each item i∈ D.
9. For each transaction t∈ D
10. {
11. Sensitivity of t = 0;
12. For each item i ∈ t
13. {
14. Sensitivity of t = Sensitivity of t + Sensitivity of i
15. }
16. }
17. Sort transactions in decreasing order of their sensitivity. If two or more transactions have same sensitivity then sort those in decreasing order of their length.
18. Sort generated clusters in decreasing order of their sensitivity.
19. For each cluster c∈ C
20. {
21. While(all the sensitive rules ∈ c are not hidden)
22. {
23. Take first transaction and delete common R.H.S. item from the transaction. If this transaction doesn't contain that item then select next sensitive transaction in order.
24. For i = 1 to no. of rule RH∈c
25. {
26. Update support and confidence of the rule r ∈c.
27. If(support of r < MST or confidence of r < MCT)
28. {
29. Remove Rule r from RH
30. }
31. }
32. Take next transaction.
33. } End while
34. } End for

Fig 1: Algorithm ADSRRC

Fig. 1 shows proposed algorithm ADSRRC. Steps 1 to 3 generates association rule and selects sensitive association rules. Step 4 clusters sensitive rules based on common R.H.S. item. Then in step 5 we are finding sensitivities of each item in cluster. Rule sensitivity and cluster sensitivity is found in steps 6 and 7. In step 8 total sensitivity of each item is found. Step 9 to 16 describes the method to find sensitivities of each transaction. It is calculated by summing sensitivities of all items belongs to that transaction. In step 17, transactions are sorted in decreasing order of their sensitivity. If two or more transactions have same sensitivity then they are sorted in decreasing order of their length. From step 18 onwards, rule hiding process starts. This process is similar as in DSRRC. But it does not modify sensitivity of transaction at each removal of item and it does not sort transactions during rule hiding process. Process of rule hiding continues until all sensitive rules are hidden.

Considering previous example of modified database, shown in Table IV, this algorithm also generates two clusters from selected sensitive rules, shown in Table II. As shown in step 8 of ADSRRC, sensitivity of each item is calculated. For example item 'b' has sensitivity 1 in cluster 1 and sensitivity 1

in cluster 2. So total sensitivity of item b is equals to 2. For transaction sensitivity, sensitivities of each item appearing in that transaction is added. Table VI shows sensitivity for all items in database as well for all transactions and it also shows transaction length. Now transaction with highest sensitivity, which is transaction having TID = 8, is chosen. According to algorithm ADSRRC, for first cluster, item 'd' is removed from this transaction. Now all rules in cluster 1 are hidden. For second cluster again same transaction is chosen and item 'a' is removed from it. This process continues until all sensitive rules in all clusters are hidden. Table VII shows final sanitized database. As compared to DSRRC, only one transaction is modified in final database and no new rule is generated.

Table VI. Item and Transaction Sensitivity

Item	Sensitivity
a	1
b	2
c	1
d	2
e	0
f	0
g	0
h	0

TID	Items	Sensitivity	Length
1	b c d e	5	4
2	a b d	5	3
3	a c d	4	3
4	a b c g	4	4
5	a c d e	4	4
6	a b d f g	4	5
7	c d e f h	3	5
8	a b c d e	6	5
9	a c d h	4	4

Table VII. Final Sanitized Database

TID	Items
1	b c d e
2	a b d
3	a c d
4	a b c g
5	a c d e
6	a b d f g
7	c d e f h
8	b c e
9	a c d h

5. PROPOSED ALGORITHM – RRLR

Concept of sensitivity is also used in RRLR algorithm. This algorithm hides sensitive association rules having multiple RHS items. In algorithm RRLR, to hide sensitive association rule we are decreasing support and confidence both. For rule x→yz, support of rule can be decreased by decreasing support of large itemset 'xyz'. We are applying here LHS deletion process to decrease the support of large itemset 'xyz'. To decrease the confidence of rule x→yz we are using LHS insertion process. In LHS insertion, confidence of rule can be decreased by inserting LHS of rule in transaction not supporting RHS of rule.

