# Hiding Sensitive Association Rules over Privacy Preserving Distributed Data Mining

*Alaa Khalil Jumaa** *Sufyan T. F. Al-Janabi*** *Nazar Abedlqader Ali****
**Computer ScienceInstitute - University of Polytechnic*
***Collegeof Computer-University of Anbar*
****Collegeof Administration - University of Sulaimani*
alaa_alhadithy@yahoo.com
Received date: 2/10/2013 Accepted date: 27/2/2014

## Abstract

**The problem of Privacy Preserving Data Mining (PPDM) has become more important in recent years because of the increasing ability to store personal data about users, and the increasing sophistication of data mining algorithms. A number of techniques have been suggested in recent years in order to perform PPDM. These techniques are used to study different transformation methods associated with privacy. In this paper, a system for PPDDM of association rules is proposed. This system works under the common and realistic assumptions that parties are semi-honest, Semi-Trusted Third Party (STTP) and the databases are horizontally distributed over these parties. New algorithm for hiding sensitive rules is presented in this system. The experimental results for this algorithm has shown that it have good hiding accuracy with acceptable level of side effects when it compared with the same algorithm in centralized system and other existing algorithms in distributed database system. Furthermore, the proposed system uses the Secure Socket Layer (SSL) with commutative encryption to support the certifications and security over system various components.**

**Keywords: datamining;semi-honest; associationrules; distributeddatabase; commutative encryption.**

# أخفاء عناصر الاقتران الحساسة في عملية التنقيب الموزع والمحافظ على سرية البيانات

علاء خليل جمعة* سفيان تايه فرج الجنابي** نزار عبد القادر علي***
*جامعة السليمانية التقنية
**كلية الحاسوب . جامعة الانبار
***كلية الادارة والأقتصاد . جامعة السليمانية
تاريخ استلام البحث:2013/10/2 تاريخ قبول البحث:2014/2/27

## الخلاصة

مشكلة حماية الخصوصية في التنقيب عن البيانات اصبحت مهمة جدا في السنوات الاخيرة وذلك بسبب زيادة القدرة على خزن المعلومات الشخصية للمستخدمين وكذلك بسبب زيادة التعقيدات في خوارزميات التنقيب عن البيانات. في السنوات الاخيرة تم افتراض عدد من التقنيات لتنفيذ عملية حماية الخصوصية في التنقيب عن البيانات، هذة التقنيات تم استخدامها لدراسة طرق التحويل المختلفة والمرتبطة بالخصوصية.
في هذا البحث تم اقتراح نظام للحفاظ على الخصوصية في التنقيب عن البيانات الموزعة لأستخراج قواعد الأقتران (Association Rules). النظام المقترح يعمل في افتراضات واقعية وشائعة وهي ان الاطراف التي تحوي على البيانات الموزعة تكون شبه امينه (Semi-Honest) وشبه موثوقه (Semi-Trusted) وان البيانات موزعة بشكل افقي على جميع الاطراف. في هذا النظام تم اقتراح خوارزمية جديدة تقوم بعملية اخفاء قواعد الأقتران الحساسة. النتائج التي تم الحصول عليها من تطبيق الخوارزمية المقترحة تشير الى قدرتها على اخفاء قواعد الاقتران الحساسة بدقة جيدة وتاثيرات جانبية مقبولة مقارنةً مع نفس الخوارزمية عند تطبيقها على قواعد البيانات المركزية وكذلك عند مقارنتها مع الخوارزميات الموجودة سابقا والمطبقة على قواعد البيانات الموزعه. أضافة الى ذلك فان النظام المقترح يستخدم طبقة مآخذ التوصيل ( Secure Socket Layer) مع التشفير المتبادل (Commutative Encryption) لدعم المصداقية والأمنية بين جميع مكونات النظام.

**الكلمات الدالة:التنقيب عن البيانات، شبه امينة، قواعد الإقتران، قواعد البيانات الموزعة، التشفير التبادلي**

## Introduction

Recent advances in data mining and knowledge discovery have generated controversial impact in both scientific and technological arenas. Data mining is capable of analyzing vast amount of information within a minimum amount of time. On the other hand, the excessive processing power of intelligent algorithms puts the sensitive and confidential information that resides in large and distributed data stores at risk. Providing solutions to database security problems combines several techniques and mechanisms. An organization may have data at different sensitivity levels. This data is made available only to those with appropriate rights. Simply restricting access to sensitive data does not ensure complete sensitive data protection. Based on the knowledge of semantics of the application, the user may infer sensitive data items from non-sensitive data. Such a problem is known as 'Inference Problem' [1]. Sensitive rule hiding is a subfield of privacy preserving data mining (PPDM), a number of techniques like perturbation and anonymization have been developed to hide association rules from being discovered in the published data. Practically for a single data set, many data altering techniques for hiding association rules have been proposed [2]. In distributed data mining also protect the privacy for the data parties is very important, Privacy Preserving Distributed Data Mining (PPDDM) techniques are used to solve the privacy issues of distributed data mining. The PPDDM algorithms require collaboration between parties to compute the results, while provably preventing the disclosure of any information except the data mining results. To achieve this goal, tools Secure Multiparty Computation (SMC) domain are usually used. Recent research in the area of PPDM has devoted much effort to determine a trade-off between the right to privacy and the need of knowledge discovery, which is crucial in order to improve decision-making processes and other human activities. Such research has resulted in several approaches to the evaluation of privacy preserving techniques. In this section, we present a brief review of the major work in this area. S. Wang et al. proposed two algorithms, ISL (Increase Support of LHS) and DSR (Decrease Support of RHS), where LHS refers to Left Hand Side and RHS refers Right Hand Side, to automatically hide informative association rule sets without pre-mining and selecting of hidden rules. The first algorithm tries to increase the support of left hand side of the rule until the support or confidence for this rule becomes less than minimum support threshold and or minimum confidence threshold. The second algorithm tries to decrease the support of the right hand side of the rule until the support or confidence for this rule becomes less than minimum support threshold and or minimum confidence threshold. Both algorithms exhibit side effects like hide failure, loss rules, and appearance of new rule [3]. M. Gupta et al. proposed an algorithm which integrates the fuzzy

