# A Novel Method for Protecting Sensitive Knowledge in Association Rules Mining

En Tzu Wang Guanling Lee Yu Tzu Lin
*Dept. of Computer Science & Information Engineering*
*National Dong Hwa University, Hualien, Taiwan, 974, R.O.C*
*Email: m9221009@em92.ndhu.edu.tw*

## Abstract

*Discovering frequent patterns from huge amounts of data is one of the most studied problems in data mining. However, some sensitive patterns with security policies may cause a threat to privacy. We investigate to find an appropriate balance between a need for privacy and information discovery on frequent patterns. By multiplying the original database and a sanitization matrix together, a sanitized database with privacy concerns is obtained. Additionally, a probability policy is proposed to against the recovery of sensitive patterns and reduces the modifications of the sanitized database. A set of experiments is also performed to show the benefit of our work.*

## 1. Introduction

Rapid advances in network communications and software /hardware technologies enable users to collect huge amounts of data. At the same time, high-speed computation has made it feasible to analyze these data. This is called *data mining*. The information discovered from data plays an important role in many applications such as business management, marketing analysis etc. However, it also brings some problems about privacy.

Data are valuable; however, they are also valuable to the opponents if they are naked. Consider a scenario in [6] and extend it, suppose that there are a server and many clients; each client has a transactional database. The clients want the server to gather statistical information about associations among items to provide recommendations to their customers. However, the clients do not want the server to know some *sensitive patterns* generated from their databases. A sensitive pattern is a frequent pattern with security policies, such as commercial consideration. Therefore, before a client sends its database to server, sensitive patterns should be hidden according to specific privacy policies. The server only can gather information from the modified databases.

In recent years, more and more researches in data mining emphasize the seriousness of the problems about privacy which can be classified into two categories: *data privacy* problems and *information privacy* problems. Data privacy problems with two broad approaches focus on the privacy of sensitive data. The randomization approach emphasizes individual privacy and sends randomized data to prevent original data revealing [3][5][6][15]. In the secure multi-party computation approach [7][9][14][16], each party owns a confidential database and wishes to build data mining models on the union of them without revealing any unnecessary information like the individual records.

On the other hand, information privacy problems which are proved NP-Hard [1] focus on hiding association rules or frequent patterns containing highly sensitive knowledge. There are many heuristic methods have been proposed to solve this problem [8][10][11][12][13][17]. In [10] and [11], a privacy presservation framework is proposed to hide sensitive patterns. A transaction retrieval engine is used to speed up the process of finding the *sensitive transactions* which are identified according to the sensitive patterns. They also bring up a *privacy threshold* controlled by users to decide the degree of alterations of these sensitive transactions. How to choose the sensitive transactions and how to choose the *victim items* from the sensitive transactions are the two most important issues in it. In [12], a heuristic approach, Sliding Window Algorithm (SWA), is proposed to enforce privacy in association rules mining. It hides association rules by decreesing their supports. Additionally, it has better performance than the framework proposed in [10]. However, the approaches proposed in [10][11][12] all suffer from *Forward-Inference Attack* (F-I Attack) problem discussed in [13]. The F-I Attack problem is that if a pattern is hidden in a set of modified patterns which has to be published, but all sub-patterns of the pattern are still frequent in the set, then the attackers can infer that the pattern is hidden painstakingly. To avoid F-I Attack problems, basing on the solution discussed in [13], at least one sup-pattern with length 2 of the pattern should be removed or the hiding pattern will be inferred recursively. However, they only modify the frequent patterns and don't consider how to modify the original database. In [8], the idea of using correlation matrix for hiding sensitive patterns is introduced. However, only the maximal patterns are considered in the work. In [17], the authors investigate confidentiality issues of a broad category of association rules and present several strategies for hiding the rules. Although the algorithms ensure privacy preserving, they may modify true data values and relationships by adding new items into the original transactions.

In this paper, we propose a novel method for modifying database to hide sensitive patterns. By observing the relations between sensitive patterns and non-sensitive patterns, a *sanitization matrix* is defined. By setting the entries to appropriate values and multiplying the original transaction database to the sanitiza-

0730-3157/05 $20.00 © 2005 **IEEE**

tion matrix, a *sanitized database* which can resist F-I Attack is gotten. The sanitized database is the database that has been modified for hiding sensitive patterns with some privacy concerns. Moreover, we use some probability policies with a level of confidence given by users to against the recovery of sensitive patterns and reduce the modifications of the sanitized database.

The rest of this paper is organized as follows. In Section 2, the basic concepts of our approach are introduced. The sanitization process is discussed in Section 3. The simulation result is shown in Section 4. Finally, Section 5 concludes our work.