Initially association rules are mined from the source database by using association rule mining algorithms e.g. Apriori algorithm. Then sensitive rules are specified from mined rules. Then transaction sensitivity and item sensitivity is found. Transactions are sorted in decreasing order of their sensitivity and length. These all steps are similar to ADSRRC algorithm except that in RRLR we are not creating clusters. After sorting process, rule hiding process hides all the sensitive rules in sorted transactions by using LHS insertion and Deletion Process. It will not update the sensitivity of transactions during rule hiding. Hiding process starts from highest sensitive transaction and continues until all the sensitive rules are not hidden. Finally modified transactions are updated in original database and produced database is called sanitized database which ensures certain privacy for specified rules and maintains data quality.

Algorithm RRLR
INPUT: Source database D, Minimum Confidence Threshold (MCT), Minimum support threshold (MST).
OUTPUT: The sanitized database D'.

1. Begin
2. Generate association rules.
3. Selecting the Sensitive rule set RH with single antecedent and multiple consequent e.g. x→yz
4. Find sensitivity of each item i∈ D.
5. For each transaction t∈ D {
6. Sensitivity of t = 0;
7. For each item i ∈ t {
8. Sensitivity of t = Sensitivity of t + Sensitivity of i
9. }
10. }
11. Sort transactions in decreasing order of their sensitivity. If two or more transactions have same sensitivity then sort those in decreasing order of their length.
12. Sort sensitive rules of RH in decreasing order of their confidence.
13. While (all the sensitive rules are not hidden) {
14. For i=1 to no. of rules ∈ RH {
15. Select rule R_i (for example x→yz) for hiding
16. **LHS Deletion Process**
17. For j = 1 to no. of transactions ∈ D {
18. If (itemset xyz ∈ transaction t_j)
19. Remove LHS of rule R_i from transaction t_j and start LHS Insertion process
20. }
21. **LHS Insertion Process**
22. For k = j to no. of transactions ∈ D {
23. If (RHS of rule (e.g. itemset 'yz') does not belongs to transaction t_k and LHS of rule (item 'x') does not belongs to transaction t_k)
24. Insert LHS of rule R_i in transaction t_k and start modification of support and confidence
25. }
26. **Modification of Support and Confidence**
27. Update support and confidence of all the rule belongs to RH
28. If(support of R_i < MST or confidence of R_i < MCT)
29. Remove Rule R_i from RH
30. Else repeat LHS deletion and insertion
31. }//End for
32. } //End while

Fig 2: Algorithm RRLR

Fig. 2 shows the algorithm of RRLR. In steps 1 to11 we are mining rules, selecting sensitive rules, finding sensitivities of items and transactions as well as sorting transaction is done. Then LHS deletion and insertion process is specified, which is explained by example as following.

Table VIII. Sample Database

TID	Itemset
1	a c d
2	a c d h
3	a b c d e f h
4	a b d f g
5	b c d e
6	a b c
7	c d e f h
8	a b c g
9	b c e f g
10	a c d g
11	b f g
12	a b c d e
13	a c d e f h
14	b g
15	a b c d h

Table IX. Item and Transaction Sensitivities

Item	Sensitivity
a	1
b	0
c	3
d	3
e	1
f	0
g	0
h	1

TID	Itemset	Sensitivity	Length
3	a b c d e f h	9	7
13	a c d e f h	9	6
7	c d e f h	8	5
12	a b c d e	8	5
15	a b c d h	8	5
2	a c d h	8	4
5	b c d e	7	4
10	a c d g	7	4
1	a c d	7	3
4	a b d f g	4	5
9	b c e f g	4	5
8	a b c g	4	4
6	a b c	4	3
11	b f g	0	3
14	b g	0	2

Table X. Final Sanitized Database

TID	Itemset
1	a c d
2	a c d h
3	a b c f
4	a b d e f g h
5	b c d e
6	a b c
7	c d e f h
8	a b c g
9	b c d e f g
10	a c d g
11	b f g
12	a b c d e
13	a c d e f h
14	b g
15	a b c d h

