A Model of Privacy Preserving in Dynamic Set-valued Data Re-publication

Dan Wang, Yi Wu, Wenbing Zhao, Lihua Fu

College of Computer Science and Technology, Beijing University of Technology, China
neuwd@sina.com, wuyi@emails.bjut.edu.cn, {zhaowb, fulh}@bjut.edu.cn

Abstract

A model of privacy preserving in dynamic set-valued data re-publication is studied in this paper. Dynamic data in most practical applications may be re-published after updating, and the sensitive information of which may confront the risk of being exposed by adversary using historical publish results. A novel k-preserving model is proposed to protect data privacy from being exposed by continuing using transactional k-anonymity and maintaining the diversity and continuity of sensitive elements in the dataset during re-publication. An anonymous algorithm is also proposed to reduce information loss of the anonymous result by integrating local generalization with suppression technique. Real-world datasets are used in the experiment, the results and evaluations demonstrate that the approach in this paper can prevent privacy disclosure effectively and acquire publishing result with better availability.

Keywords: Privacy preserving, Transactional k-anonymity, Set-valued data, Dynamic dataset, Re-publication

1 Introduction

With the rapid development of information technology, the kinds of type and quantity of data shared and published in the open computing network increase dramatically. Great convenience is provided by high degree of data collection, sharing and publication for cooperation and research among various organizations. However, sensitive information may be revealed if the original dataset is published directly [1, 19-20]. Set-valued data [2], known as transactional data, is one of representative type of data in publishing and has the form of (id, $\{elem_1, elem_2, \ldots, elem_n\}$). The id of each record is associated with a set of elements [3]. Supermarket shopping list in which $\{elem_1, elem_2, \ldots, elem_n\}$ represents a set of items that the customer with the id purchased, is an typical examples of set-valued data as well as medical diagnosis, web query logs, movie ratings and recommendation data. Large amount of set-valued data is continually published since it is one of extremely important source for knowledge discovery and data mining [4].

However, set-valued data publishing faces the same problem with relational data [5] and multiple sensitive data [6] that personal privacy and sensitive information may be exposed during publication. Besides, the sensitive attributes of set-valued record may also have the characteristics of in-distinguishability and unfixed quantity [3]. Supposing a shopping center has published some data without identifier of customers for business analysis, some commodities, such as health drugs, may involve sensitive information of certain customers. If the adversary gets part of commodities of a certain customer by seeing the top of his/her shopping cart, and uses background knowledge to analyze the published data, it is possible for the adversary to infer the complete shopping records of the customer. Thus, personal privacy is revealed.

On the other hand, data change continuously due to insertion, deletion and modification in practical applications; it has to re-publish the data to keep up with the updates. That may also lead to disclose sensitive information because of the association between the data re-published before and after. The characteristics of set-valued data may bring more uncertainty into the update-repuplish process.

Therefore, it should be pay more attention to privacy preserving in re-publication of dynamic set-valued data. Several works have presented good solutions for anonymous protection of set-valued data, like *transactional k-anonymity* [3], which prevents adversary from associating a record with the corresponding individual under eliminating the distinction between sensitive and non-sensitive attributes. However, both of generalization [7] and suppression [8] anonymous algorithm suffer from higher information loss or less efficient. The study of privacy preserving in re-publication of dynamic set-valued data remains to be developed. Sensitive attribute Update Graph (*SUG*) theory and *m-Distinct* [9] are strict privacy protection methods for traditional relational data. They can prevent privacy in the dataset that simultaneously existing insertion, deletion and

*Corresponding Author: Dan Wang; E-mail: neuwd@sina.com
DOI: 10.3966/160792642019012001013

modification from being exposed. Moreover, they all have a good scalability.

Motivated by prior works, we consider that *SUG* theory which has good applicability and scalability can be introduced into privacy preserving in re-publication of dynamic set-valued data. Moreover, unnecessary information loss always exists if we use only one of the existing anonymous methods. Therefore, we think it is a non-trivial problem that proposing a dynamic set-valued data re-publication model. Integrating generalization with suppression technique to improve the anonymous algorithm and enforce the anonymous result satisfies transactional k-anonymity with less data distortion and better availability is also important.

The main contribution of this paper is as follows:

(1) Proposed a privacy preserving model for re-publication of dynamic set-valued data based on extending SUG theory.

(2) Proposed an algorithm of integrating local generalization with suppression to conform to transactional k-anonymity and reduce the information loss of anonymous result as well.

(3) Proposed an algorithm of dynamic set-valued dataset re-publication.

The rest of the paper is organized as follows. Section 2 introduces the related works. Section 3 proposes the privacy preserving model of dynamic set-valued data re-publication. Section 4 introduces the re-publication algorithm. Section 5 presents an experimental evaluation of our study and compares it with existing state-of-art. Section 6 concludes this paper.

2 Related Works

In relational data publication, elements of a tuple can be divided into identifier, quasi-identifiers and sensitive attributes. However, the quantity of elements in the record of set-valued data is unfixed, and it is difficult to distinguish between these kinds of attributes exactly. Terrovitis et al. [2, 10-11] first considered this characteristics and proposed k^m*-anonymity*, which required that every subset of no more than m elements in a given dataset should be contained in at least k records. K^m*-anonymity* can be denoted as k^∞*-anonymity* by setting m as the greatest length of all the records. However, it was hard to determine the value of m, and it may produce a higher information loss. With the increase of parameter m, the time performance also decreased rapidly.