## 2. Preliminary

### 2.1. Problem Definition

The problem of discovering association patterns is defined as finding relations between the occurrences of items within transactions [2]. For example, an association pattern might be "bread, milk; support = 10%", which means there are 10% of the transactions contain both items. In the association patterns, each pattern should have a measure of certainty associated with it that accesses the validity of the pattern. It is called support. The support of an association pattern refers to the percentage of task-relevant transaction for which the rule is true. Therefore, *minimum support* is defined to be the minimum threshold for an association pattern to be meaningful. A *frequent pattern* is the association pattern that satisfies the minimum support.

In this approach, a transactional database is represented as a binary matrix $D$ where the rows represent transactions and the columns represent items. Entry $D_{ij}$ is set to 1, if item $j$ is purchased in transaction $i$ and 0, otherwise. The problem of hiding sensitive patterns can be formulated as follows, let $P$ be the set of all frequent patterns mined from $D$ except the patterns with length 1, $P_H$ be the set of sensitive patterns, $\sim P_H$ be the set of remainder frequent patterns (non-sensitive patterns), i.e. $P_H \cup \sim P_H = P$. Our approach is to transform $D$ into a sanitized $D'$, such that only the patterns belong to $\sim P_H$ can be mined from $D'$. Moreover, the patterns belong to $P_H$ will never suffer from F-I Attacks discussed in the previous section.

### 2.2. Basic Concepts of Sanitization Matrix

In our approach, $D$ is multiplied by a sanitization matrix $S$ to get $D'$. That is, $D'_{n\times m} = D_{n\times m} \times S_{m\times m}$. If $S$ is an identity matrix (i.e., $S_{ij} = 1$ if $i = j$, otherwise, $S_{ij} = 0$), $D'$ will be equal to $D$. By setting $S_{ij}$ where $i \neq j$ to the appropriate value, $D'$ will be gotten. In the following, the basic concept of our approach is discussed.

**2.2.1 New Definition of Matrix Multiplication.** In order to fit properties of transactional databases, some different definitions of matrix multiplication are given as follows:

1. If $D_{ij} = 0$, $D'_{ij}$ is set to 0 directly. Because our goal is to hide sensitive patterns by decreasing their supports, therefore, only need to take care of how and when an entry with its value equal to 1 in $D$ should be converted to 0 in $D'$. Moreover, it also guarantees that there are no new artificial patterns created by the sanitization process.
2. If the value of $\sum_{k=1}^{m} D_{ik} \cdot S_{kj}$ is not smaller than 1, set $D'_{ij}$ to 1.
3. If the value of $\sum_{k=1}^{m} D_{ik} \cdot S_{kj}$ is not larger than 0, set $D'_{ij}$ to 0.

**2.2.2. Setting of "–1".** To hide a pattern *{i, j}*, we have to decrease its support. For example, if $D_{ki}$ and $D_{kj}$ are both equal to 1 for some $k$, we can re- place the value of $D_{ki}$ or $D_{kj}$ by 0 to decrease the support of *{i, j}*. If a sufficient amount of such entries could be replaced by 0, *{i, j}* will no longer be a frequent pattern. Refer to Fig. 1, if $S_{21}$ is set to –1, $D'_{21}$ and $D'_{41}$ will become 0, the support of item 1 is decr- eased; if $S_{12}$ is set to –1, $D'_{22}$ and $D'_{42}$ will become 0, the support of item 2 is decreased. Therefore, the support of *{1, 2}* can be decreased by setting $S_{21}$ or $S_{12}$ to –1. Moreover, if $S_{ij}$ is set to –1, for the row that $D_{ti}$ and $D_{tj}$ both equal 1, $D'_{tj}$ will become 0.

$$\underset{D}{\begin{pmatrix}0&0&1\\1&1&0\\0&0&1\\1&1&1\end{pmatrix}_{4\times3}} \times \underset{S}{\begin{pmatrix}1&0&0\\-1&1&0\\0&0&1\end{pmatrix}_{3\times3}} = \underset{D'}{\begin{pmatrix}0&0&1\\0&1&0\\0&0&1\\0&1&1\end{pmatrix}_{4\times3}} \quad \underset{D}{\begin{pmatrix}0&0&1\\1&1&0\\0&0&1\\1&1&1\end{pmatrix}_{4\times3}} \times \underset{S}{\begin{pmatrix}1&-1&0\\0&1&0\\0&0&1\end{pmatrix}_{3\times3}} = \underset{D'}{\begin{pmatrix}0&0&1\\1&0&0\\0&0&1\\1&0&1\end{pmatrix}_{4\times3}}$$

**Fig.1: Effect of setting –1 in *S*.**

**2.2.3. Setting of "1".** Setting entries to –1 in $S$ can reduce the support of sensitive patterns; however, it may also lead to conceal non-sensitive patt- erns. This kind of conditions can be solved by setting proper ent- ries to 1 in $S$. Consider the example in Fig2, let minimum supp- ort be 50%, *{1, 2}* and *{1, 3}* be the sensitive and non-sensitive pattern respectively. Refer to the equation on the left-hand side, *{1, 2}* and *{1, 3}* are both hidden in $D'$. However, if $S_{31}$ is set to 1, those entries that $D_{t1}$ and $D_{t3}$ are both equal to 1, $D'_{t1}$ will keep the same value as $D_{t1}$. Therefore, setting $S_{ij}$ to 1 can keep the relation between item $i$ and item $j$ by enhancing the strength of $j$.