set concepts and Apriori mining algorithm to find useful fuzzy association rules and then to hide them through using privacy preserving technique. For hiding purpose, they decreased the support of the rule so as to be hidden by decreasing the support value of the item in either LHS or RHS of the rule [4]. Then, S. Wang et al. proposed a framework to hide sensitive association rules where the data sets are horizontally distributed and owned by non-trusting parties. In their proposal, hiding process depends on support-based and confidence-based distortion schemes. The process is accomplished by either decreasing its supports to be smaller than pre-specified minimum support or decreasing its confidence to be smaller than pre-specified minimum confidence. This framework was used to hide sensitive rules in each site depending on the global *Min_Supp* and *Min_Conf*threshold, and then each site sends sanitized database to non-trusted third party. Later, this third party merges the individually sanitized data and publishes the result. This framework suffers from large side effects because it depends on Min_Supp threshold and Min_Conf threshold to hide rules in each site (it needs more data modifications), and also it may hide rules that are frequent in local site but not frequent globally. This leads to an unnecessary modification of a number of transactions [5]. N. Dhutraj et al. proposed a system for hiding sensitive association rules using hybrid algorithm where the dataset is distributed over the network. For dataset collection, they used Secure Multi-party Computation (SMC) model in which cryptographic techniques are used for providing better security when data are transferred from each party to the trusted third party. The used hybrid algorithm was a combination of ISL and DSR techniques (depending on the location of sensitive itemset), and the association rule hiding was based on modifying the database transactions so that the confidence of the association rules could be reduced [6].

Finally, D. Jain et al. proposed an approach using the data distortion technique where the position of the sensitive item is altered but its support is never changed. The size of the database remains the same. It uses the idea of representative rules to prune the rules first and then it hides the sensitive rules. Advantage of this approach is that it hides maximum number of rules. This approach can be applied by removing the sensitive item from the transactions that fully support the sensitive rule and add this item to other transactions that do not or partially support this rule. Now the sensitive rule will be hidden without changing the support for the sensitive item. However, the existing approaches failed to hide all the desired rules which are supposed to be hidden in minimum number of passes. This approach also suffered from large side effects especially new rules are generated [7].

## Association Rules in Horizontally Partitioned Database

In a horizontally partitioned database, the transactions are distributed among ***n***sites. The global support count of an item set is the sum of all the local support counts. An itemsetX is globally supported if the global support count of X is bigger than minimum support of the total transaction database size. The global confidence of a rule X ⇒Y can be given as ***{X ∪Y}.sup / X.sup***. A k-itemset is called a globally large k-itemset if it is globally supported.

The DM algorithm is a method for distributed mining of association rules, the following steps shows how the distributed association rules can be calculated [8]:

1. Candidate Set Generation: Intersect the globally large itemsets of size k−1 with locally large k−1 itemsets to get candidates. From these, the classic Apriori candidate generation algorithm is used to get the candidate k itemsets.
2. Itemset Exchange: Broadcasts locally large itemsets to all sites – the union of locally large itemsets, a superset of the possible global frequent itemsets. (It is clear that if X is supported globally, it will be supported at least at one site.) Each site computes (using Apriori) the support of items in union of the locally large itemsets.
3. Support Count Exchange: Broadcasts the computed supports. From these, each site computes globally large k-itemsets.

## Problem Description

Distributed system assumed that there are ***n*** sites $S_0, S_1, \ldots, S_{n-1}$ , and the transaction database DB is horizontally divided into ***n*** non-overlapping partitions $db_0, db_1, \ldots, db_{n-1}$, where $DB = db_0 \cup db_1 \cup \ldots \cup db_{n-1}, db_i \cap db_j = \emptyset, 0 \leq i \neq j \leq n-1$. Each partition $db_i$ is assigned to site$S_i$. Clearly, $|DB| = |db_0| \cup |db_1| \cup \ldots \cup |db_{n-1}|$. $X.Sup_i$ is the local support counts of itemset X at site$S_i$, for $0 \leq i \leq n-1$. The global support count of X in DB is given as$X.sup = \sum_{i=0}^{n-1} X.Sup_i$. X is globally frequent if X.sup≥ min_support× |DB|. Similarly, X is locally frequent if X.supi≥ min_support× |dbi|. Also the global confidence for rule *X⇒Y in DB* given as [4]:-

$$\frac{(x \cup y).sup}{x.sup} = \frac{\sum_{i=0}^{n-1}(X \cup Y).Sup_i}{\sum_{i=0}^{n-1} X.Sup_i}., \qquad \ldots (1)$$