K-anonymity was first proposed in [1] for relational data privacy protecting, He and Naughton [3] introduced *k-anonymity* to set-value data and proposed transactional *k-anonymity*. It demanded that every record should exist at least k times in the dataset, and meant that the size of each equivalence class was at least k. A top-down greedy partition algorithm based on local recoding generalization was also provided in [3], in which Generalization Hierarchy was used to decide which records were similar and should be grouped together. The Partition algorithm had a good time performance, while it anonymized the sub-partitions that dissatisfy transactional *k-anonymity* by former generalization values during the partition process. This roll back process led to high generalization level but finally resulted in high information loss of the anonymous dataset.

Motwani and Nabar [12] proposed a two-phase algorithm approximate without GH and satisfy transactional *k-anonymity*. It was based on the concept of flipping on the suppressed elements under elimination of sensitive and non-sensitive attributes. But this algorithm showed poor performance for greater k and sparse data. An improvement on approximate was made by [13], it presented a new measurement to estimate the number of insertion and deletion operations required to achieve transactional k-anonymity. This suppression algorithm can also achieve *O(log k)-approximation*. An integrating method was studied in [14] to compensate the insufficient of global recoding generalization and suppression techniques and achieve k^m*-anonymity*.

In the study of dynamic relational data re-publication, Wang and Fug [15] first studied the problem of securely publishing multi-shots of a static dataset. They proposed a solution to properly anonymize the current publication so as to control possible inferences. Fung et al. [16] proposed a model that using *k-anonymity* continuously to resist privacy exposing lead by growth of dataset. However, it cannot deal with record deletion and modification.

Xiao and Tao [17] conducted anonymization on datasets that updated by both record insertion and deletion. They proposed m-Invariance to prevent privacy disclosure by guaranteeing the equivalence class to which a record belongs contains the same set of sensitive attributes all the time. However, *m-Invariance* did not consider record modification. Li and Zhou [9] addressed the problem of both internal and external updates, presented the general privacy disclosure framework SUG, which is applicable to all anonymous re-publication problems. They also proposed a counterfeited generalization principle called *m-Distinct* to anonymize dataset with both external and internal updates effectively. An algorithm to generalize dataset to meet *m-Distinct* was also developed in [9]. *M-Distinct* has good scalability and can ensure sensitive information not to be revealed by continuously updates and re-publication. But the privacy preserving method for dynamic set-valued data remains to be studied.

3 Anonymous Re-publication Model

Detailed definitions of dynamic relational data and its updates were carried out in [9]. Considering the differences between relational and set-valued data,

expanded concepts and definitions of set-valued data are given first in this paper.

3.1 Definition of Dynamic Set-valued Data

Suppose I={I_1, I_2, ..., $I_{|I|}$} is the set of items from which the elements of the sets are drawn, and assume D={r_1, r_2, ..., $r_{|D|}$} is a set-valued dataset over I that is going to be published, where each record ri$\in D$ is a non-empty subset of I. For any element e, record r's value on e is represented as $r_{[e]}$. Accordingly, the anonymous result is denoted by D^* and the anonymous record by r^*. If there are same anonymous elements in several records of D^*, these records form an equivalence class g. If $r \in g$, we denote r's Candidate Attribute Set as the set of all of elements in g.

Generally, we say a set-valued dataset is dynamic if it changes from time to time through both internal and external update. Internal update means modification occurs on elements, and external update means record insertion and deletion.

3.2 Sensitive Attribute Update Graph and Privacy Preserving Theory

M-Distinct is effective on anonymous re-publication of dynamic relational data with good scalability, it guarantees sensitive information will not be revealed during both external and internal updates. Therefore, we propose a novel anonymous model for dynamic set-valued data re-publication by extending SUG theory.

As mentioned in [9], even if sensitive information has been well protected in each separate publication, due to the connection between original data and the publishing result, sensitive information can still be exposed by analyzing SUG. The main idea of SUG is to represent all the possible sensitive attributes and updates of a record in a graph. Each node in the graph represents a possible sensitive attribute. Each directed edge represents a feasible update on a sensitive attribute. Only if sensitive value si can be sure update to s_j, we consider update U(s_i, s_j) is feasible. Figure 1 shows a simple SUG example, in which a solid circle stands for an attribute with sensitive value.

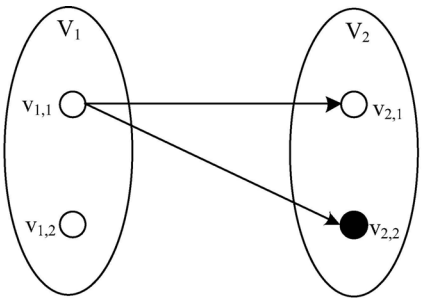

Figure 1. A Simple SUG

However, it is easy to find that some nodes and edges can be directly deleted from the graph, such as a node without edges on it. Thus we can get the feasible sub-SUG, which contains only feasible sensitive attributes, by deleting invalid nodes and edges from SUG. Figure 3 is the feasible sub-SUG deduced from Figure 2 by removing $V_{2,1}$, $V_{2,2}$, $V_{3,1}$ and $U(V_{2,2}, V_{3,1})$, $U(V_{2,2}, V_{3,2})$.

