K-anonymous Association Rule Hiding[*]

Zutao Zhu and Wenliang Du
Department of Electrical Engineering and Computer Science
Syracuse University, Syracuse, NY, USA 13244
{zuzhu,wedu}@syr.edu

ABSTRACT

In the paper we point out that the released dataset of an association rule hiding method may have severe privacy problem since they all achieve to minimize the side effects on the original dataset. We show that an attacker can discover the hidden sensitive association rules with high confidence when there is not enough "blindage". We give a detailed analysis of the attack and propose a novel association rule hiding metric, K-anonymous. Based on the K-anonymous metric, we present a framework to hide a group of sensitive association rules while it is guaranteed that the hidden rules are mixed with at least other K-1 rules in the specific region. Several heuristic algorithms are proposed to achieve the hiding process. Experiment results are reported to show the effectiveness and efficiency of the proposed approaches.

Categories and Subject Descriptors

K.4.1 [**COMPUTERS AND SOCIETY**]: Public Policy Issues—*Privacy*; H.2.8 [**DATABASE MANAGEMENT**]: Database Applications—*Data mining*

General Terms

Algorithm, Security

Keywords

Association Rule Hiding, k-anonymity

1. INTRODUCTION

Association rule mining [3,4,9] was introduced to discover strong patterns, for example, "ninety percent of customers who purchase bread also buy milk". Armed with this mining technique, a retail company can make decisions based on how its customers behave. Moreover, data sharing can gain mutual benefits to all participants. Data owners usually release their data as well as the mining parameters to other partners. However, these advanced technologies have increased the risks of disclosing the association rules that the owner considers sensitive when the dataset is shared with other organizations.

To address the problem of preventing the sensitive association rules from being disclosed, researchers have studied methods for Association Rule Hiding (ARH) [7, 8, 11, 12, 18–21]. In general, existing approaches sanitize the original dataset such that the sensitive rules cannot be discovered in the released dataset while preserving as much knowledge as possible using the same minimum confidence threshold (MCT) and minimum support threshold (MST), even if the dataset is shared with other parties.

Example 1: consider that a company wants to distribute its transaction dataset D in Figure 1 to other parties. D has 24 transactions. TID is the index for the transactions. Items is the transaction. The frequent itemsets with support larger than 9 are: DB(10), D(12), HA(10), H(13), IB(10), I(15), A(14), and B(15). The number in the parentheses is the support value for the itemset. t_3 (TID = 3) *fully supports* AGH and *partially supports* EG. The *support* of an itemset X is defined as the number of transactions that fully support X, which is denoted as $Supp(X)$. The company uses association rule mining tool to mine the rules using MST (10) and MCT (76.9%). $D \Rightarrow B$ (Support: 10, Confidence: 83.3%), and $H \Rightarrow A$ (Support: 10, Confidence: 76.9%) are the two strong rules. The *generating set* for the rule $D \Rightarrow B$ is DB. The company finds that the rule $H \Rightarrow A$ is sensitive and wants to hide it. Adopting an existing algorithm, the publisher produces the release dataset D' by removing an item "H" in the fourth transaction of D. The rule $H \Rightarrow A$ is hidden because either its confidence (75%) is less than MCT or its support (9) is less than MST in dataset D'. Using the same MST and MCT, we can only get one rule, that is, D ⇒ B. All existing hiding algorithms try to break the two conditions for an association rule by reducing either the support or the confidence of the sensitive rules.

2. ISOLATION ATTACK

We use a rectangular coordinate system to demonstrate the hiding process in Figure 2. The x-axis represents the support of the association rule while the y-axis represents the confidence of the association rule. A point (s, c) in the system is a rule whose support value is s and whose confidence value is c. The set of association rules from dataset D with MST s and MCT c is denoted as $\xi(D, s, c)$. Any rule in $\xi(D, s, c)$ is called a (s, c)-strong rule with respect to D. Therefore, the (S, C)-strong rules are within the zone Z_1.

After applying the association rule hiding algorithms, the sensitive rule $r : X \Rightarrow Y$, originally in zone Z_1, falls into the zone Z_2, which is between solid lines and the dotted lines.