*and*X⇒Y is globally confidence if

$$\frac{\sum_{i=0}^{n-1}(X \cup Y).Sup_i}{\sum_{i=0}^{n-1} X.Sup_i} \geq \text{Min\_conf threshold}. \ldots (2)$$

However, two problems are addressed here, one is the protection of sensitive rules contained in the database (protect sensitive rules contained in the database from being discovered, while

non-sensitive rules can still be mined normally), the other is the protection of private data and the privacy of each site in distributed database. Thus all sites get just the result of mining process without knowing anything about the original database (extract relevant knowledge from large amounts of data distributed in different sites while protecting the privacy for each sites) [9].

The problem here is to hide the sensitive rules and minimize the loss items. When the global frequent for the sensitive rules satisfies these two conditions [10]:-

*i.* $Support(X \Rightarrow Y) = P(X \text{ and } Y) >= Min\_supp \ldots$ (3)

*ii.* $Confidence(X \Rightarrow Y) = P(X/Y) = [Support(X \cup Y) / Support(X)] >= Min\_conf. \ldots$ (4)

Where X and Y represent the candidate attributes. It shows that this rule is frequent and it should be hidden. This rule can be hidden by:

- Reduce the support of confidential rules (by decreasing the support of the corresponding largeXY).
- Reduce the confidence of rules (by increasing the support of X in transactions not supporting Y or decreasing the support of Y in transactions supporting both X and Y)

This can be done by deleting or adding a new data to the original database. This way prevents tools from discovering these rules, but the challenge is the data quality. When a support of items is changed, some other insensitive rules will also be affected either by hiding it or supporting another frequent rule. Thus good ways to reduce the negative side effects on data quality should be defined [10].

## Proposed Approaches and Hiding Algorithm

The main aim of the proposed system is to securely and efficiently preserve the privacy of distributed data mining. The sensitive rules and items are hided during protecting the privacy of each site in the system when the database is horizontally partitioned, and it works with non-trusted parties and semi-honest system. The proposed system generally used SSL (secure Socket Layer) to support certifications among all sites, SMC protocol to preserve privacy of each site and the proposed hiding algorithm to hide sensitive rules. This system generally can be divided into two phases: The first phase is responsible for protection of the privacy of each site during evaluation of the global association rules. This can be done by using SSL and SMC (commutative encryption tool is used to perform SMC). Each site encrypts its own sensitive frequent itemsets for the sensitive rules, and then passes them to other sites until all the sites have all the encrypted frequent itemsets for the sensitive rules which will be passed

to a common site to begin decryption. This set is then passed to each site which decrypts each frequent itemset. The final result represents the global confidence of sensitive rules.

The second phase tries to hide sensitive rules according to the global confidence that are calculated from phase one. This can be done when we reduce the support of confident rules by change (increase or decrease) the number of items that support these rules. This can be done by removing or adding these items to/from original database in each site until either the support for frequent itemsets become less than *Min_support* threshold or the confidence for the sensitive rules become less than *Min-conf*. Figure 1 represents the proposed system.

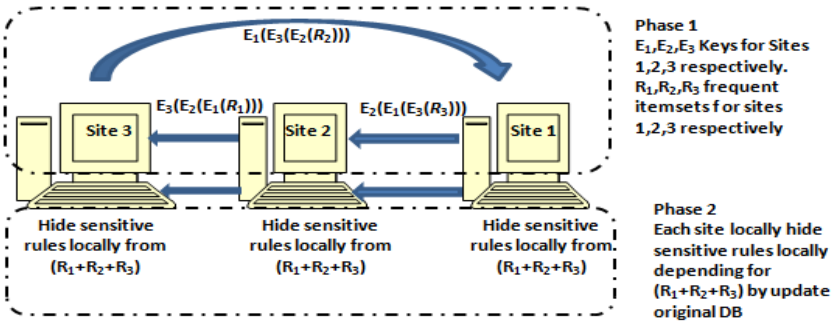


**Figure (1) Generalarchitectureoftheproposedsystem**

The major steps for *phase one* can be explained as follows (Assuming that we have three sites S1, S2 and S3):

1. *Determination of the local frequent itemset:*

Each site determines local frequent itemset for the sensitive rules ($R_i$) using the Apriori algorithm that is explained in Figure (2).

2. *Determining the globalconfidence for the sensitive rules for all site without disclosing the privacy of the sites:*

   a. Assume that the R1, R2, and R3 represent the local support items for the sensitive rules, and E1, E2, and E3 represent the commutative encryption algorithm with its keys for sites S1, S2, and S3 respectively (Pohlig–Hellman algorithm used to perform the commutative encryption, and RSA and SHA are used in SSL to satisfy the certification over all sites in the system).