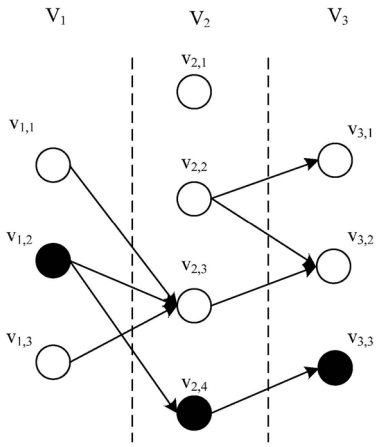

Figure 2. A Sample of Initial SUG

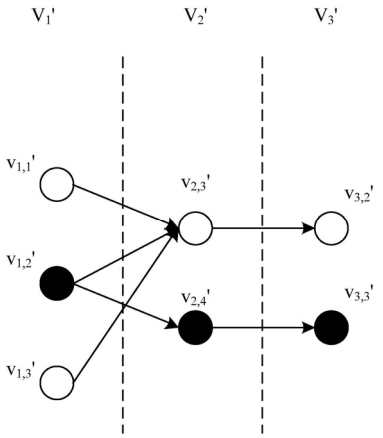

Figure 3. The Feasible Sub-SUG of Initial SUG

If there are always several indistinguishable feasible updates and the amount of sensitive attributes $|V_i| > 1$ in every feasible sub-SUG at any time i, disclosure of sensitive information will not happen. What's more, for all the records, it has to guarantee no deletion on all the records at any time, as to prevent the chain-actions of disclosure.

Therefore, the privacy and sensitive information are well protected during re-publication if it can meet the following two conditions:

All sensitive elements of every record are well preserved in each separate publication.

There is no deleting to all the record's feasible sub-SUG so as to maintain the in-distinguishability of sensitive elements all the time.

Considering the characteristics of set-valued data, transactional *k-anonymity* is used to ensure sensitive information not be revealed in each separate publication in this paper.

Even though all the records are well protected by

generalization, due to the characteristics of set-valued data, un-generalized elements still face the risk of being exposed in re-publication. Adversary can make use of the previous publications to deduce association between records and individuals and then expose the privacy. Therefore, by using transactional *k-anonymity* continuously, the un-generalized elements of a record are regarded as a complex-sensitive-attribute, and a feasible sub-SUG is created for records that contains complex-sensitive-attributes. If there is always more than one node and update path on the nodes in feasible sub-SUG, these elements will not be revealed.

3.3 Re-publication Principle and Method

Based on SUG theory for set-valued data re-publication, an anonymous k-preserving principle for dynamic set-valued dataset is proposed by modifying and extending m-Distinct. Update scenario and some other definitions is as follows.

Definition 1 Candidate Update Set Suppose e is an un-generalized element in an equivalence class, its Candidate Update Set $CUS(e)$ is the union of elements that e has non-zero probability to be changed to.

If $e' \in CUS(e)$, then $CUS(e') \subseteq CUS(e)$ must hold.

Definition 2 Update Combination Set Suppose every record of equivalence class g contains n un-generalized elements and their values are $\{e_1, e_2, \ldots, e_n\}$ respectively, then g's Update Combination Set $UCS(g)$ is a multi-set $\{CUS(e_1), CUS(e_2), \ldots, CUS(e_n)\}$. If an equivalence class group G contains m equivalence classes, then G's Update Combination Set $UCS(G)$ is $\{UCS(g_1), UCS(g_2), \ldots, UCS(g_m)\}$.

Since UCS is a multi-set of *CUS* and several elements may have the same CUS, thus a *CUS* may appear several times in a UCS. For any record, its *UCS* is inherited from the equivalence class it is in. During re-publication of dynamic set-valued dataset, the equivalence class and the *UCS* will update accordingly to the change of the elements.

Definition 3 Intersecting of UCS For any different times i, j ($i \neq j$), UCS_i and UCS_j are intersecting if they have equal number of *CUS* and there exists a one-to-one mapping between their *CUS*, such that the intersection of UCS_i and UCS_j is not empty.

Moreover, if the *CUS* of UCS_j is a subset of the *CUS* of UCS_i, we call UCS_i implies UCS_j.

Definition 4 Rational Update Result A set of un-generalized elements $EG = \{E_1, E_2, \ldots, E_n\}$ is a Rational Update Result if the following conditions hold:

(1) The number of complex-sensitive-attributes in *EG* is equal to the number of *CUS* in the *UCS*.

(2) For any element E_i in *EG*, there is at least one CUS_j such that $E_i \in CUS_j$.

(3) For any CUS_j, there is at least one complex-sensitive-attribute E_i such that $E_i \in CUS_j$.

If the un-generalized elements are a Rational Update Result of the *UCS* during re-publication, every node in the feasible sub-SUG cannot be excluded by the adversary, thus any sensitive attribute will not be revealed.

Definition 5 *k-preserving* Principle D is a dynamic set-valued dataset, a sequential publications $\{D_1^*, D_2^*, \ldots, D_n^*\}$ are *k-preserving* if they meet the following conditions:

(1) For any time $i \in [1,n]$, D_i^* meets *transactional k-anonymity*.

(2) For any record r and any time i, j ($i<j$), D_i and D_j are two consecutive publications that both contain r. For all $i \in [1,n]$, r_j's Candidate Attribute Set is a Rational Update Result of $UCS(r_i)$.

Soundness: The rationale of *k-preserving* is adopting *transactional k-anonymity* to maintain the in-distinguishability of records in each separate publication. When releasing new publication the records are divided carefully so that the in-distinguishability will be maintained. Rational Update Result is to guarantee that the sensitive information will not be revealed by the adversary using the historical publications.

Deriving from Definition 5, at any time i, if the new version of D_i^* meets *transactional k-anonymity* and for any record r in the new D_i^*, suppose r_{i-1} is r's most recent version, then r_{i-1}'s Candidate Attribute Set is a Rational Update Result of $UCS(r_{i-1})$, and the sequential publishing results including D_i^* are well protected.

Elements introduced into feasible sub-SUG are decided by the result of the static anonymization phase. After the static anonymization phase, all records in the same equivalence class must be identical, and some elements in the record may be generalized, others may be un-generalized. In order to prevent the un-generalized elements from being revealed, we choose the equivalence classes with the same generalized elements as a group and introduce the un-generalized elements as complex-sensitive-attributes into SUG, then obtain the feasible sub-SUG by pruning the SUG.