Considering database shown in Table VIII . Following rules are mined from database under support = 50% and confidence = 70% : e→ c, h→c, h→d, de→c, dh→c, ch→d, h→cd, a→c, d→c, ad→c, g→b, e→d, ab→c, ce→d, e→cd, d→a, a→d, cd→a, ac→d, c→a, c→d, bc→a, b→c, d→ac, a→cd. Suppose rules h→cd (confidence=100%), e→cd (confidence =83%) and d→ac (confidence =70%) are sensitive and needs to be removed. First sensitivity of each item is calculated. For example item 'd' is appearing in all sensitive rules. So its total sensitivity is calculated as 1+1+1 = 3. For transaction sensitivity, sensitivities of each item appearing in that transaction is added. Table IX shows item sensitivity and transactions sorted in decreasing order of sensitivity and length.

According to algorithm RRLR all sensitive rules are sorted in decreasing order of their confidence. Then rule having highest confidence is chosen for hiding. So rule h→cd is chosen first for hiding. Transaction having TID = 3 is most sensitive transaction. So it is chosen to be modified. To hide a rule two procedures are developed: LHS Deletion and LHS Insertion. LHS deletion process deletes LHS of sensitive rule from the selected transaction. So item 'h' is deleted from transaction having TID = 3. So support of large itemset 'hcd' is decreased.

After deletion, LHS item is inserted in most sensitive transaction not having large itemset 'cd' and no item 'h'. Thus transaction having TID = 4 is chosen and 'h' is inserted into it. So confidence of h→cd is decreased due to increase in support of 'h' item. After deletion and insertion confidence and support of rule h→cd is modified. The process of deletion and insertion continues until all rules are hidden in sensitive rule set. Table X shows final sanitized database. Now, if we mine association rules from final sanitized database, we can see that all of the sensitive rules are hidden and very few side effects produced.

6. ANALYSIS OF PROPOSED ALGORITHM

For performance comparison, we have selected algorithm DSRRC because both proposed algorithms are based on this algorithm. We used algorithm DSRRC and ADSRRC to sanitize sample database as shown in Table IV and applied apriori algorithm on sanitized database produced by both algorithms to mine association rules. In our experiment, we selected 3 sensitive association rules as in example. The sample dataset has 14 association rules with support count ≥ 3 and confidence ≥ 75. Then we evaluated proposed algorithm with respect to following parameters: hiding failure (HF), misses cost (MC), artifactual patterns (AP), dissimilarity (DISS), and completeness etc. A detailed overview of these parameters is given in [18]. As shown in Table XI, performance of ADSRRC is better than DSRRC, in terms of number of sorting, artifactual patterns, percentage of transaction modified and completeness.

To hide rules having multiple R.H.S. items, we have applied algorithm RRLR on database shown in Table VIII. Then we have applied apriori algorithm on sanitized database produced by RRLR algorithm to mine association rules. Results with respect to different parameters are shown in Table XI. It is shown that performance of RRLR is better than DSRRC, in terms of sorting, misses cost, artifactual patterns, dissimilarity, transaction modified and completeness.

Table XI. Comparative Analysis With DSRRC Algorithm

Parameters	DSRRC	ADSRRC	RRLR
No. of Sorting	> Two	Two	Two
Hiding Failure	0%	0%	0%
Misses Cost	36.36%	36.36%	22.73%
Artifactual Patterns	27.28%	0%	0%
Dissimilarity	5.40%	5.40%	0%
Transaction	22.22%	11.11%	20%
Completeness	77.78%	88.89%	80%

7. CONCLUSION AND FUTURE EXTENSIONS

In this paper, all existing approaches for association rule hiding are briefed. Then two algorithms are proposed based on item heuristic approach. Algorithm ADSRRC is modification of algorithm DSRRC such that side effects and time complexity both are reduced. Algorithm RRLR is proposed for hiding rules having multiple items in RHS. Both algorithms are analyzed with respect to hiding failure, misses cost, artifactual patterns, dissimilarity and completeness by taking suitable examples.

In future algorithm ADSRRC can be extended for hiding rules which are not from transactional data base. A more general rule hiding algorithm can be proposed. In algorithm RRLR, clustering of association rule is not done. So in future it can be incorporated to it. Currently algorithm RRLR hides rules only having multiple RHS items. It can be extended to multiple LHS and multiple RHS items.