Where $R1 = \sum_{i=1}^{n} R_{1i}$
$R2 = \sum_{i=1}^{n} R_{2i}$
$R3 = \sum_{i=1}^{n} R_{3i}$

   b. Secure connection established among all sites by using SSL techniques, and all the sites use public and private keys for SSL to certify each other.

c. All sites encrypts its frequent itemsets for the sensitive rules and sends it to the next site, then each site also encrypts frequent itemsets from other sites and send it to each other circularly. After encryption operations are completed for all sites, and because the commutative algorithm is used here, the encrypted frequent itemsets in each sites can be written as:

- E1 (E2 (E3 (R1)))
- E1 (E2 (E3 (R2)))
- E1 (E2 (E3 (R3)))

d. Then, the above encrypted frequent itemsets are decrypted in each site respectively using its decryption key (the decryption operations can occur in any order) and sends the result to the next site.

e. After all sites decrypt the encrypted frequent itemsets by its keys, they can be getting the results (R1, R2 and R3). These combined files (R1+R2+R3) represent the global confidence for the sensitive rules of all sites.

f. Now all sites have the global confidence for the sensitive rules without knowing from which site of these sensitive rules has come.

*procedureApriori (T, minSupport) { //T is the database and min-Support is the minimum support*
*Ck: Candidate itemset of size k*
*Lk: frequent itemset of size k*
*L1= {frequent items};*
*for(k= 1; $L_k$ !=Ø ; k++) do begin*
*$C_{k+1}$= candidates generated from $L_k$;*
*for each transaction t in database do{*
*increment the count of all candidates in $C_{k+1}$ that are contained in t*
*$L_{k+1}$= candidates in $C_{k+1}$ with min_support*
*}end*
*return $U_k L_k$;*

**Figure (2) Pseudo code for Apriori algorithm [11]**

In *Phase two,* a proposed algorithm for hiding sensitive rules in distributed database is used to reduce the support of confident rules by change (increase or decrease) the number of items that support these rules. The steps for hiding sensitive rules for each site can be explained as follows and the pseudo code and block diagram for the proposed algorithm in each site are explained in Figure (3) and Figure (4) respectively:

1. *Each site has Global confidence for the sensitive rule (G_Conf), local database $D_1$, Min_Supp and Min_Conf.*
2. *Input sensitive rules to be hidden.*
3. *For each sensitive rule {*
4. *Calculate the local confidence of sensitive rule (L_Conf).*
5. Calculate the new confidence of each site (*N_conf*) by

$$N\_Conf = L\_Conf - \left(\frac{L\_Conf}{G\_Conf} * (G\_Conf - min\_conf)\right) \quad \ldots (5)$$

*Where N_Conf = new confidence in local site.*
*G_Conf = Global confidence of all sites.*
*L_Conf = Confidence for local site.*
*Min_conf = minimum confidence threshold.*

6. *Extract all transactions that fully support sensitive rule ($T_r$).*
7. *Extract all transactions that partially support sensitive rule ($T_l$).*
8. *If $\frac{|T_r|}{|T_l|}$ <N_Conf, then go to 17 (end removing loop).*
9. *Evaluate the number of transaction ($|T_{Mr}|$) needed to be modified only with consequent (RHS) by.*

$$|T_{Mr}| = |T_r| - (N\_Conf * |T_l|). \quad \ldots(6)$$

10. *Evaluate the number of transaction ($|T_{Ml}|$) needed to be modified only with rule's antecedent (LHS) by*

$$|T_{Ml}| = \frac{|T_r|}{N\_Conf} - |T_l|. \quad \ldots (7)$$

11. *Evaluate the ratio for rule's consequent ($R_r$) by*

$$R_r = \frac{|T_l|}{(|T_r|+|T_l|)}$$

12. *Evaluate the ratio for rule's antecedent ($R_l$) by*

$$R_l = \frac{|T_r|}{(|T_r|+|T_l|)}$$

13. *Evaluate the number of transaction ($|T_{Mr}|$) needed to be modified by consequent according to the ratio by using*

$$|T_{MR}| = |T_{Mr}| * R_r$$

14. *Evaluate the number of transaction ($|T_{Ml}|$) needed to be modified by rule's antecedent according to the ratio by using*

$$|T_{ML}| = |T_{Ml}| * R_l$$

15. *Apply the procedure for adding items to rule's antecedent at LHS (As illustrated in Figure 3)*
16. *Apply the procedure for removing items from rule's consequent at RHS (As illustrated in Figure 3)*
17. *If all rules are hidden then go to 19*
18. *Else go to 2*
19. *END*

To clarify the operation of the proposed hiding algorithm, this algorithm used to hide number of sensitive rules in three local sites S1, S2 and S3, which have DB1, DB2 and DB3, will be considered respectively.