Completeness: Due to the *transactional k-anonymity* has a broader applicability, it has no restraints on the diversity of un-generalized elements in anonymous equivalence classes. In order to solve the *k-preserving* dynamic set-valued data publishing issue, counterfeit records will be imported to force each equivalence class that must has the same generalized values with at least 1 other equivalence class, and its complex-sensitive-attribute must can be updated to the one in the other equivalence classes.

In this way, Theorem 1 about *k-preserving* is described as follows.

Theorem 1 If sequential publications of set-valued dataset are *k-preserving*, for any record $r \in D$, $|V_i'| > 1$ holds for all elements in feasible sub-SUG.

Proof of Theorem 1 By description above, if no other equivalence classes contains identical generalized elements and corresponding complex-sensitive-attribute with a given equivalences class in the

anonymous result of static *k-anonymity*, counterfeit records will be used to guarantee that size each equivalence class group is greater than 1, thus the number of complex-sensitive-attribute in a group must be greater than 1, and the number of nodes in the feasible sub-SUG of records in the equivalence class $|V_i'|$ must be greater than 1.

The Candidate Attribute Set of r_{i+1} must be a Rational Update Result of the most recent version of r, which also means that any node in V_i has at least one outgoing edge connecting to a node in V_{i+1}, as well as any node in V_{i+1} has at least one incoming edge from a node in V_i. Since this holds, there will be no deletion happen in deduction of feasible sub-SUG, hence the theorem is proved.

Theorem 1 indicates that disclosure will not occur if the sequential publications are *k-preserving*. More equivalence classes means it is more difficult for the adversary to expose sensitive information due to that there will be more indistinguishable update path on nodes in feasible sub-SUG, which will reduces threat of privacy exposing. A larger k also means a larger number of records in each equivalence class, which will make internal update more complex and result in greater difficulties for privacy exposing.

4 Anonymous Re-publication Algorithms

4.1 Re-publication Algorithm

The crucial part of the re-publication algorithm is that, every record's new Candidate Attribute Set should be a Rationale Update Result of its previous Update Combination Set when publishing the current version. To achieve this goal, the most important is that records should be allocated to the corresponding equivalence classes effectively and rationally. Our algorithm use queue of bucket to store the original records and assign them to proper bucket according to their Update Combination Set, such that we can always find a way to assign records to an equivalence class, and the Candidate Attribute Set of each equivalence class is a Rationale Update Result of the record. The general algorithm is described in Algorithm 1.

The algorithm will first (1~21) create buckets that the records are possibly in, it makes all the possible buckets for the records of Recordsn that have ever appeared before. For the newly inserted records, new buckets for them will be created later if no suitable buckets exist. Besides, an entry means the only identifier of a bucket by its Update Combination Set, and the number of entries is equal to the number of Candidate Update Set of the bucket. In the second phase (22~34) of the algorithm, the main task is to assign records to their proper buckets. Counterfeit records will be used to balance the buckets so that sufficient number of records will be in every entry of every bucket. The requirement of assigning a record r to a bucket B is that, *UCS(rpre)* implies *UCS(B)* and *r[e]* is covered by *UCS(B)*. The last phase (35) of the algorithm is result publishing. Records in every entry of every bucket in the second phase will be

Algorithm 1. Dynamic Set-valued Data Re-publication *DSR*

Input: Recordsn: the n^{th} version of Dataset D.
D_{n-1}^*: the most recent publication of D, namely the previous equivalence classes.
k: the user configured parameter for *transactional k-anonymity*.
Output: anonymous result and counterfeit statistics.

1. create empty queues of buckets Q_{buc} and Q_{tmp};
2. **for all** record r in $Records_n$ **do** // r_{n-1}^* is the previous anonymous result of r in D_{n-1}^*
3. **if** $r \in D_{n-1}^*$ **then**
4. $B_{tmp} = createBucket(UCS(r_{n-1}^*))$; // $UCS(B_{tmp}) = UCS(r_{n-1}^*)$
5. **if** $B_{tmp} \notin Q_{buc}$ **then**
6. $Q_{buc} \leftarrow B_{tmp}$;
7. **end if**
8. **else**
9. $Q_{buc} \leftarrow$ create new bucket for r;
10. **end if**
11. **end for**
12. **for** any 2 buckets $B_i, B_j \in Q_{buc}$ $(i<j)$ **do**
13. **if** $UCS(B_i)$ and $UCS(B_j)$ are intersecting **then**
14. choose UCS_{tmp} with highest proportion of the overlapped elements of the intersection plan;
15. $B_{tmp} = createBucket(UCS_{tmp})$; // $UCS(B_{tmp}) = UCS_{tmp}$
16. **if** $B_{tmp} \notin Q_{buc}$ and $B_{tmp} \notin Q_{tmp}$ **then**
17. $Q_{tmp} \leftarrow B_{tmp}$;
18. **end if**
19. **end if**
20. **end for**
21. $Q_{buc} \leftarrow Q_{tmp}$;
22. **for all** r in $Records_n$ **do**
23. compute $COUNT[r]$ for possible distributions of r;
24. **end for**
25. $sort(Records_n, COUNT)$; // Non-decreasing order of $COUNT$
26. **for all** r in $Records_n$ **do**
27. **if** $COUNT[r] > 0$ **then**
28. $Q_r = getBuckets(r)$;
29. $B_r, Entry_r \leftarrow getDistribution(r, Q_r)$;
30. $assign(r, B_r, Entry_r)$;
31. **end if**
32. **end for**
33. $EPPSAnonymize(Q_{rec})$; // use static anonymous algorithm for records that first appear in D
34. use counterfeit records to maintain all entries of every bucket with suitable number of records
35. publish the anonymous result and counterfeit statistics

transformed into equivalence classes. At last, we publish them together with the corresponding counterfeit statistics. Note that counterfeit record is introduced only if records in the original dataset are not enough to prevent the un-generalized element from being revealed, it helps the re-publication meet *k-preserving* in the algorithm.