[*]This work was partially supported by Awards No. 0430252 and 0618680 from the United States National Science Foundation.

ASIACCS'10 April 13–16, 2010, Beijing, China.
Copyright 2010 ACM 978-1-60558-936-7/10/04 ...$10.00.

Based on the association rule hiding algorithm parameters MST (S) and MCT (C), the adversaries can deduce that the sensitive rules will fall in a certain region. For example, if the adversaries know that the hiding algorithm is to decrease the support of the sensitive rules, and the hiding process needs to minimize the side effect, they can learn that the support for the sensitive rules will be the maximum integer that is less than the given MST. If there is only one rule whose support is equal to the maximum integer in the sanitized dataset, the hidden rule can be identified by the adversaries with 100% confidence. The scenario is like an isolated island in the map which makes it easy to be identified. We call it the *isolation attack*.

TID	Items	TID	Items	TID	Items
1	ABCDHI	9	BDEGH	17	ADH
2	BCDFHI	10	AHIJ	18	AHIJ
3	AGHIJ	11	BCDIJ	19	BCDIJ
4	ABDEGHI	12	ABDIJ	20	BCDEGI
5	AHJ	13	ADEFH	21	AB
6	ABCDEHI	14	BDEG	22	BC
7	ABCFI	15	BEGH	23	BEI
8	AGHJ	16	ACEI	24	I

Figure 1: Original dataset D

To the best of our knowledge, none of the existing ARH algorithms have addressed this type of attack.

The Confidence and Support Lower Bounds of Hidden Rules. Based on the "minimal impact" principle, we can derive two lower bounds regarding the support value and the confidence value of the sensitive rules after the hiding process. Note that the analysis here is based on the assumptions in [19], that is, the sensitive rules are disjoint, and we hide the rule one unit at a time.

Theorem 1 shows the lower bound of the support of the hidden sensitive rule.

THEOREM 1. *Given MST s and MCT c, the lower bound of the support $s_\perp$ for the hidden sensitive rules in D' is $s-1$.*

It is quite straightforward for the lower bound of the support. However, calculating the lower bound of the confidence of the hidden rule is different. The proof of the theorem is omitted due to the page limit.

THEOREM 2. *Given MST s and MCT c, when adopting confidence based hiding approach, the lower bound of the confidence value $c_\perp$ for the hidden sensitive rules is $(c - \frac{1}{s})$.*

K-anonymous. Given the hiding parameter s and c, let $s_\perp$ be (s-1) and $c_\perp$ be $(c - \frac{1}{s})$. The *cloak zone M* of a sanitized dataset D' is the difference between $\xi(D', s_\perp, c_\perp)$ and $\xi(D', s, c)$. The cloak zone is exactly the area where the region between the dotted lines and the solid lines is in Figure 2. We have to point out that there may be other rules rather than the hidden ones in the cloak zone.

An association rule hiding algorithm has *K-anonymous* property if and only if the number of rules (called size) in the cloak zone M is at least K. K-anonymous concept is widely used in the field of Privacy-Preserving Data Publishing (PPDP). Generally speaking, in PPDP, the K-anonymous dataset guarantees that an adversary cannot link sensitive information to a group of less than K individuals, or in other words, the probability that the adversaries can link an individual to the sensitive information is not more than $\frac{1}{K}$. Similar to the original definition, our definition also guarantees that no hidden sensitive rules can be linked to a group of less than K association rules. Remember that the lower boundaries of the cloak zone are calculated based on the assumptions made in [19]. The dataset will be over-sanitized without these assumptions than that with them. Without them, some hidden sensitive rules may fall out of the cloak zone. If the number of rules in the cloak zone is not less than K, we guarantee that the probability that the adversary can guess a sensitive rule is lower than $\frac{1}{K}$.

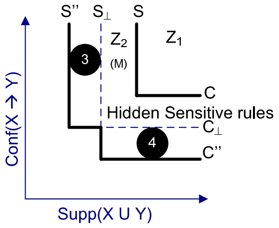


Figure 2: Our approach

Existing hiding algorithms have no guarantee that the rules in the cloak zone is larger than or equal to K. In Example 1, D' does not satisfy 2-anonymity.