**Proposed Hiding Algorithm**

*Input: a source database $D_1$, global confidence, min_support, min_confidence, set of sensitive items X, and number of iteration*
*Output: a transformed database $D_1'$, where rules containing X on LHS will be hidden.*
*For each iteration {*
*1. For each item in x ∈X {*
*2. Generate all rules that contain x in LHS*
*3. For each rule r do {*
*4. Calculate L_Conf.*
*5. N_Conf= L_Conf – ( $\frac{L\_Conf}{G\_Conf}$ * (G_Conf – min_conf)).*
*6. Extract $T_r$= {t ∈ D / t fully support r }*
*7. Extract $T_l$= {t ∈ D / t partially support r}*
*8. If $\frac{|T_r|}{|T_l|}$ <N_Conf, then go to 31 (end hide loop).*
*9. Calculate $|T_{Mr}| = |T_r| - (N\_Conf * |T_l|)$. //RHS.*
*10. Calculate $|T_{Ml}| = \frac{|T_r|}{N\_Conf} - |T_l|$. // LHS*
*11. Calculate $R_r = \frac{|T_l|}{(|T_r|+|T_l|)}$*
*12. Calculate $R_l = \frac{|T_r|}{(|T_r|+|T_l|)}$*
*13. Calculate $|T_{MR}| = |T_{Mr}| * R_r$*
*14. Calculate $|T_{ML}| = |T_{Ml}| * R_l$*
*// **Add items to (LHS)***
*15. For each item i in LHS {*
*16. Count $|ID_{Ni}|$ // support for LHS without item i and RHS items in DB*
*17. $|ID_{tNi}| = \sum_{i=1}^{n} ID_{Ni}$ // summation for support items;*
*18. If $|T_{ML}| > |ID_{tNi}|$ // no enough transactions can hide sensitive rule*
*{ $|T_{ML}| = |ID_{tNi}|$*
*$|T_{MR}| = |T_r| - (N\_Conf * (|T_l| + |ID_{tNi}|))$. }*
*19. Calculate$|Iri| = |ID_{Ni}|/|ID_{tNi}| * |T_{ML}|$*
*20. Extract ($T_{lNi}$) {t ∈ D / t partially support r and not support i}.*
*21. Sort ($T_{lNi}$) // in ascending order.*
*22. Set_to_one (t.values_of_items i, $T_{lNi}$)*
*23. } // end for add loop*
*// **Remove items from (RHS)***
*24. For each item i in RHS {*
*25. Count |IDi| // support of item i in database,*
*26. Calculate $|ID_{tr}| = \sum_{i=1}^{n} ID_i$*
*27. Sort ($T_r$) // in ascending order according to number of items in transaction*
*28. Sort (Ir) // in descending order according to (|Ir|).*
*29. Set_to_zero (t.values_of_items i, $T_r$)*
*30. } // end ofremove loop*
*31. } // end hiding rule*
*32. } // end of loop x rule*
*33. } // end of iteration*
*END*