Except for meeting the re-publication principle, there is also one goal of minimizing information loss of the anonymous result. Because that, common used anonymous algorithm for *transactional k-anonymity* is local generalization, the original data may be generalized into a general and abstract value to meet the privacy protection principle, thus the anonymous result will lose some information and will not be conducive to practical applications.

4.2 Improved Anonymous Algorithm

As we mentioned earlier, the Partition algorithm of transactional *k-anonymity* has the limitation of higher generalization degree, which increases the information loss of the anonymous result. Without using *GH*, [13] provided a suppression algorithm. They select records to constitute the optimal equivalence classes by Minimum Operation, and obtain anonymous result by least data flipping operations and time cost. In this paper, we improved the set-valued data anonymous algorithm by integrating local generalization with suppression technique. We regard the sub-partitions as the objects to be suppressed. Also note that, if elements should be inserted into sub-partitions for the anonymization, we just simply add a random element of the generalization node into each record of the sub-partition. In contrast, we should remove all the leaf elements of the flipped generalization node out of the sub-partition. Suppose N is the set of generalization nodes of the sub-partitions group G, Minimum Operation means the minimum insertion and deletion is needed to achieve *k-anonymity*, which can be also defined as follows:

$$MO(G) = \sum_{n \in N} O_G(n) \quad \textbf{(1)}$$

Where N stands for all of the generalization nodes of G, $OG(n)$ stands for the operations to anonymous node n, $n \in N$. If the anonymous result remains n, the value of $OG(n)$ is the number of records that do not contain n, otherwise the value is the number of all the leaf elements of n in G. The final result of $OG(n)$ is valued by the smaller one of the above two cases.

At the first phase of the EPPS algorithm, we still adopt the top-down partitioning technique in distributing records of current partition into the sub-partitions created by node splitting, so that every sub-partition will meet transactional *k-anonymity*. For the sub-partitions dissatisfy transactional *k-anonymity*, we first merge them into a remaining partition with high level generalization. Consider that the remaining partition may not be the anonymous result that have the least data distortion, thus in the second phase, we adopt suppression technique with the least flipping operations on the remaining partition. We try to specialize each of the generalization nodes to get result with less information loss at first, otherwise, we directly suppress these sub-partitions. By computing Minimum Operation on the sub-partitions, we can form optimal suppression groups, and then we also try to specialize the generalization value of each group. Finally, we choose the anonymous result with less practical data distortion as the equivalence classes and insert them into the publishing results. Thus, the information loss of each static anonymization can be reduced. The general EPPS algorithm is described in Algorithm 2.

Algorithm 2. Recursive Partition with Suppression Algorithm on Set-valued Data

EPPSAnonymize
Input: Original Partition partition
Output: Global Result *equivalenceClasses*
1. **if** impossible to further split for *partition* **then**
2. return and put *partition* into *equivalenceClasses*
3. **else**
4. *splitNode*, *resultPartitions* ← *pickNode*(*partition*);
5. **for all** *subPartition* in *resultPartitions* **do**
6. *isSatisfy* = *verifyPartition*(*subPartition*);
7. **if** !*isSatisfy* **then** // if *subPartition* does not satisfy *k-anonymity*
8. *remainingPartitions* ← *subPartition;*
9. **end if**
10. **end for**
11. **if** *remainingPartitions* satisfy *k-anonymity* **then**
12. *Suppress*(*remainingPartitions*);
13. **end if**
14. **for all** *subPartition* in *resultPartitions* **do**
15. *EPPSAnonymize*(*subPartition*);
16. **end for**
17. **end if**

In Algorithm 2, we use Practical Data Distortion (*PDD*) instead of Normalized Certainty Penalty [3] to choose splitNode and evaluate the anonymous dataset. PDD will evaluate the information loss by comparing the anonymous result with the original records directly, thus it can adapt to the difference of evaluating between local generalization algorithm and our integrating local generalization with suppression algorithm. Suppose I is the set of all elements, i is an original element or its generalization value of equivalence class EC, the PDD is defined as follows:

$$\mathrm{PDD}(i)=\begin{cases}0, & |n_i|=1;\\ 1, & \nexists n_i;\\ |n_i|/|I|, & otherwise.\end{cases} \quad (2)$$

Where n_i is the generalization node of i, and $|n_i|$ is the total number of leaf nodes under it. Intuitively, the first equation states that when the element i is not generalized, there is no distortion. The second equation states that if the element is suppressed, its distortion value is 1 because the corresponding node has been removed away during the suppression procedure. The last equation states that the distortion for a generalized element is the ratio of leaves it covers to total number of leaves in the *GH*.

Suppose D is the original set-valued dataset, D' is the anonymous result of D, r is a record in D', i is an element in r, and Cr is the count of elements of r, then the *PDD* of D' is defined as follows:

$$\mathrm{PDD}(D')=\frac{\sum_{r\in D'}\sum_{i\in r}\mathrm{PDD}(i)}{\sum_{r\in D}C_r} \quad (3)$$

In other words, the overall distortion of the anonymous result is the weighted average of the distortion of all the elements, with a possible range from 0 to 1.