3. PROBLEM FORMULATION

We make two assumptions in the paper. First, the hiding parameters, MST and MCT, are available to the adversaries. Usually, they are published with the released dataset. Second, the adversaries know that there is at least one sensitive rule to be hidden. Having the two assumptions, the adversaries can infer that either the supports of the hidden sensitive rules are lower than MST or the confidences of them are lower than MCT.

Hidden sensitive rules can be identified with a high confidence when the released dataset does not have a large enough cloak zone M. An association rule hiding approach is *K-anonymous* if the number of rules in the cloak zone is at least K in the released dataset of the approach. Given the K-anonymous metric, we can formally formulate our problem: *given a dataset D, MST, MCT, K, and a subset R_H of (MST, MCT)-strong rules, transform D to D' such that every rule in R_H is not (MST, MCT)-strong rule for D' and there are at least K rules in the cloak zone.*

4. K-ANONYMOUS HIDING

In this section, we propose our K-anonymous association rule hiding approach.

We use Figure 2 to intuitively show how our approach, post-sanitization, works. Using existing association rule hiding algorithms, we transform D to D_{hide}, and move the sensitive rules from zone Z_1 to zone Z_2. If D_{hide} does not satisfy K-anonymous, we obtain the *blindage rules* from either zone 3 or zone 4 in the figure. The rules in zone 3 is $\xi(D', s'', c_\perp)$ - $\xi(D', s_\perp, c_\perp)$, where s'' is less than $s_\perp$. By increasing their support or confidence, the selected rules can move to the cloak zone M (same as Z_2) such that the number of rules in M increases. If the sanitized dataset does not satisfy K-anonymity, we promote K blindage rules into the cloak zone instead of making the number of rules in the cloak zone to be K. If we choose to let the number of rules in the cloak zone to be K, we may end up with less than K rules in the zone when some rules fall out of the cloak zone in the sanitization.

Adopting post-sanitization, we can take advantage of the existing association rule hiding methods. Moreover, we can reduce the impact for the dataset. By carefully choosing the blindage rules, we will not affect any hidden sensitive rule.

Step 1: Generate Blindage Rules.

The main purpose of K-anonymous association rule hiding is to hide the sensitive rules while maintaining the K-anonymous property of the cloak zone. We have to guarantee that the sensitive rules are still hidden after further sanitizing. No matter how we sanitize D_{hide}, we cannot make a sensitive rule to become a (MST, MCT)-strong rule.

We observe that when adding or removing items that are not an element of the generating set of the sensitive rule, the support and confidence of the rule are not changed. Therefore, we always choose the blindage rules that are disjoint with the sensitive rules. Based on the same reasoning, we also require that the blindage rules are disjoint with each other, that is, for every two generating sets of the blindage rules, they are not overlapped. By doing so, increasing the support (or confidence) of one blindage rule will not affect the other blindage rule. Therefore, it leads to the binary integer programming approach.

The Binary Integer Programming Approach. We find a maximum cardinality subset of all the non-strong rules in zone 3 or zone 4 of Figure 2, in which the rules are disjoint with each other. If the cardinality of the subset is larger than K, we can pick K from it as the blindage rules. Otherwise, we cannot find such a disjoint set. In the latter case, we will develop heuristic algorithms to generate blindage rules and defer it to the future work.

The first step can be formulated as an binary integer programming (BIP) optimization problem: given a set of rules S, find the maximum cardinality subset S_m of S such that any two rules in S_m are disjoint.

We solve the problem in three steps. The first step is to define variables x_i (i = 1, ..., $|S|$), which will be 1 if the i-th rule is selected into the result subset, and 0 otherwise. The second one is to build the buckets and place the rules into them. For each distinct item in S, we build a bucket. The set of buckets is denoted as B. For each rule, we put it into the buckets according to the items it supports. We use B_j to denote the j-th bucket. The third step is to derive the constraints and object function as the following:

$$\textbf{maximize}(\sum_{i=1}^{|S|} x_i)$$

$$\text{subject to:} \quad (\sum_{x_j \in B_k} x_j) \leq 1 \qquad \forall B_k \in B, \qquad (1)$$

$$x_i \in \{0,1\} \qquad \forall i \in \{1, ..., |S|\}. \qquad (2)$$

The objective function maximizes the number of rules included. Constraint (1) states that no more than one rule can be selected from the same bucket because these rules are overlapped. Constraint (2) imposes the binary requirement on all x_i variables.