**Figure (3) Pseudo code for the proposed hiding algorithm**

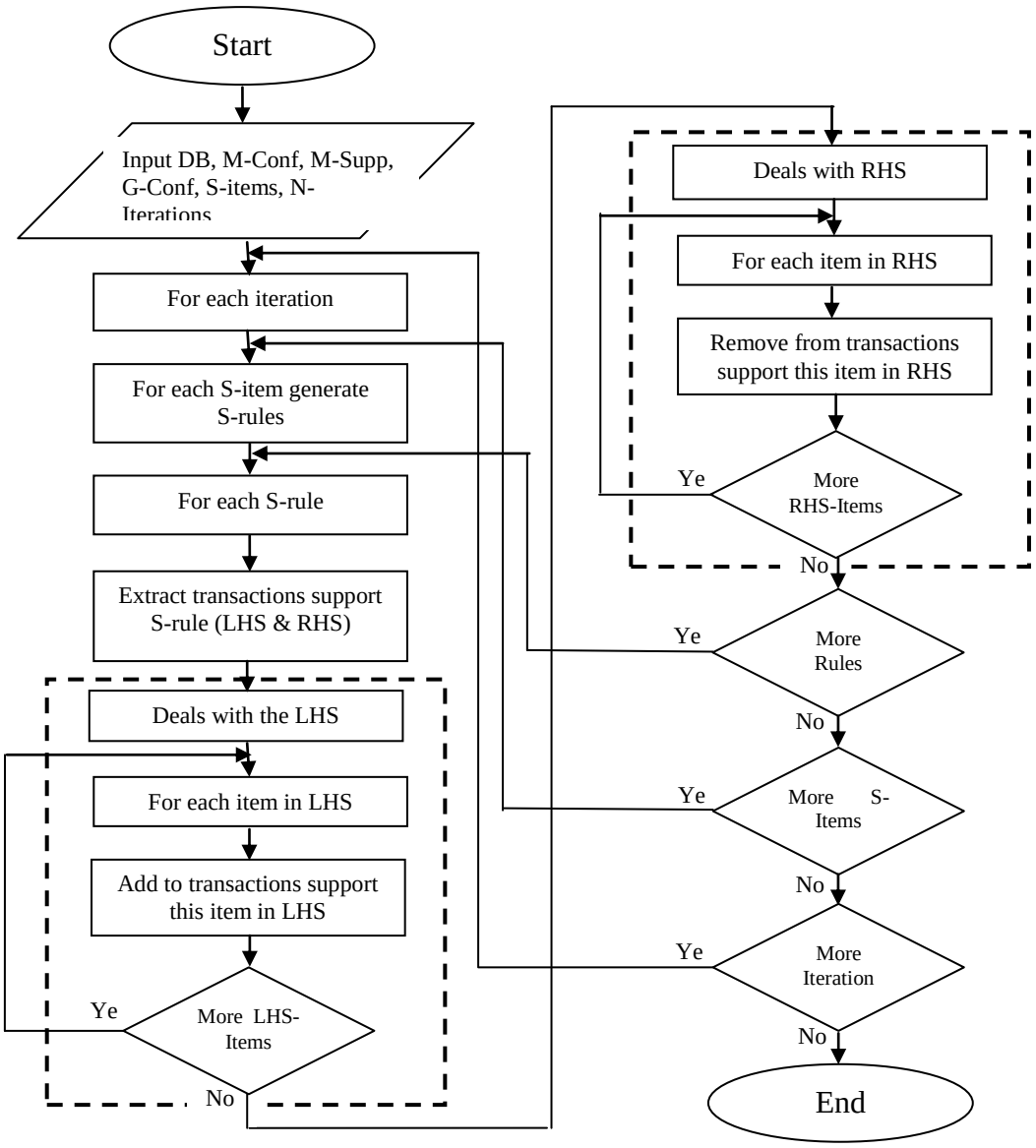


**Figure (4) Block diagram for the proposed algorithm**

## Results Analysis and Performance Evaluation

Two main effects have been considered to evaluate the performance for the proposed algorithm: execution time and side effects. For execution time, the running time required to hide sensitive rules is measured. For side effects, the percentages of hiding failure, the new rules generated and the lost rules are measured, respectively. The hiding failure side effect measures the percentage of the number of sensitive association rules that cannot be hidden to the number of rules that need to be hidden. The new rules side effect measures the percentage of the number of new rules appeared in the sanitized data set but not in the original data set to the number of total association rules in the original data set. The lost rules side effect measures the percentage of the number of non-sensitive rules that are in the original data set but not in the sanitized data set to the number of association rules in the original data set.

Theexperiments for the proposedalgorithmperformed onanotebookwith2GMHz processor and2GBmemory,underWindowsXPoperating system(inadistributed system setting there are three notebooks with the same properties). The sequence database (Binary database)generatedfortheexperiments canbegenerated byusingaSequenceDatabase Generator "SeqDBGen" [12]thatworkslikeIBMdatagenerator[13]. To evaluate the performance of the proposed algorithm to hide sensitive rules in distributed database system, it is used to hide all sensitive rules that include specific or sensitive item in LHS. Hiding process is applied in each site. Datasets of 30000, 60000, and 90000 transactions are distributed for three sites, in each site all the frequent itemsets are generated and aggregated with the frequent itemsets of other sites. Then, all the association rules that have the minimum support and minimum confidence threshold are evaluated and stored in an appropriate file. Now the proposed algorithm is applied in each site to hide all the rules that have sensitive item in LHS. When the hiding process is completed, the released database will be mined and the new frequent itemset are extracted. These itemsets are aggregated for all sites and all association rules that have minimum support and minimum confidence threshold are extracted and saved in a new file. The side effects of this algorithm can be evaluated by comparing the results of the association rules of these two files.

Time measuring represents the average time required for hiding process in all sites. Finally the results (side effects and required time) in distributed system are compared with the results of the proposed algorithm with the same database in central system. The experiments here use range of minimum support threshold 6-10% and minimum confidence threshold 40-50 % in central and distributed database. The experimental results are obtained by averaging from 4 independent trials for each size of transaction with different sensitive rules. The following Figures below explain the average of the experimental results (hide ratio, side effects, and time measurements) for hiding sensitive rules in both central and distributed database. Figures 5 and 6 represent the ratios for the hiding rules to the all association rules. Figures 7 and 8 shows that there is no clear change in the ratios of hiding failures, lost rules, and new rules in the distributed database when it is compared with the central database for the same hiding ratios. This shows that proposed algorithm for hiding sensitive rules in distributed system works properly and the results for hiding process are not affected when the data is distributed. Figures 9 and 10 shows that the measured time is a linear growth with the size of database and the time required in distributed database is less than the time required in central database.

Figures (11) and (12) shows that the transactions needed to be modified in database for the proposed hiding algorithm is less than the number of the transactions needed to be modified by other existing algorithms used in distributed systems. The proposed algorithm here reduce modified transactions in both side (LHS and LHS) compared to the algorithm proposed by Wang et al. in [5]. This will also reduce the side effects (new and lost rules) that occur in database during hiding operations.