4.3 Privacy Preserving Re-publication Model

The privacy preserving re-publication model for set-valued datais shown in Figure 4. The *k*-preserving principle provides a theoretical basis for anonymous processing method. Algorithm DSR and EPPS can well protect sensitive information against the adversary's attacks and retain information integrity. PDD is to evaluate the mid-results in EPPS and verify the availability of the model.

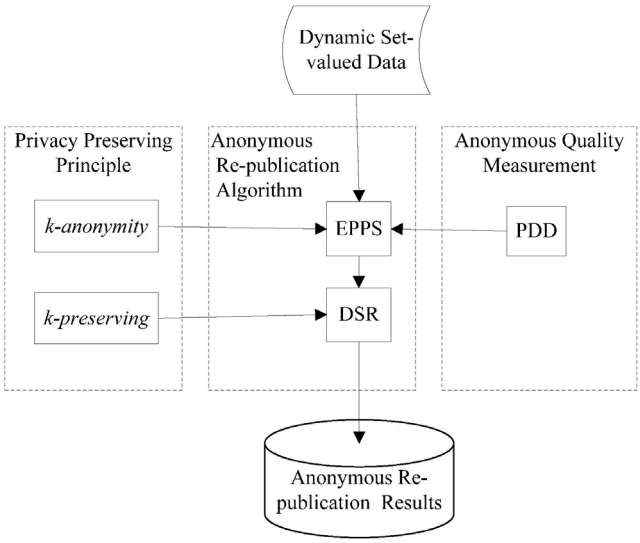

Figure 4. Privacy Preserving Re-publication Model for Set-Valued Data

5 Experiment and Evaluation

We evaluate and verify our approach by experiments in this section. We implement all the algorithms by Java. Experiments are performed on a PC with Intel Xeon E5410 and 3GB available RAM.

As to the market-basket data, there are several public and real datasets, such as BMS-POS, BMS-WebView-1 and BMS-WebView-2, etc. BMS-POS is transaction logs of an electronics retailer's sales for several years, while BMS-WebView-1 and BMS-WebView-2 are clickstream data for several months from two e-commerce web sites. These datasets are widely used as benchmark datasets in knowledge discovery community. Some information about the datasets is listed in Table 1.

Table 1. Characteristics of the Three Benchmark Datasets

Dataset	Distinct Elements	Max Record Size	Avg Record Size
BMS-POS	515,597	1,657	6.5
BMS-WebView-1	59,602	497	2.5
BMS-WebView-2	77,512	3,340	5.0

5.1 Experiment for EPPS

We use BMS-POS which contains more than 500,000 records as the original experimental data, because that in practical application, the quantity of data is always large, and the average record size of BMS-POS is greater so that it is easier to simulate inner update procedure. Our anonymous algorithm uses Generalization Hierarchy on experimental dataset, due to these datasets do not include a given GH, we followed the example from [2] and artificially constructed a corresponding one on the union of all items appearing in BMS-POS. The node fan-out f of GH specifies how many items are generalized from one level to its parent level. We set a uniform node fan-out as 5 as [3] did. Besides, we set parameter k in specific experiment with different settings.

The simulation of external update is that, 200,000 records of the original dataset are used for the first publishing, then in each re-publication, 5,000 randomly-selected records will be removed and 10,000 records from the rest of records will be inserted into the dataset. The dataset will be re-published 20 times and all the records used are randomly selected. The simulation of internal update is that, for all the elements of the dataset are specified by integer using *GH*, we set value of the elements randomly change within its domain on the GH. The number of each internal update is limited to no more than 5% of the number of total elements of the current dataset.

During the anonymizing process some elements may be added to or removed from the raw data, so it is necessary to measure the quality of anonymous results.

We experiment on BMS-WebView-2, using Kullback-Leibler divergence [20] as measurement with different parameter *k*. The result in Figure 5 shows that with the increase of *k*, the *KL* divergence of the anonymous results increases but keeps a low level. When *k* is more than 10, *KL* divergence is stable at about 0.03, therefore, we choose the values of *k* in {2, 5, 10}.

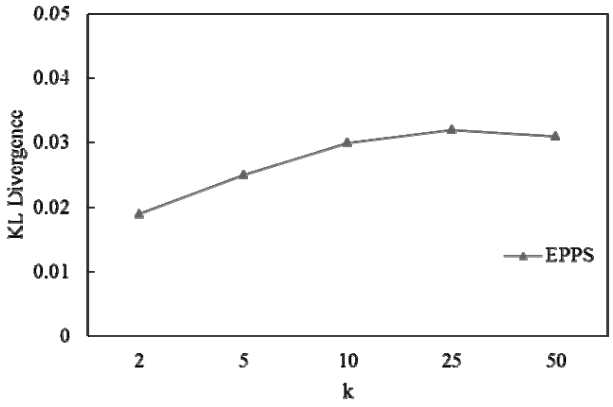

Figure 5. KL Divergence of Anonymous Results

In the first experiment, we use our improved static anonymous algorithm EPPS to do dynamic set-valued dataset re-publication experiment. Figure 6 shows the relationship between publishing times and proportion of revealed elements in the case of different parameter *k*. Since generalization degree of the anonymous result grows with increase of the parameter *k*, the number of un-generalized elements will decrease and cannot directly reflect the real result of privacy disclosure, so that we use revealed elements in the proportion of all un-generalized elements as evaluation of privacy disclosure.