$$\underset{D}{\begin{pmatrix}1&0&1\\1&1&0\\0&0&1\\1&1&1\end{pmatrix}_{4\times3}} \times \underset{S}{\begin{pmatrix}1&0&0\\-1&1&0\\0&0&1\end{pmatrix}_{3\times3}} = \underset{D'}{\begin{pmatrix}1&0&1\\0&1&0\\0&0&1\\0&1&1\end{pmatrix}_{4\times3}} \quad \underset{D}{\begin{pmatrix}1&0&1\\1&1&0\\0&0&1\\1&1&1\end{pmatrix}_{4\times3}} \times \underset{S}{\begin{pmatrix}1&0&0\\-1&1&0\\1&0&1\end{pmatrix}_{3\times3}} = \underset{D'}{\begin{pmatrix}1&0&1\\0&1&0\\0&0&1\\1&1&1\end{pmatrix}_{4\times3}}$$

**Fig.2: Effect of setting 1 in *S*.**

## 3. Sanitization Process

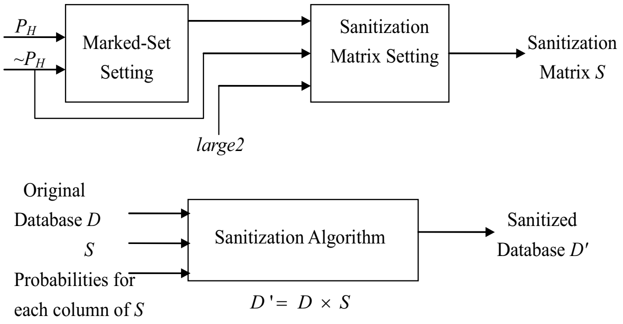


**Fig.3: The flowchart of the sanitization process.**

Fig.3 shows the flowchart of the work. Each of the components will be discussed in this section.

### 3.1. Sanitization Matrix Setting

As discussed in section 1, to avoid F-I Attack, for each sensitive pattern $P$ in $P_H$, at least one pattern belong to the *pair-subset* of $P$ should be hidden or the attacker can infer from the subset with length 2 to the sensitive pattern recursively. The pair-subset is defined as follows:

**Definition 1**: Let $F$ be a frequent pattern. The *pai-subpattern* of $F$ is a sub-pattern of $F$ with length 2. The set which includes all pair-subpatterns of $F$ is called the pair-subset of $F$.

For example: if *{1, 2, 3}* is a frequent pattern, then *{1, 2}* is a pair-subpattern of *{1, 2, 3}*. Moreover, *{{1, 2}, {1, 3}, {2, 3}}* is the pair-subset of *{1, 2, 3}*. Assume that *{1, 2, 3}* is sensitive, in order to avoid F-I Attack, at least one of the elements which belong to *{{1, 2}, {1, 3}, {2, 3}}* need to be hidden.

```
Marked-Set Generation
Input: ~$P_H$ and $P_H$
Output: Marked-Set
1. If $P \in P_{H,}$ and the length of $P$ is 2 then put $P$ into Marked-Set.
2.∀ Remainder $P \in P_H$ (the length of $P$ is greater than 2) do
        If ($P$ has no pair-subpatterns included in Marked-Set) {
            Generate $k$ groups; $k$ is equal to the number of pair-subpatterns
            generated from $P$. A class label of each group is named by each
            pair-subpattern of $P$. $P$ is stored in each group.
        }
3. Merge the groups which have the same class label.
4. ∀ $NP \in$ ~$P_H$ do
     Generate pair-subpatterns of $NP$. Count the frequency of each
     pair-subpattern in ~$P_H$. e.g. Let ~$P_H$ = {{1, 3}, {1, 3, 5}} ⇒ The
     pair-subpatterns of {1, 3} and {1, 3, 5} are {1, 3} and {1, 3}, {1, 5},
     {3, 5} respectively; the frequencies of {1, 3}, {3, 5}, {1, 5} are 2, 1, 1
     respectively.
5. ∀ Groups do
        If the class label of the group doesn't equal any pair- subpattern
        generated in step4, the frequency of the group is set to 0.
        Otherwise, it is set to the frequency of the pair-subpattern which
        equals the class label of the group.
6. Sort the groups by frequency in the increasing order.
7. for ($i$ = 1 to the number of groups − 1){
        for ($j$ = i + 1 to the number of groups){
           Compare $G_i$, $G_j$. For all overlap patterns stored in $G_i$ and $G_j$ do {
                if (the number of patterns stored in $G_i$ isn't equal to the
                number of patterns stored in $G_j$ )
                            Remove overlaps from the smaller one.
                else{
                     if (the frequency of $G_i$ isn't equal to the frequency of
                     $G_j$ )
                     Remove overlaps from the group with larger frequency.
                else
                          Remove overlap from the group chosen randomly.
                }
           }
        }
}
8. ∀ Groups do
     If the number of patterns stored in a group is greater than 0, put the
     class label of the group into Marked-Set.
```

