Privacy Preserving Publishing on Multiple Quasi-Identifiers

Jian Pei [†], Yufei Tao [‡], Jiexing Li [‡], Xiaokui Xiao [‡]

[†]*Simon Fraser University, Canada* [‡]*The Chinese University of Hong Kong, China*

[†]jpei@cs.sfu.ca [‡]{taoyf, jxli, xkxiao}@cse.cuhk.edu.hk

Abstract— **In some applications of privacy preserving data publishing, a practical demand is to publish a data set on multiple quasi-identifiers for multiple users simultaneously, which poses several challenges. Can we generate one anonymized version of the data so that the privacy preservation requirement like k-anonymity is satisfied for all users and the information loss is reduced as much as possible? In this paper, we identify and tackle the novel problem by an elegant solution.**

The full paper [1] can be found at http://www.cs.sfu.ca/~jpei/publications/butterfly-tr.pdf.

I. Introduction

In some applications of privacy preserving data publishing, a practical demand is to publish a data set simultaneously on multiple quasi-identifiers for users carrying different background knowledge. For example, the traffic management board of a region collects records of road accidents for research and analysis. Suppose each record has five attributes, namely occupation, age, vehicle-type, postcode, and faulty. Consider the tuples in Table I.

Such records are interesting to auto insurance companies which can use such information to analyze the risk of their business and define their policies accordingly. Simultaneously, such traffic accident records are also interesting to the human resource department in the government since they can be used to analyze the impact of accidents on working groups. Therefore, the traffic management board may want to release the data to multiple users.

Importantly, different users may carry different background knowledge. For example, the auto insurance company may join the traffic accident records with the vehicle registration records on attributes age, vehicle-type, and postcode to find out whether its customers were at fault in some accidents. Typically, the company does not have the occupation information of its customers, as such information is not required in applying for auto insurance. Therefore, to protect privacy, the traffic management board has to anonymize the traffic accident records on attributes age, vehicle-type, and postcode before the data can be released to the auto insurance company. Suppose 2-anonymity is required. Table II shows a 2-anonymous release of the records with respect to quasi-identifier (age, vehicle-type, postcode).

TABLE I
A SET OF TRAFFIC ACCIDENT RECORDS.

Occupation	age	vehicle	postcode	faulty
Dentist	30	Red Truck	31043	No
Family doctor	30	White Sedan	31043	Yes
Banker	30	Green Sedan	31043	No
Mortgage broker	30	Black Truck	31043	No

TABLE II
A 2-ANONYMOUS RELEASE OF THE TRAFFIC ACCIDENT RECORDS IN TABLE I WITH RESPECT TO QUASI-IDENTIFIER (age, vehicle-type, AND postcode).

Occupation	age	vehicle	postcode	faulty
Dentist	30	Truck	31043	No
Family doctor	30	Sedan	31043	Yes
Banker	30	Sedan	31043	No
Mortgage broker	30	Truck	31043	No

TABLE III
A 2-ANONYMOUS RELEASE OF THE TRAFFIC ACCIDENT RECORDS IN TABLE I WITH RESPECT TO QUASI-IDENTIFIER (occupation, age, AND postcode).

Occupation	age	vehicle	postcode	faulty
Medical	30	Red Truck	31043	No
Medical	30	White Sedan	31043	Yes
Finance	30	Green Sedan	31043	No
Finance	30	Black Truck	31043	No

Simultaneously, the human resource department may join the traffic accident records with the resident records on attributes occupation, age, and postcode to find out which residents were faulty in some accidents. Therefore, to protect privacy, the traffic management board needs to anonymize the traffic accident records on attributes occupation, age, and postcode. Note that vehicle-type is not part of the anonymization, because the human resource department typically does not have information about residents' vehicle types. Again, suppose 2-anonymity is required. Table III shows a 2-anonymous release of the records with respect to quasi-identifier (occupation, age, postcode).

The traffic management board needs to protect the privacy against attacks using different background knowledge. Releasing a data set to multiple users leads to serious concerns on privacy preservation. Even though we ensure that the release to each user satisfies the corresponding privacy-preservation requirement such as k-anonymity, privacy still can be disclosed if collusion happens.

Suppose an adversary obtains both releases in Tables II and III. By comparing the two tables, the adversary immediately knows that a family doctor of age 30 driving a white Sedan living in area 31043 was faulty in an accident. The vic-

1084-4627/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDE.2009.183

tim may be easily re-identified by both the vehicle registration record and the human resource resident record. The loophole is serious since the attack can be made even without sharing any background knowledge (i.e., vehicle registration records and resident records) from the two users. Instead, any adversary obtaining both releases can intrude the privacy.