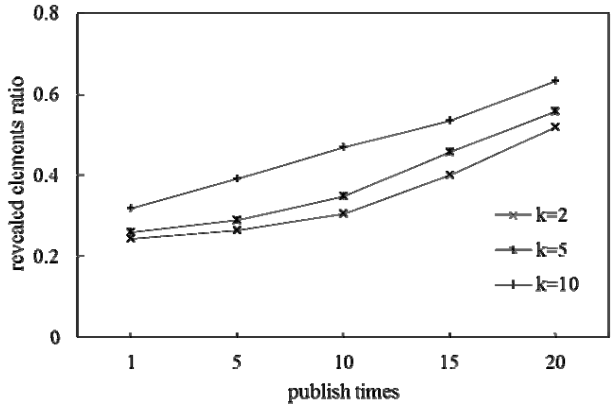

Figure 6. Revealed Elements Ratio with Publishing

The result shows that static anonymous algorithm is insufficient to re-publication of dynamic dataset, and the number of revealed elements increases as the process evolves. With the increases of *k*, diversity of the anonymous results becomes smaller and sensitive information become easier to be revealed.

5.2 Experiment on Re-publication

In the next experiment, Figure 7 shows the comparison on running time and Practical Data Distortion of employing Partition [3], EPPS and DSR on re-publication. Parameter *k* is fixed to 10 in this experiment. We can see that, EPPS spends about 15% more time than Partition on average, this time cost is spent by suppression phase. DSR spends even more time on implementing privacy preserving of updated dataset. Though EPPS and DSR need more time cost, Figure 8 shows that EPPS reduces about 18% information loss compared with Partition. Though the information loss of *DSR* is more than static anonymous algorithm due to the introduced counterfeit records, it tends to be stable because that the number of counterfeit records also gradually stabilize as the re-publication process evolves.

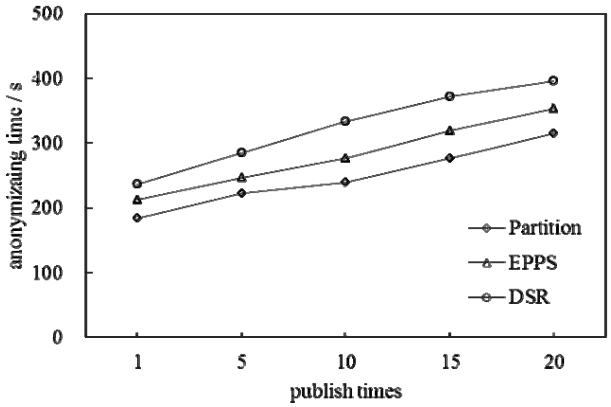

Figure 7. Anonymizing Time Comparison

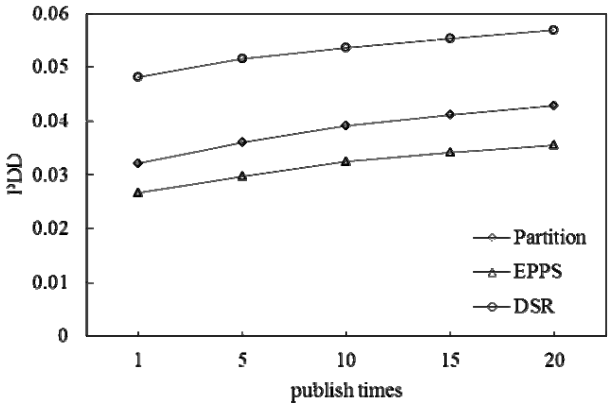

Figure 8. Information Loss Comparison

In our approach, counterfeit record is used to maintain the diversity of sensitive elements in the anonymous results and prevent privacy disclosure. Therefore, we did the next experiment to count the counterfeit records used in re-publication with different parameter *k*. Due to the same reason with the first experiment, we use the number of counterfeit records in proportion of the whole dataset as evaluation.

Figure 9 shows that more counterfeit records are

introduced with increasing of publishing times and parameter k, this is because greater k leads to more limitation for diversity of sensitive attributes in the anonymous results, so that we need more counterfeit records to prevent the un-generalized element from being revealed. However, we can also see that, the proportion of counterfeit records becomes stable as re-publication evolves, and the Practical Data Distortion of the sequential publications has been controlled with 6%, in other words, our approach effectively limits the information loss. Therefore, although the method proposed in this paper uses counterfeit records, on the basis of remaining better data integrity and availability, it still well protects sensitive information of publishing results from being malicious exposed.

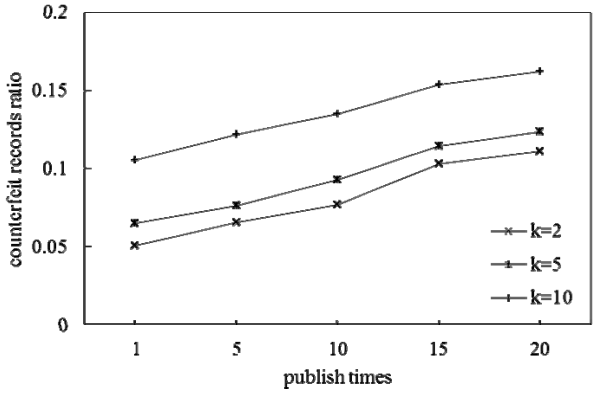

Figure 9. Counterfeit Record Statistics of DSR

6 Conclusion

In this paper, we studied privacy preserving on re-publication of dynamic set-valued data problem based on existing dynamic traditional relational data re-publishing research, proposed an anonymous protecting model and re-publication algorithm to guarantee sensitive information of set-valued dataset with both internal and external update not be revealed during re-publishing scenario effectively, and the sequential anonymous publications all meets *k-preserving*. In order to reduce information loss of anonymous result, we integrated local recoding generalization and suppression on set-valued data anonymization. The approach in this paper still needs to be improved. We use transactional *k-anonymity* to protect set-valued dataset under eliminating the distinction between sensitive and non-sensitive attributes, thus more counterfeit records must be used to guarantee in-distinguishability between the anonymous records and diversity of un-generalized elements relatively. Consider that distinction of sensitive attribute in set-valued data may exist in certain application, novel privacy preserving principle and method can be further studied for such applications in future works [18-19], so that more rigorous privacy expose risk limit can be provided with less or without counterfeit records.