**Fig.4: The algorithm of Marked-Set Generation.**

In our work, a temporary set, *Marked-Set*, is used to store the *victim pair-subpatterns*. A victim pair-subpattern is the pair-subpattern selected from the pair-subset of a sensitive pattern and used to be hidden. After setting up Marked-Set, all patterns in Marked-Set are used to set the related entries to –1 in $S$. The Marked-Set Generation is showed in Fig.4.

In step 2, the remainder patterns whose pair-subpatterns are not stored in Marked-Set are used to generate groups for finding suitable pair-subpatterns. Because the patterns belong to ~$P_H$ should not be affected, the selection of victim pair-subpatterns should take the patterns in ~$P_H$ into account. In step4, the frequencies of the pair-subpatterns of the non-sensitive patterns are calculated. If the frequency of a pair-subpattern appearing in the pair-subsets of the non-sensitive patterns is small, the number of patterns in ~$P_H$ affected by hiding this pair-subpattern is also small. Therefore, in step7, the pair-subpatterns with smaller frequencies are chosen to put in Marked-Set. In step 6, we sort the groups by frequency in increasing order to aggregate the patterns in the groups with small frequency; therefore fewer patterns in ~$P_H$ may be affected. In step3~7, the groups are merged according to the patterns stored in it, such that the sensitive patterns can be hidden by hiding a common pair-subpattern at one time. Moreover, the dissimilarity between $D$ and $D'$ can also be reduced.

After getting the Marked-Set, $S$ can be set as Fig. 5:

```
Sanitization Matrix Setting
Input: Marked-Set, ~$P_H$, large 2
Output: Sanitization Matrix
1. $S_{ii} = 1$,  $\forall i, 1 \le i \le m$ .        // diagonal entries
2.∀ {i, j} ∈ Marked-Set do{
      if (the number of patterns contain $i$ in ~$P_H$ < the number of patterns
      contain $j$ in ~$P_H$){
          Set $S_{ji}$ to –1.   // Decreasing item $i$'s support.
     }
      else if (the number of patterns contain $i$ in ~$P_H$ > the number of
      patterns contain $j$ in ~$P_H$){
          Set $S_{ij}$ to –1.   // Decreasing item $j$'s support.
     }
     else{
          if (the number of patterns contain $i$ in Marked-Set > the
          number of patterns contain $j$ in Marked-Set){
             Set $S_{ji}$ to –1.   // Decreasing item $i$'s support.
          }
          else if (the number of patterns contain $i$ in Marked-Set < the
          number of patterns contain $j$ in Marked-Set){
             Set $S_{ij}$ to –1.   // Decreasing item $j$'s support.
          }
          else{
             set $S_{ij}$ or $S_{ji}$ to –1 randomly.
          }
     }
}
3.   ∀ {i, j} ∈ {large 2 – Marked-Set} do {
          Set $S_{ij}$ and $S_{ji}$ to 1.  // enhance $i$'s and $j$'s strengths at once.
    }
4.   $S_{ij}$ = 0, otherwise.
```

**Fig.5: The algorithm of Sanitization Matrix Setting.**

In step 2, because all patterns stored in Marked-Set are needed to be hidden, thus, the item with smaller effect on the patterns in ~$P_H$ is chosen to be a *victim item*. A victim item is an item whose support is selected to be decreased to hide the pattern stored in Marked-Set. When the effect on the patterns in ~$P_H$ is the same, a victim item is chosen according to the number of times it appeared in Marked-Set. This is because choosing the more frequent one can decrease supports of many patterns which belong to Marked-Set at one time and reduce the dissimilarity between $D$ and $D'$. In step 3, the correlations of the patterns belong to $\sim P_H$ are enhanced by setting the related entries in $S$ to 1.