After solving the above BIP problem, we get a candidate set consisting of those x_i whose value is 1. We can simply pick K rules from the candidate set as the blindage rules if the size of candidate set is at least K. However, if the size of the candidate set is less than K, we cannot find enough disjoint blindage rules.

We use an example to demonstrate the BIP method. We have three rules: r_1: $A \Rightarrow B$, r_2: $BC \Rightarrow D$ and r_3: $D \Rightarrow C$. We have four distinct items (ABCD), therefore we build four buckets: B_1 for A, B_2 for B, etc. We have three variables x_1, x_2, x_3 because there are three rules. x_i corresponds to r_i (i=1,2,3). B_1 has x_1; B_2 has x_1 and x_2; B_3 has x_2 and x_3; B_4 has x_2 and x_3. x_1 is placed into B_1 and B_2 because it supports both A and B. Therefore, our objective function is to maximize $x_1 + x_2 + x_3$ and the constraints are: $x_1 \leq 1$, $x_1 + x_2 \leq 1$, $x_2 + x_3 \leq 1$, $x_2 + x_3 \leq 1$, and all variables are either 0 or 1. Using BIP solver, we can get the optimal solution, that is, x_1, x_2, x_3 are 1, 0, 1, respectively. The objective function is evaluated to be 2. If K is 2, we pick r_1 and r_3 as the blindage rules.

Step 2: Association Rule Cloaking.

After we produce the blindage rules, we have to increase the support (or confidence) value of the blindage rules such that these rules enter the cloak zone. Therefore, the number of rules in the cloak zone increases. We call this process *cloaking*. The association rule cloaking algorithms can be either support-based or confidence-based depending on from which zone the blindage rules are produced. If they are from zone 3, we use support-based cloaking; if they are from zone 4, we use confidence-based cloaking.

In both approaches, we cloak the blindage rules one after another. For each blindage rule we increase the support or confidence only one unit at a time. For instance, we increase the support of the blindage rule by one in each iteration. If we use confidence-based approach, we either increase $Supp(X \cup Y)$ by one or decrease $Supp(X)$ by one. By doing in this way, we can minimize the side effect. We would not sanitize the dataset more than it is necessary to do so. In each sanitization iteration, we first choose which transaction to be operated on. Second, we determine which items to be added into or removed from the transaction. We proposed heuristic algorithms to promote the blindage rules into the cloak zone while minimizing the side effects on the information that is not sensitive.

Cloak by Support. We have two ways to increase the support values of the blindage rules. One is to make some transactions that fully support X and partially support Y to fully support the generating set $X \cup Y$, that is, we add the missing items of Y to the selected transactions. It should be pointed out that the confidence for the blindage rule increases because $Supp(X)$ is constant and $Supp(X \cup Y)$ increases. As long as we can guarantee that $Supp(X \cup Y)$ is less than MST, the blindage rule cannot enter the strong zone Z_1. We call this algorithm *incrX*.

The other is to make those transactions that fully support Y and partially support X to fully support $X \cup Y$. Under such modification, $Supp(X)$ and $Supp(X \cup Y)$ increase by one simultaneously, which makes $Conf(X \Rightarrow Y)$ ($Supp(X \cup Y)/Supp(X)$) increases. Same as the former method, the blindage rule is outside of the strong zone Z_1 if only $Supp(X \cup Y)$ is less than MST. We call the algorithm *incrY*.

The incrX algorithm is shown in Figure 3. The basic heuristic is that the less items a transaction has, the less association rules would be effected when adding items into it. We first scan the sanitized dataset D_{hide} to find those transactions that fully support Y and partially support X. The transactions are then sorted descendently according to the number of items in X that they have. The more items in X a transaction have, the less number of items we need to add to the transaction. At last, we change the sanitized dataset by adding the missing items of X to the transaction.