8. REFERENCES

[1] ArisGkoulalas–Divanis; Vassilios S. Verykios "Association Rule Hiding For Data Mining" Springer, DOI 10.1007/978-1-4419-6569-1, Springer Science + Business Media, LLC 2010

[2] Elisa Bertino; Dan Lin; Wei Jiang," A Survey of Quantification of Privacy Preserving Data Mining Algorithms", Springer, Pages: 183–205,2008

[3] RamakrishnanSrikant;"Privacy Preserving Data Mining: Challenges and Opportunities", PAKDD '02 Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, Springer-Verlag London, UK , 2002

[4] Ahmed K. Elmagarmid; Amit P. Sheth , "Privacy-Preserving Data Mining Models and Algorithms - ADVANCES IN DATABASE SYSTEMS - Volume 34"

[5] M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim, and V. S. Verykios "Disclosure limitation of sensitive rules,".*In Proc. of the 1999 IEEE Knowledge and Data Engineering Exchange Workshop (KDEX'99)*, pp. 45–52, 1999.

[6] Vassilios S. Verykios, A.K. Elmagarmid, E. Bertino, Y. Saygin, and E. Dasseni, "Association Rule Hiding," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 4, pp. 434-447, 2004.

[7] Shyue-Liang Wang; Bhavesh Parikh,; Ayat Jafari, "Hiding informative association rule sets", ELSEVIER, Expert Systems with Applications 33 (2007) 316–323,2006

[8] Shyue-LiangWang ;*Dipen Patel ;Ayat Jafari ;*Tzung-Pei Hong, "Hiding collaborative recommendation association rules", Published online: 30 January 2007, Springer Science+Business Media, LLC 2007

[9] Shyue-Liang Wang; Rajeev Maskey; Ayat Jafari; Tzung-Pei Hong " Efficient sanitization of informative association rules" ACM , Expert Systems with Applications: An International Journal, Volume 35, Issue 1-2, July, 2008

[10] R. M. Oliveira; Osmar R. Za¨_ane, "Privacy Preserving Frequent Itemset Mining", IEEE International Conference on Data Mining Workshop on Privacy, Security, and Data Mining, Maebashi City, Japan. Conferences in Research and Practice in Information Technology, Vol. 14.2002

[11] Chih-Chia Weng; Shan-Tai Chen; Hung-Che Lo, "A Novel Algorithm for Completely Hiding Sensitive Association Rules", IEEE Intelligent Systems Design and Applications, 2008.,vol 3, pp.202-208, 2008

[12] Modi, C.N.; Rao, U.P.; Patel, D.R., "Maintaining privacy and data quality in privacy preserving association rule mining", IEEE 2008 Seventh International Conference on Machine Learning and Applications, pp 1-6, 2010

[13] Y.Saygin, V. S. Verykios, and C. Clifton, "Using Unknowns to Prevent Discovery of Association Rules," *ACM SIGMOD*, vol.30(4), pp. 45–54, Dec. 2001

[14] H. Mannila and H. Toivonen, "Levelwise search and borders of theories in knowledge discovery," *Data Mining and Knowledge Discovery*, vol.1(3), pp. 241–258, Sep. 1997.

[15] X. Sun and P. S. Yu. Hiding sensitive frequent itemsets by a border–based approach. Computing science and engineering, 1(1):74–94, 2007.

[16] A. Gkoulalas-Divanis and V.S. Verykios, "An Integer Programming Approach for Frequent Itemset Hiding," *In Proc. ACM Conf. Information and Knowledge Management (CIKM '06)*, Nov. 2006.

[17] A. Gkoulalas-Divanis and V.S. Verykios, "Exact Knowledge Hiding through Database Extension," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21(5), pp. 699–713, May 2009.

[18] Charu C. Aggarwal, Philip S. Yu, *Privacy-Preserving Data Mining: Models and Algorithms*. Springer Publishing Company Incorporated, 2008, pp. 267-286.