## 3.2. Probability Policies

**3.2.1. Distortion Probability $\rho$ .** By setting –1 in sanitization matrix, the supports of the pair-subpatterns belong to $P_H$ can be decreased. However, refer to Fig.6, it brings the following problem.

$$\underset{D}{\begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}_{4\times 3}} \times \underset{S}{\begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}_{3\times 3}} = \underset{D'}{\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}_{4\times 3}} \quad \underset{D}{\begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}_{4\times 3}} \times \underset{S}{\begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}_{3\times 3}} = \underset{D'}{\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}_{4\times 3}}$$

**Fig.6: Over Hiding problem of setting –1 in *S*.**

No matter the left-hand or right-hand equation, the support of *{1, 2}* in $D'$ is 0. That is, item *1* and item *2* never appear together, and they are mutual exclusive! This situation almost never happens in the normal database. The attackers may interest in this situation and infer that *{1, 2}* is hidden deliberately. To hide the sensitive patterns, only need to make their supports smaller than minimum support and need not to decrease their support to 0. To solve the problem, we inject a probability ρ which is called *Distortion probability* into this approach. Distortion probability is used only when the column $j$ of the sanitization matrix $S$ contains only one "1". (i.e. $S_{ij} = 1$), and it works as follows: if $\sum_{k=1}^{m} D_{ik} \cdot S_{kj} \le 0 \quad \forall i, j \;\; 1 \le i \le n, 1 \le j \le m$ , $D'_{ij}$ has $\rho_j$ probability to be set to 1 and $1-\rho_j$ probability to be set to 0.

**Lemma 1**: *Given a minimum support σ, and a level of confidence c. Let {i, j} be a pattern in Marked-Set, $n_{ij}$ be the support count of {i, j}. ρ is the Distortion probability of column j. Without loss of generality, we assume that $S_{ij} = -1$. If ρ satisfies* $n_{ij} \times \rho < \sigma \times |D|$ *and* $\sum_{x=0}^{\lceil \sigma \times |D| \rceil - 1} \binom{n_{ij}}{x} \rho^x (1-\rho)^{n_{ij}-x} \ge c$, *where |D| is the number of transactions in D, we can say that we are c confident that {i, j} isn't frequent in D'.*

**Proof**: If probability policies are not considered and $S_{ij} = -1$, that is, while a row (transaction) $t$ of $D$ includes $D_{ti} = 1$ and $D_{tj} = 1$, according to the matrix multiplication discussed in the previous section, $D'_{tj}$ will be set to 0. We can view the $n_{ij}$ original $D'_{t_i j}$ , $D'_{t_1 j}, D'_{t_2 j}, \ldots D'_{t_{n_{ij}} j}$ , for each row $t_i$ contains *{i, j}* in $D$ as realizations of $n_{ij}$ independent identically distributed (i.i.d.) random variables $X_1, X_2, \cdots, X_{n_{ij}}$ , each with the same distribution as *Bernoulli distribution*, $B(1, \rho)$. The success is defined as $X_i = 1$ and the failure is defined as $X_i = 0$, $\forall\, i = 1$ to $n_{ij}$. The probability $\rho$ is attached to the success outcome and $1-\rho$ is attached to the failure outcome. Let $X$ be a random variable and could be viewed as the number of transactions which include *{i, j}* in $D'$, such that $X = X_1 + X_2 + \cdots + X_{n_{ij}}$ . Thus, the distribution of $X$ is the same as *Binomial distribution*, $X \sim B(n_{ij}, \rho)$. We get

$$P(X_{=x}) = \begin{cases} \binom{n_{ij}}{x} \rho^{x} (1-\rho)^{n_{ij}-x} & , x = 0, 1, \cdots, n_{ij} \\ 0 & , otherwise \end{cases}$$

, the mean of $X = n_{ij}\rho$, and the variance of $X = n_{ij}\rho(1-\rho)$.

If we want to hide *{i, j}* in $D'$, we must let its support in $D'$ be smaller than $\sigma$. Therefore, the expected value of $X$ must be smaller than $\sigma \times |D|$. That is, $\rho$ must satisfy $n_{ij} \times \rho < \sigma \times |D|$. Moreover, the probability of the number of success which is equal to or smaller than $\lceil \sigma \times |D| \rceil - 1$ should be higher than $c$. Therefore, $\rho$ must satisfy $\sum_{x=0}^{\lceil \sigma \times |D| \rceil - 1} \binom{n_{ij}}{x} \rho^{x} (1-\rho)^{n_{ij}-x} \ge c$ .

If $\rho$ satisfies the two equations, we can say that we are c confident that *{i, j}* isn't frequent in $D'$. ▓

When $n_{ij} > 30$, the Central Limit Theorem (C.L.T.) can be used to reduce the complexity of the equation and speed up the execution time of the sanitization process.

$$\sum_{x=0}^{\lceil \sigma \times |D| \rceil - 1} \binom{n_{ij}}{x} \rho^{x} (1-\rho)^{n_{ij}-x} \ge c \overset{bound}{\Rightarrow} \sum_{x=0}^{\lceil \sigma \times |D| \rceil - 1} \binom{n_{ij}}{x} \rho^{x} (1-\rho)^{n_{ij}-x} = c$$

$$\overset{C.L.T}{\Rightarrow} P\left( Z \le \frac{\lceil \sigma \times |D| \rceil - 1 - n_{ij}\rho}{\sqrt{n_{ij}\rho(1-\rho)}} \right) = c \Rightarrow \Phi\left( \frac{\lceil \sigma \times |D| \rceil - 1 - n_{ij}\rho}{\sqrt{n_{ij}\rho(1-\rho)}} \right) = c$$