We demonstrate algorithm incrX with an example. $DB \Rightarrow I$ is the selected blindage rule for Example 1. In order to in-

```
Input: a set $R_B$ of rules to cloak, a dataset $D_{hide}$, the $MCT$, the $MST$,
       the $MCT_\perp$ and the $MST_\perp$
Output: the sanitized dataset $D'_{hide}$ from $D_{hide}$ such that the rules in
        $R_B$ are in the cloak zone of $D'_{hide}$
begin
    sort the rules in $R_B$ in descendent order of the support values;
    foreach rule $r \in R_B : X \Rightarrow Y$ do
        $T_X = \{t \in D$: $t$ fully supports $Y$ and partially supports $X\}$;
        count the number of items in $X$ for every transaction of $T_X$;
        sort $T_X$ in descending order of the number of items in $X$;
        Iterations $= MST_\perp - Supp(X \cup Y)$;
        for $i$ = 1 to Iterations do
            t = $T_X[0]$;
            add to $t$ all the missing items in $X$;
            remove $t$ from $T_X$;
        end
        $R_B = R_B$ - r;
    end
end
```

Figure 3: Cloak by support (incrX)

crease the support of the rule, we can select the transaction t=$< 23, BEI >$ which partially supports DB and fully supports I and insert an item D to t such that it fully supports DBI. The rule $DB \Rightarrow I$ appears in the cloak zone (support = 9, confidence = 81.8%).

Algorithm incrY is similar to Algorithm incrX with the only changes, replacing X with Y and replacing Y with X. No matter which approaches we use, we have to scan the dataset to find out those transactions that fully or partially support the generating set of a blindage rule. The scanning iterations depend on the number of blindage rules.

5. EXPERIMENTS

We experimentally evaluate the effectiveness and efficiency of our K-anonymous association rule hiding technique.

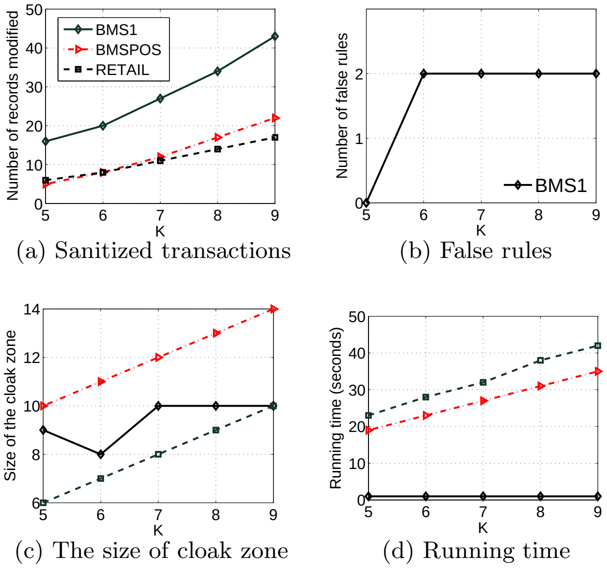
The three datasets we use in the experiments are downloaded from the FIMI repository [1]. They are BMS-WebView-1 (bms1), BMS-POS (bmspos), and retail. The characteristics of the real datasets are listed in Figure 4. Usually, when people use association rule mining, they like to find the association rules with high confidence. That is why we setup relatively high MCT value.

Database name	Num. of transactions	Num. of items	Avg. trans. length	MST	MCT	Size of cloak zone
bms1	59,602	497	2.50	136	80%	4
bms-pos	515,597	1,657	7.50	2578	80%	5
retail	88,162	16,470	10.30	441	90%	1