In this paper, we tackle the problem and make the following contributions. First, we identify the novel problem of privacy preserving publishing on multiple quasi-identifiers. Second, we indicate that it is possible to generate only one anonymized table to satisfy the k-anonymity on all quasi-identifiers for all users without significant information loss. Our method is substantially better than the naïve method which conducts anonymization using the union quasi-identifier. Last, we systematically develop an effective method to generate such an anonymized table for multiple users.

II. Problem Definition

Consider a micro-data table $T = (A_1, \ldots, A_n)$, where a record in the table represents the data for one individual. An *external table* $E = (B_1, \ldots, B_m)$ also containing records of individuals is used to model the background knowledge of a user. A *re-identification attack* to the privacy of individuals in table T is that the user can join tables T and E on the common attributes of the two tables so that individuals in T may be re-identified. The set of common attributes between tables T and E, i.e., $S = T \cap E$, is called the *quasi-identifier* (*QID* for short) with respect to the re-identification attack using E.

To protect privacy against re-identification attacks, the owner of T may change the values of tuples in T on attributes in QID S so that at least k tuples look the same on QID S. Then, each individual cannot be re-identified with a probability over $\frac{1}{k}$. Technically, an *anonymization* is a function f on T such that for each tuple $t \in T$, $f(t)$ is a tuple where some values of t may be changed.

Suppose a table T is anonymized as T', i.e., $T' = \{f(t) | t \in T\}$. For a tuple $t \in T'$, the set of tuples $t' \in T'$ which have the same values as t on all attributes in S form an *equivalence class* (*EC* for short) on S, i.e., $E(t) = \{t' \in T' | \forall A \in S, t'[A] = t[A]\}$. Clearly, $t \in E(t)$. T' is *k-anonymous* ($k > 0$) on QID S if for each tuple $t \in T'$, $\|E(t)\| \geq k$.

A general representation of anonymized tuples [2], [3] is to generalize an attribute value to a range. For example, if we want to make k tuples into an EC and the values of those tuples on attribute age range from 20 to 30, we can generalize the values to a range $[20, 30]$. Apparently, the larger the range, the more information loss is introduced by the anonymization.

Some methods have been developed to measure the information loss in anonymization. In this paper, we adopt the uncertainty penalty measure of information loss which is also used in [2], [3].

Definition 1 (Uncertainty penalty): Suppose table T is anonymized to T'. In the domain of each attribute in T, suppose there exists a global order on all possible values in the domain. If a tuple t in T' has range $[x, y]$ on attribute A, then the **uncertainty penalty** in t on A is $loss_A(t) = \frac{\|y-x\|}{\|A\|}$, where $\|A\| = \max_{t' \in T}\{t'[A]\} - min_{t' \in T}\{t'[A]\}$ is the range of attribute A in T. For tuple t, the **uncertainty penalty** in t is $loss(t) = \sum_{A \in S} loss_A(t)$, where S is the QID.

The **uncertainty penalty** in T' is $\sum_{t \in T'} loss(t)$. ■

TABLE IV

An example showing the rationale of Theorem 1.

A	B	C
a_1	b_1	c_1
a_1	b_1	c_2
a_1	b_2	c_1
a_1	b_2	c_2
a_2	b_1	c_1
a_2	b_1	c_2
a_2	b_2	c_1
a_2	b_2	c_2

In this paper, we consider the situation where a micro-data table $T = (A_1, \ldots, A_n)$ needs to be anonymized and released for a group of users $U_1, \ldots, U_m$. For each user U_i $(1 \leq i \leq m)$, we assume a quasi-identifier $S_i \subseteq T$ that models U_i's background knowledge to attack the privacy of individuals in T. Thus, we need to make sure that the release for U_i is k-anonymous with respect to S_i.

We are interested in generating only one anonymized version T' such that T' is k-anonymous with respect to all S_i $(1 \leq i \leq m)$. The problem of *privacy preserving publishing for multiple users* is to generate the k-anonymous table T' so that the k-anonymity requirement for each user is satisfied, and the information loss is as small as possible.

A naïve approach is to generate a table T' such that T' is k-anonymous with respect to the *union QID* $S = \cup_{i=1}^{m} S_i$. Apparently, if T' is k-anonymous with respect to S, T' is also k-anonymous with respect to any individual S_i. We call this method the *union QID method*.

The union QID may contain many more attributes than any individual QID, and thus the union QID method may introduce substantial information loss. Interestingly, we can show that the union QID may not be necessary to ensure k-anonymity with respect to all individual QIDs.