Moreover, if several entries in column $j$ of $S$ are equal to –1, such as $S_{ij} = -1$, $S_{kj} = -1$, $S_{mj} = -1$ … etc, several candidate Distortion probabilities such as $\rho_i$ , $\rho_k$ , $\rho_m$ , etc are gotten. The Distortion probability of column $j$, $\rho_j$, is set to the minimal candidate Distortion probability to guarantee that all corresponding pair-subpatterns have at least $c$ confident to be hidden in $D'$.

## 3.3. Sanitization Algorithm

```
Sanitization Algorithm
Input: D, S, probabilities for each column of S
Output: D'
for ( i = 1 to n ){              // i is a row of D
    for ( j = 1 to m ){              // j is a column of S
        if ( $D_{ij} = 0$ )
            $D'_{ij} = 0$
        else{
            temp = $\sum_{k=1}^{m} D_{ik} \cdot S_{kj}$ .
            if ( column j in S contains some entries = –1, only $S_{jj} = 1$ and
            temp $\le 0$ )
                $D'_{ij}$ is set to 1 with probability $\rho_j$ and 0 with $1-\rho_j$.
            else if ( temp $\le 0$ )
                $D'_{ij}$ is set to 0.
            else if ( temp $\ge 1$ )
                $D'_{ij}$ is set to 1.
        }
    }
}
```

**Fig.7: Sanitization Algorithm.**

According to the probability policies discussed above, the sanitization algorithm $D'_{n\times m} = D_{n\times m} \times S_{m\times m}$ is showed in Fig.7.

Notice that, if the sanitization process causes all the entries in row $r$ of $D'$ equal to 0, randomly choose an entry from some $j$

where $D_{rj} = 1$, and set $D'_{rj} = 1$. It is to guarantee that the size of the sanitized database equals the size of the original database.

## 4. Performance Evaluation

### 4.1. Performance Quantifying

There are three potential errors in the problem. First of all, some sensitive patterns are hidden unsuccessfully. Secondly, some non-sensitive patterns cannot be mined from $D'$. And the third, new artificial patterns may be produced. However, the third condition never happens in both of our approach and SWA. Besides, we also introduce the other two criteria, dissimilarity and weakness. All of the criteria are discussed as follows:

Criterion 1: some sensitive patterns are frequent in $D'$. This condition is denoted as *Hiding Failure* [10], and measured by $HF = \frac{|P_H(D')|}{|P_H(D)|}$, where $|P_H(X)|$ represents the number of patterns contained in $P_H$ which is mined from database $X$.

Criterion 2: some non-sensitive patterns are hidden in $D'$. It is also denoted as *Misses Cost* [10], and measured by $MC = \frac{|\sim P_H(D)| - |\sim P_H(D')|}{|\sim P_H(D)|}$, where $|\sim P_H(X)|$ repress-ents the number of patterns contained in $\sim P_H$ which is mined from database $X$.

Criterion 3: the dissimilarity between the original and the sanitized database is also concerned, and it is measured by

$$Dis = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m}(D_{ij} - D'_{ij})}{\sum_{i=1}^{n}\sum_{j=1}^{m}D_{ij}}.$$

Criterion 4: according to the previous section, Forward-Inference Attack is avoided while at least one pair-subpattern of a sensitive pattern is hidden. The attack is quantified by

$$Weakness = \frac{|(P_H(D) - P_H(D')) \cap PairS(D')|}{|P_H(D) - P_H(D')|},$$

where $PairS(D')$ is the set of sensitive patterns whose pair-subsets can be completely mined from $D'$.

### 4.2. Experimental Results

**Table 1: Experiment factor**

| Factor | Default | Range | *C*: the level of confidence<br>*DS*: disclosure threshold of SWA<br>*WS*: window size of SWA<br>*RS*: ratio of sensitive pattern = $\frac{\textit{number of sensitive patterns}}{\textit{number of frequent patterns}}$<br>*RL2*: ratio of large2 in sensitive pattern set = $\frac{\textit{number of sensitive patterns generated from the seeds with length 2}}{\textit{number of sensitive patterns}}$ |
|---|---|---|---|
| *C* | 0.95 | — | |
| *DS* | 0.1 | — | |
| *WS* | 5000 | — | |
| *RS* | 0.1 | 0.0205 ~ 0.1006 | |
| *RL2* | 0.7 | 0.37844 ~ 1 | |

The experiment is to compare the performance between our approach and SWA which has been compare with Algo2a [4] and IGA [10] and is so far the algorithm with best performances published and presented in [12] as we know. The IBM synthetic data generator is used to generate experimental data. The dataset contains 1000 different items, with 100K transactions where the average length of each transaction is 15 items. The Apriori algorithm with minimum support = 1% is used to mine the dataset and 52964 frequent patterns are gotten.