Figure 4: Characteristics of Datasets

We first generate the association rules based on the MST and MCT in Figure 4 for the dataset. After that, we randomly pick one rule as the sensitive rule. We use one support-based or confidence-based association rule hiding algorithms described in [19] to hide the sensitive rule. Once we have the disguised dataset D_{hide}, we generate the association rule set Q for MST and MCT and the association rule set P for $MST_\perp$ and $MCT_\perp$. The isolation attack is performed by checking the rules in the cloak zone. The dataset publisher provides the value of K which is served as the degree of anonymity for D_{hide}. Based on the cloaking heuristic, K rules that are not in P are picked as the *blindage* for the sensitive rule. As is described in Section 4, the performance of producing the blindage rules is fixed. We need to mine the sanitized dataset twice. Calculate the difference. For the BIP approach, we use MOSEK [2] to solve the optimization problem. It is fast enough that we would ignore the performance of this part. Finally, we apply incrX algorithms on D_{hide}. Therefore, we get the release dataset D'_{hide}. From Figure 4, we can see that the size of the cloak zone is no larger than 5. We can even identify the hidden sensitive rules for `retail` dataset with 100% confidence because the size of the cloak zone is one. In the following experiments, we vary K from 5 to 9 for each dataset to achieve K-anonymous.

Measurements. We need to measure the side effects of the released dataset D'_{hide}. Following the convention in [6, 14, 19, 21], we consider *false rule* (FR) (the rule that is falsely generated as a (MST, MCT)-strong rule in the hiding process), and *lost rule* (LR) (non-sensitive (MST, MCT)-strong rules that are falsely hidden). The lower value they are, the better our algorithms performs. Furthermore, we measure the size of the cloak zone after using our hiding process. The larger the size of the cloak zone, the lower probability that the adversaries can identify the hidden sensitive rule. In the ideal condition, the number of FR as well as LR is zero while the size of the cloak zone is the summation of K and the number of rules that are originally in the cloak zone. It is worth to note that the hidden sensitive rules are still hidden as is discussed in Section 4. In addition to the above three measures, we take into consideration the number of transactions affected by the cloaking process. At the end, we measure the CPU time of the cloaking process to evaluate the efficiency of our method.



(a) Sanitized transactions

(b) False rules

(c) The size of cloak zone

(d) Running time

Figure 5: Experiment results

Performance Evaluation of incrX. The side effect evaluation and the time consumed for algorithm incrX are depicted in Figure 5(a)-5(d). The number of modified transactions increases when K increases as is shown in Figure 5(a). We have to add more items to the transactions when the number of blindage rules increases. No strong rules are lost for all three datasets. Dataset `bms1` has two false rules when K is larger than 5 in Figure 5(b). The reason is that adding items to a transaction leads to the support of other non-strong rule also increases, which makes the rule to disappear in the cloak zone and become a strong rule. The other two datasets do not have any false rule.

When it comes to the size of the cloak zone in Figure 5(c), `bmspos` and `retail` datasets behave as what we expect, that is, it increases when K increases. The cloak zone size for dataset `bms1` does not follow this trend. The size when K=5 is larger than that of K=6. It is because adding to some transactions items of X in one blindage rule increases Supp(X') of other non-blindage rule in the cloak zone (X' is the antecedent of that rule). Therefore, the size of the cloak zone decreases. However, the size of the cloak zone is larger than or equal to K for three datasets.

The time needed by algorithm incrX increases proportionally to the value of K, as is shown in Figure 5(d). It is in accordance to the analysis.

6. RELATED WORK AND CONCLUSIONS

Atallah et al. [5] first studied the problem of hiding association rules. They proved that finding the optimal sanitization solution to hide association rules for a dataset is NP-hard, and proposed a heuristic approach to hide the sensitive rules by deleting items from the transactions that support the generating itemsets of the sensitive rules. Dasseni et al. [7] proposed three heuristic algorithms by reducing either the support or the confidence of the sensitive rules. It was further extended in [19].Some frequent itemset hiding algorithms are proposed which can be used to hide association rules, including [13]. However, these works hide the sensitive rules until their supports or confidences are less than the given threshold, which make them vulnerable to the isolation attack. The work [17] replaced items with "unknown" rather than deleting them to hide the association rules. They modified the definition of MST and MCT, which made the current association rule mining tools unusable. Unlike theirs, our approach can take advantage of existing mining tools.

Wu et al. [21] eliminated the assumption that required all sensitive rules disjoint in [19]. However, they provided no guarantees to hide all pre-selected sensitive rules. This is not allowed when the sensitive rules can lead to huge profit lost for the company. Our approach guarantees that the sensitive rules are hidden after the hiding process.