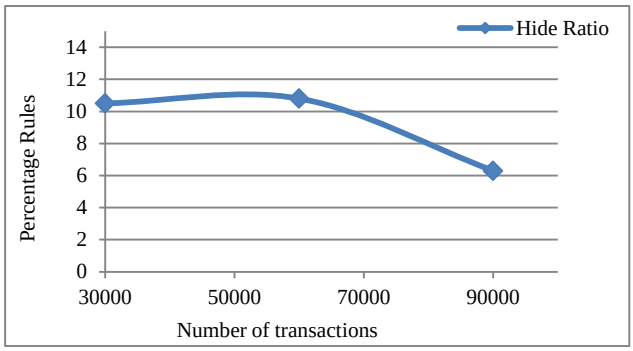


**Figure (5) Hiding ratios of PROPOSED HIDING ALGORITHM IN CENTRAL**

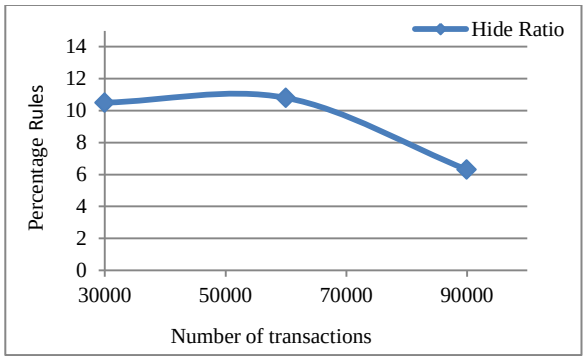


**Figure (6) Hiding ratios ofPROPOSED HIDING ALGORITHM IN DISTRIBUTED**

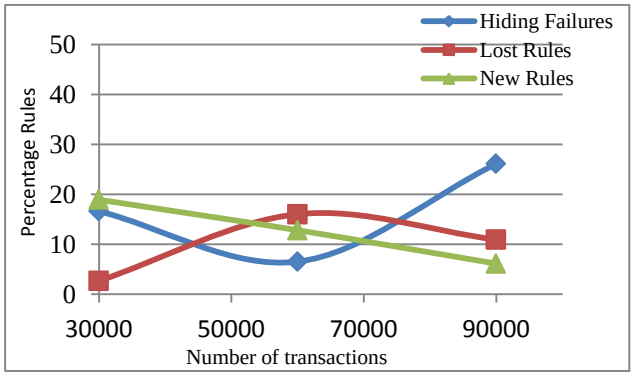


**Figure (7)Side effects of PROPOSED HIDING ALGORITHM IN CENTRAL**

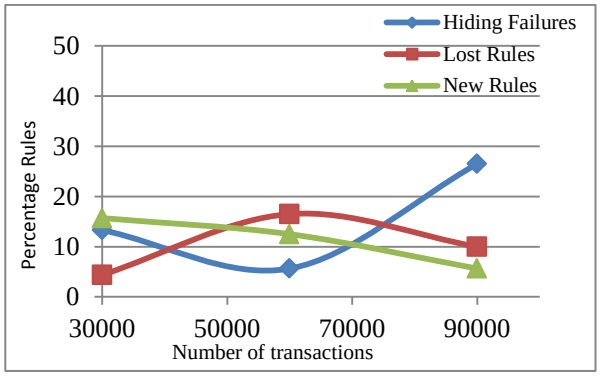


**Figure (8)Side effects of PROPOSED HIDING ALGORITHM IN DISTRIBUTED DATABASE**

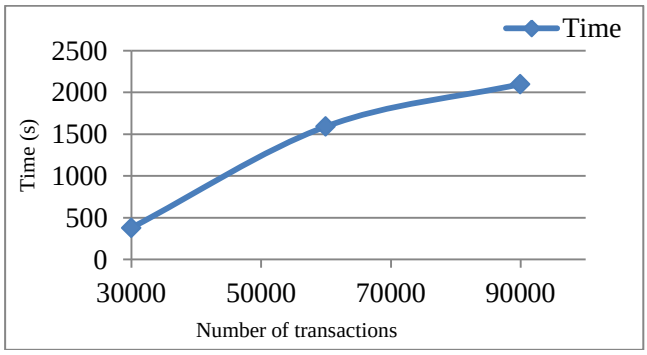


**Figure (9) Required time for PROPOSED HIDING ALGORITHM IN CENTRAL DATABASE**

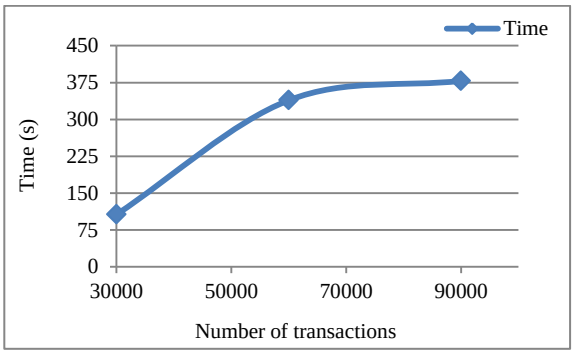


**Figure (10) Required time for PROPOSED HIDING ALGORITHM IN DISTRIBUTED DATABASE**

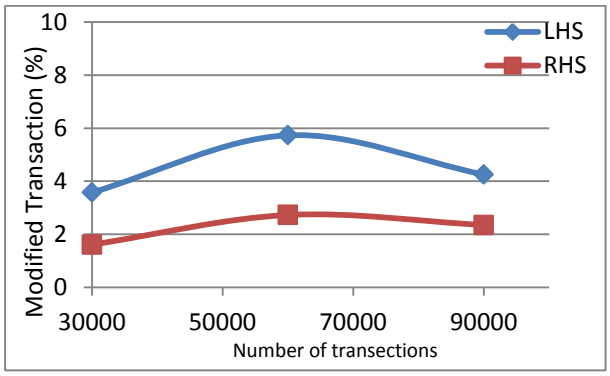


**Figure (11) Modified Transactions ratio for Wang algorithm in DISTRIBUTED DATABASE**

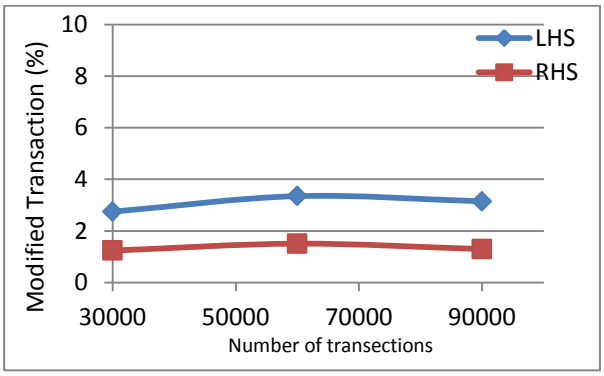


**Figure (12) Modified Transactions ratio for proposed algorithm in DISTRIBUTED DATABASE**

## Conclusion and Future Work

Inthispaperweproposedasystemtoallowsiteslikecompanies,banksorother organization stoshare knowledge while protect in gat the same time the privacy ofeach site.We allow all system sites to certify one another by using SSL protocol and also protect the privacy for these sites during evaluate the global association rules. Also the proposed hiding algorithm is presented to hide sensitive association rules in distributed data mining, the operation for this algorithms depends on theratioofconfidence for the association rulesineach siteandtheratioof count foreach iteminthe sensitiveassociation rulesforlocaldatabase.

According totheobtainedresults,proposed system and algorithmhavea reasonableside effect (hiding failures, new and lost association rules), while obtaining a significant reduction in the time requirementforthecaseofthedistributeddatabasesystem. Also the results shows that proposed hiding algorithm in distributed system works properly when it compared with the same algorithm in central database system, that mean proposed algorithm in distributed system is efficient and it has a good accuracy. Furthermore the proposed system reduces the communication overhead that can happen during redundant operations (encryption and decryption) in commutative encryption by using a small size of data transfer. This data represents only the sensitive frequent itemsets for the sensitive rules.

Asafuturework,the proposed system can be developed to support solutions when system parties' shares vertically distributed database and also when it shares hybrid distributed database, and also it can be enhanced to support PPDDM for other data mining techniques such as clustering and classifications.