Acknowledgments

This paper is supported by Beijing Natural Science Foundation P.R.China (4173072).

References

[1] S. Zhou, F. Li, Y. Tao, X. Xiao, Privacy Preservation in Database Applications: A Survey, *Chinese Journal of Computers*, Vol. 32, No. 5, pp. 847-861, May, 2009.

[2] M. Terrovitis, N. Mamoulis, P. Kalnis, Privacy-Preserving Anonymization of Set-Valued Data, *Proceedings of the VLDB Endowment*, Vol. 1, No. 1, pp. 115-125, August, 2008.

[3] Y. He, J. -F. Naughton, Anonymization of Set-Valued Data via Top-Down, *Proceedings of the VLDB Endowment*, Vol. 2, No. 1, pp. 934-945, August, 2009.

[4] M. Atzori, F. Bonchi, F. Giannotti, D. Pedreschi, Anonymity Preserving Pattern Discovery, *The VLDB Journal*, Vol. 17, No. 4, pp. 703-727, July, 2008.

[5] P. Samarati, L. Sweeney, Protecting Privacy When Disclosing Information: K-Anonymity and its Enforcement through Generalization and Suppression, *Proceedings of the Symposium on Security and Privacy*, Washington, DC, 1998, pp.1-19.

[6] X. Yang, Y. Wang, B. Wang, G. Yu, Privacy Preserving Approaches for Multiple Sensitive Attributes in Data Publishing, *Chinese Journal of Computers*, Vol. 31, No. 4, pp. 574-587, April, 2008.

[7] B. Fung, K. Wang, R. Chen, P. -S. Yu, Privacy-Preserving Data Publishing: A Survey of Recent Developments, *ACM Computing Surveys (CSUR)*, Vol. 42, No. 4, pp. 14-20, April, 2010.

[8] H. Park, K. Shim, Approximate Algorithms for K-Anonymity, *Proceedings of the 2007 ACM SIGMOD*, New York, NY, 2007, pp.67-78.

[9] F. Li, S. Zhou, Challenging More Updates: Towards Anonymous Re-Publication of Fully Dynamic Datasets, arXiv preprint arXiv:0806.4703, http://arxiv.org/ abs/0806.4703.

[10] M. Terrovitis, N. Mamoulis, P. Kalnis, Local and Global Recoding Methods for Anonymizing Set-Valued Data, *The VLDB Journal*, Vol. 20, No. 1, pp. 83-106, January, 2011.

[11] L. Sweeney, K-Anonymity: A Model for Protecting Privacy, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 10, No. 5, pp. 557-570, May, 2002.

[12] R. Motwani, S. -U. Nabar, Anonymizing Unstructured Data, arXiv preprint arXiv:0810.5582, http://arxiv.org/abs/0810.5582.

[13] S. Wang, Y. Tsai, H. Kao, T. Hong, Extending Suppression for Anonymization on Set-Valued Data, *International Journal of Innovative Computing, Information and Control*, Vol. 7, No. 12, pp. 6849-6843, December, 2011.

[14] J. Liu, K. Wang, Anonymizing Transaction Data by Integrating Suppression and Generalization, *Proceedings of*

the Advances in Knowledge Discovery and Data Mining, Berlin Heidelberg, Germany, 2010, pp.171-180.

[15] K. Wang, B. Fung, Anonymizing Sequential Releases, *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, 2006, pp. 414-423.

[16] B. Fung, K. Wang, A. -W. Fu, J. Pei, Anonymity for Continuous Data Publishing, *Proceedings of the 11th International Conference on Extending Database Technology: Advances in Database Technology*, New York, NY, 2008, pp.264-275.

[17] X. Xiao, Y. Tao, M-Invariance: Towards Privacy Preserving Re-publication of Dynamic Datasets, *Proceedings of the 2007 ACM SIGMOD*, New York, NY, 2007, pp. 689-700.

[18] A.-E. Cicek, M.-E. Nergiz, Y. Saygin, Ensuring Location Diversity in Privacy-preserving Spatio-temporal Data Publishing, *The VLDB Journal*, Vol. 23, No. 4, pp. 609-625, April, 2014.

[19] P. Belsis, and . Pantziou, A k-anonymity Privacy-preserving Approach in Wireless Medical Monitoring Environments, *Personal and Ubiquitous Computing*, Vol. 18, No. 1, pp. 61-74, April, 2014.

[20] D. Kifer, J. Gehrke, Injecting Utility Into Anonymized Datasets, *Proceeding of the 2006 ACM SIGMOD International Conference on Management of Data*, New York, NY, 2006, pp. 217-228.

Biographies

Dan Wang received the Ph.D. degree in computer software from Northeastern University in 2002. She is currently a professor at Beijing University of Technology. Her research interests include software verification, trustworthy software privacy protection and distributed computing.

Yi Wu was born in 1987, he received his master degree in computer science and technology from Beijing University of technology in 2013. His current research interests include database and data privacy.

Wenbing Zhao received the Ph.D. degree in Signal and Information Process from Peking University in 2004. She is an assistant professor at Beijing University of Technology. Her research interests include data mining, trustworthy software.

Lihua Fu received the Ph.D. degree in computer software from Northwestern Polytechnical University in 2005. She is currently an associate professor at Beijing University of Technology. Her research interests include computer vision, image processing, and image understanding.