In [11], the hiding of the frequent itemset was formulated as an integer programming problem. The object function was to minimize the number of transactions that were altered. The work in [8] was also an integer programming approach. They could drastically reduce the number of constraints, which made the computation much faster. The border-based approach is demonstrated in [12, 18]. These works make use of the border theory [10] to determine the set of the association rules that are affected. These approaches also have no guarantee for the K-anonymous. Researchers also propose frameworks to measure the utility of the result of the hiding process [6, 14].

K-anonymity [15, 16] has been extensively studied in the field of PPDP. We apply the idea into association rule hiding process.

In this paper we have shown that the released dataset of association rule hiding is vulnerable to the isolation attack. We introduce K-anonymity for the association rule hiding, a principle that has a stronger privacy guarantee. A framework to achieve K-anonymous association rule hiding is presented. Binary integer programming approach is presented to produce the blindage rules. And two heuristic cloaking algorithms are elaborated to mix the blindage rules with the hidden sensitive rules in the cloak zone. Experiments have shown that our proposed method generates the sanitized datasets with little side effect.

There are several directions to follow up. We want to extend the techniques to remove the requirement that the blindage rule be disjoint with each other. Moreover, we want to find an optimal solution to the K-anonymous association rule hiding such that the impact on the dataset is minimal.

7. REFERENCES

[1] URL: http://fimi.cs.helsinki.fi/.

[2] URL: http://www.mosek.com/.

[3] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *SIGMOD'93*.

[4] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *VLDB'94*.

[5] M. Atallah, A. Elmagarmid, M. Ibrahim, E. Bertino, and V. Verykios. Disclosure limitation of sensitive rules. *KDEX'99*.

[6] E. Bertino, I. Fovino, and L. Provenza. A framework for evaluating privacy preserving data mining algorithms. *Data Mining and Knowledge Discovery*, 11:121–154(34), September 2005.

[7] E. Dasseni, V. S. Verykios, A. K. Elmagarmid, and E. Bertino. Hiding association rules by using confidence and support. In *Information Hiding 2001*. Springer-Verlag.

[8] A. Gkoulalas-Divanis and V. S. Verykios. An integer programming approach for frequent itemset hiding. In *CIKM'06*.

[9] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *SIGMOD'00*.

[10] H. Mannila and H. Toivonen. Levelwise search and borders of theories in knowledge discovery. *Data Min. Knowl. Discov.*, 1(3):241–258, 1997.

[11] S. Menon, S. Sarkar, and S. Mukherjee. Maximizing accuracy of shared databases when concealing sensitive patterns. *Info. Sys. Research*, 16(3):256–270, 2005.

[12] G. V. Moustakides and V. S. Verykios. A maxmin approach for hiding frequent itemsets. *Data Knowl. Eng.*, 65(1):75–89, 2008.

[13] S. R. M. Oliveira and O. R. Zaiane. Privacy preserving frequent itemset mining. In *PSDM'02*.

[14] S. R. M. Oliveira and O. R. Zaiane. A unified framework for protecting sensitive association rules in business collaboration. *Int. J. Bus. Intell. Data Min.*, 1(3):247–287, 2006.

[15] P. Samarati. Protecting respondents' identities in microdata release. *IEEE TKDE*, (6), 2001.

[16] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, 1998.

[17] Y. Saygin, V. S. Verykios, and C. Clifton. Using unknowns to prevent discovery of association rules. *SIGMOD Rec.*, 30(4):45–54, 2001.

[18] X. Sun and P. S. Yu. A border-based approach for hiding sensitive frequent itemsets. In *ICDM'05*.

[19] V. S. Verykios, A. K. Elmagarmid, E. Bertino, Y. Saygin, and E. Dasseni. Association rule hiding. *IEEE TKDE*, 16, 2004.

[20] V. S. Verykios and A. Gkoulalas-Divanis. *A Survey of Association Rule Hiding Methods for Privacy*. Springer US, 2008.

[21] Y.-H. Wu, C.-M. Chiang, and A. L. Chen. Hiding sensitive association rules with limited side effects. *TKDE*, 19(1), 2007. Senior Member-Arbee L. P. Chen.