Several sensitive patterns are randomly selected from the frequent patterns with length two to three items to be seeds. With the several seeds, all of their supersets are included into the sensitive patterns set since any pattern which contains sensitive patterns should also be sensitive.

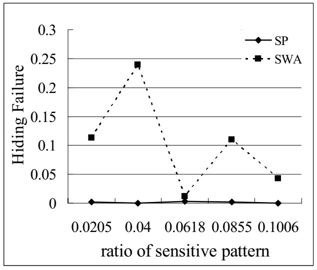


**Fig.8: Rel. bet. *RS* and *HF*.**

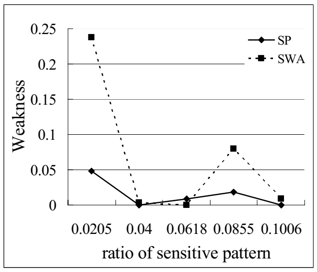


**Fig.9: Rel. bet. *RS* and *weakness.***

Fig.8, Fig.9, Fig.10, Fig.11 and Fig.12 show the effect of *RS* by comparing our work to SWA. Refer to Fig.8, because the level of confidence in our sanitization process takes the minimum support into account, no matter how the distribution of the sensitive patterns, we still are *C* confident to avoid hiding failure problems. However, there is no correlation between the disclosure threshold in SWA and the minimum support. Under the same disclosure threshold, if the frequencies of the sensitive patterns are high, the hiding failure will get high too. In Fig.9, because our work is to hide the sensitive patterns by decreasing the supports of the pair-subpatterns of the sensitive patterns, the value of weakness is related to the level of confidence. However, according to the disclosure threshold of SWA, when all the pair-subpatterns of the sensitive patterns have large frequencies, it may cause serious F-I Attack problems. Hiding failure and weakness of SWA change with the distributions of sensitive patterns.

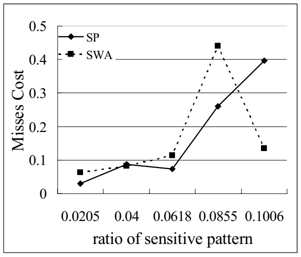


**Fig.10: Rel. bet. *RS* and *MC*.**

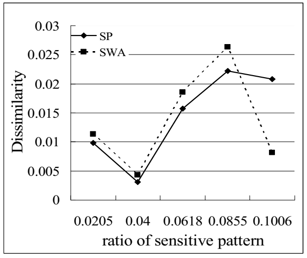
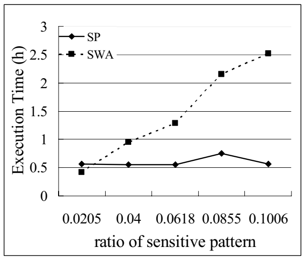


**Fig.11: Rel. bet. *RS* and *Dis.***

In Fig.10 and Fig.11, ideally, the misses cost and dissimilarity increase as the *RS* increases in our work and SWA. However, if the sensitive pattern set is composed of too many seeds, a lot of victim pair-subpatterns will be contained in Marked-Set. And as a result, cause a higher misses cost, such as x = 0.04 in Fig.10. Moreover, refer to the turning points of SWA under x = 0.0855 to 0.1006 in Fig.10 and Fig.11. The reason of violation is that, the result of the experiment has strong correlation with the distribution of the sensitive patterns. Because the sensitive patterns are chosen randomly, several variant factors of the sensitive patterns are not under control such as the frequencies of the sensitive patterns and the overlap between the sensitive patterns if the overlap between the sensitive patterns is high, some sensitive patterns can be hidden by removing a common item in SWA. Therefore, decrease the misses cost and the dissimilarity.



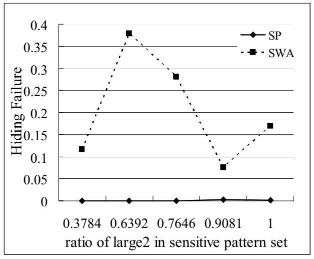
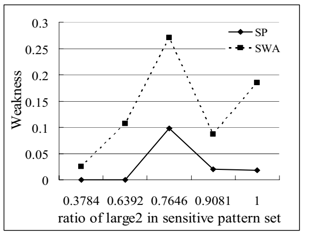


**Fig.12: Rel. bet. *RS* and time. Fig.13: Rel. bet. *RL2* and *HF*.**

Fig.12 shows the execution time of SWA and our approach. As shown in the result, the execution time of SWA increases as *RS* increases. On the other hand, our approach can be separated roughly into two parts, one is to get Marked-Set which is strong dependent on the data, the other is to set sanitization matrix and execute multiplication whose execution time is dependent on the numbers of transactions and items in the database. In the experiment, because the items and transactions are fixed, therefore, the execution time of our approach is decided by the setting of Marked-Set which makes the execution time changing slightly.