## References

[1] K. Sathiyapriya, G. SudhaSadasivam, " **A Survay on Privacy Preserving Association Rule Mining**", International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.3, No.2, March 2013.

[2] E. Dasseni, V. S. Verykios, A. K. Elmagarmid, and E. Bertino, "**Hiding Association Rules by Using Confidence and Support**", Proceedings of the 4th International Workshop on Information Hiding, pages 369–383, 2001.

[3] S. Wang, B. Parikh, and A. Jafari, "**Hiding informative association rule sets**", Journal of Expert Systems with Applications, Vol. 33, pp.316–323, 2007.

[4] M. Gupta and R. C. Joshi, "**Privacy Preserving Fuzzy Association Rules Hiding in Quantitative Data**", International Journal of Computer Theory and Engineering, Vol. 1, No. 4, October, 2009.

[5] S. Wang, T. Zheng, T. Hong, and Y. Wu, "**Hiding Predictive Association Rules on Horizontally Distributed Data**", IEA/AIE 2009, Springer-Verlag Berlin Heidelberg LNAI 5579, pp. 133–141, 2009.

[6] N. Dhutraj, S. Sasane, and V. Kshirsagar, "**Hiding Sensitive Association Rule for Privacy Preservation**", Advanced Information Management and Service (IMS), 6th International Conference Publication-2010.

[7] D. Jain, A. Sinhal, N. Gupta, P. Narwariya, D. Saraswat, and A. Pandey, "**Hiding Sensitive Association Rules without Altering the Support of Sensitive Items**", international Journal of Artificial Intelligence & Applications,Vol.3, No.2, 2012.

[8] M. Kantarcioglu and C. Clifton, "**Privacy-Preserving Distributed Mining of Association Rules on Horizontally Partitioned Data**", IEEE transactions on Knowledge and Data Engineering, Vol. 16, No. 9, September 2004.

[9]Chang,C.-C.,J.-S.Yeh,andY.-C.Li,"**Privacy-PreservingMiningofAssociation Ruleson Distributed Databases**",*IJCSNS International Journal of Computer Science and Network Security*,VOL.6 No.11,November 2006.

[10] A. K. Juma'a, S. T. Faraj, N. A. Ali, "**Hiding Sensitive Frequent Itemsets over Privacy Preserving Distributed Data Mining**", Fifth Scientific Conference in Information Technology, University of Mosul, 2012.

[11] S. Samet and A. Miri, "**Secure Two and Multi-party Association Rule Mining**", Proceedings of the 2009 IEEE, Symposium on Computational Intelligence in Security and Defense Applications (CISDA), 2009.

[12] Sequence Database Generator "Seq DBGen", 2013.http://www.philippe-fournier-viger.com/seqdbgen.

[13] IBM data generator, 2013. http://www.ibmquestdatagen.sourceforge.net.