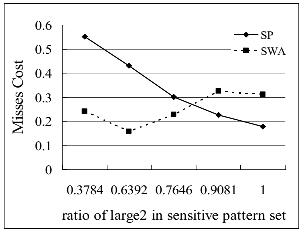


**Fig.14: Rel. bet. *RL2* and *weakness*. Fig.15: Rel. bet.*RL2* and *MC*.**

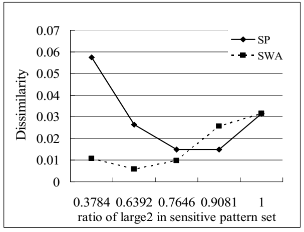


**Fig.16: Rel. bet. *RL2* and *Dis*.**

Fig.13, Fig.14, Fig.15and Fig.16 show the effect of *RL2*. Refer to Fig.13 and Fig.14, our work outperforms SWA no matter what *RL2* is. And our process is almost 0% hiding failure. The reason of the hiding failure of SWA is the same in Fig.8. Notice the result at x = 0.7646 in Fig.14, because the hiding failure is occurred at the seeds of the sensitive patterns, a high weakness is produced.

As shown in Fig.15 and Fig.16, the misses cost and dissimilarity of our work decreases as *RL2* increases. This is because the larger *RL2* is, the less effect on non-sensitive patterns. Also, weakness and dissimilarity of SWA are independent of *RL2*.

## 5. Conclusion

In the paper, a novel method improving the balance between sensitive knowledge protecting and discovery on frequent patterns has been proposed. By setting entries of a sanitization matrix to appropriate values and multiplying the original database by the matrix with some probability policies, a sanitized database is gotten. Moreover, it can avoid F-I Attack absolutely when the confidence level given by users approximates to 1. The experimental results revealed that although misses cost and dissimilarity between the original and sanitized database of our process are little more than SWA, ours provide more safely protection than SWA. Unlike SWA, our sanitization process could not suffer from F-I Attack and the probability policies in our approach also take the minimum support into account, the users only need to decide the confidence level which affects the degree of patterns hiding.

## 6. Reference

[1] M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim and V. Verykios, Disclosure Limitation of Sensitive Rules", Proc. of IEEE Knowledge and Data Engineering Exchange Workshop 1999.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. VLDB, Santiago, Chile, 1994.

[3] R. Agrawal and R. Srikant. Privacy preserving data mining. In ACM SIGMOD, Dallas, Texas, May 2000.

[4] E. Dasseni, V. Verykios, A. Elmagarmid and E. Bertino, Hiding Association Rules by Using Confidence and Support", Proc. of 4th Intl. Information Hiding Workshop (IHW), April 2001.

[5] A. Evfimievski, J. Gehrke, and R. Srikant. Limiting Privacy Breached in privacy preserving data mining. SIGMOD/PODS, 2003.

[6] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. KDD 2002.

[7] M. Kantarcioglu and C. Clifton. Privacy-preserving distributed mining of association rules on horizontally partitioned data. In ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, June 2002.

[8] Guanling Lee, Chien-Yu Chang and Arbee L.P Chen. Hiding sensitive patterns in association rules mining. The 28th Annual International Computer Software and Applications Conference.

[9] Y. Lindell and B. Pinkas. Privacy Preserving Data mining. In CRYPTO, pages 36-54, 2000.

[10] S. R. M. Oliveira and O. R. Zaïane. Privacy Preserving Frequent Itemset Mining. In Proc. of IEEE ICDM'02 Workshop on Privacy, Security, and Data Mining.

[11] S. R. M. Oliveira and O. R. Zaïane. Algorithms for Balancing Privacy and Knowledge Discovery in Association Rule Mining. IDEAS'03.

[12] S. R. M. Oliveira and O. R. Zaïane. Protecting Sensitive Knowledge By Data Sanitization, ICDM'03.

[13] S. R. M. Oliveira, O. R. Zaïane and Yücel Saygin. Secure Association Rule Sharing, PAKDD-04.

[14] Benny Pinks. Cryptographic Techniques For Privacy-Preserving Data Mining. ACM SIGKDD Explorations Newsletter Vol. 4, Is. 2, 2002

[15] S. J. Rizvi and J. R. Haritsa. Maintaining data privacy in association rule mining. VLDB, 2002.

[16] J. Vaidya and C. W. Clifton. Privacy preserving association rule mining in vertically partitioned data. KDD2002.

[17] Verykios, V.S.; Elmagarmid, A.K.; Bertino, E.; Saygin, Y.; Dasseni, E. Association rule hiding. IEEE Transactions On Knowledge And Data Engineering, Vol. 16, No. 4, April 2004.