Theorem 1 (Union QID): Let $\mathcal{R} = (A_1, \ldots, A_n)$ be a schema of micro-data where the domain of each attribute has the cardinality of at least 2. Let $S_1, \ldots, S_m$ be m QIDs and $S = \cup_{i=1}^{m} S_i$ be the union QID. If there does not exist S_{i_0} $(1 \leq i_0 \leq m)$ such that $S_{i_0} = S$, then there exists a table R on $\mathcal{R}$ such that R is k-anonymous with respect to every S_i $(1 \leq i \leq m)$ but R is not k-anonymous with respect to S.

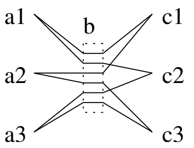
Rationale. Limited by space, we omit the formal proof. Instead, we give an example to illustrate the idea. Table IV shows a table constructed as such which is 2-anonymous with respect to any proper subset of ABC, but is not 2-anonymous with respect to ABC. ■

III. The Butterfly Method

Let us consider the basic case where there are 2 users, U_1 and U_2, using QIDs S_1 and S_2, respectively. We need to anonymize a table $T = (A_1, \ldots, A_n)$ to a table T' such that T' is k-anonymous with respect to QIDs S_1 and S_2. Let $S = S_1 \cup S_2$. Generally, we assume $S \neq S_1$ and $S \neq S_2$.

A	B	C
a_1	b	c_1
a_1	b	c_2
a_2	b	c_1
a_2	b	c_3
a_3	b	c_2
a_3	b	c_3

(a) A 2-anonymous table



(b) Visualization of a butterfly.

Fig. 1. A butterfly in a 2-anonymous table.

Example 1 (Butterfly): Table $T = (A, B, C)$ in Fig. 1(a) is 2-anonymous with respect to $S_1 = AB$ and $S_2 = BC$, but not 2-anonymous with respect to $S = ABC$.

On QIDs $S_1 = AB$ and $S_2 = BC$, respectively, the tuples form ECs such that each EC is of size 2. Interestingly, the tuples share the same values on B, the common attribute between S_1 and S_2. This sharing is critical to achieve tuples that do not need to form ECs on ABC but still can satisfy the k-anonymity requirements on S_1 and S_2.

In Fig. 1(b), a tuple is a line connecting the values on attributes A, B and C. It looks like a butterfly: the tuples sharing the same value on B which is the body of the butterfly. Different values on A and C form "wings" of the butterfly. Generally, a butterfly structure in our study may have multiple "wings", but a biological butterfly has only 4 wings. ■

Based on Example 1, we define butterfly, the essential structure in anonymizing tables for multiple QIDs.

Definition 2 (Butterfly): Given a table T, and two QIDs S_1 and S_2 on T such that $S_1 \cup S_2 \neq S_1$ and $S_1 \cup S_2 \neq S_2$. A set of tuples $P \subseteq T$ is a k-**butterfly** with respect to S_1 and S_2 if

1) P can be partitioned into ECs on $S_1 - S_2$ such that each EC is of size at least k;
2) P can be partitioned into ECs on $S_2 - S_1$ such that each EC is of size at least k;
3) All tuples in P have the same values on attributes in $S_1 \cap S_2$. ■

According to Definition 2, all tuples in Fig. 1(a) form a 2-butterfly. A k-butterfly has several interesting and desirable properties.

Proposition 1 (k-anonymity of butterfly): Let P be a k-butterfly with respect to QIDs S_1 and S_2. Then, P is k-anonymous with respect to S_1 and S_2. ■

Proposition 1 indicates that k-butterflies can be used to anonymize a table for two QIDs since the k-anonymity requirement on each QID can be satisfied.

Proposition 2 (EC and butterfly): In table T, an equivalence class of size k with respect to union QID $S_1 \cup S_2$ is a k-butterfly with respect to S_1 and S_2, where S_1 and S_2 are two QIDs on T. ■

Proposition 2 indicates that, in anonymization for multiple QIDs, ECs with respect to the union QID is a special case of butterflies. Importantly, a butterfly provides more flexibility that it does not require all values be the same on the union QID. The flexibility brings in the opportunity for reducing information loss in anonymizaiton, as will be explored by our anonymization algorithm.

Are butterflies sufficient to anonymize a table for multiple QIDs?

Theorem 2 (Butterfly): A table T is k-anonymous with respect to QIDs S_1 and S_2 if and only if the tuples in T can be partitioned into exclusive subsets $P_1, \ldots, P_l$ such that each P_i $(1 \leq i \leq l)$ is a k-butterfly with respect to S_1 and S_2. ■

According to Theorem 2, the problem of anonymizing a table for QIDs S_1 and S_2 can be reduced to transforming the tuples in T into a set of k-butterflies. We define the k-butterfly anonymization problem as follows.

Given a table T and QIDs S_1 and S_2, the problem of *k-anonymization using butterflies* is to transform T into table T' consisting of a set of k-butterflies with respect to S_1 and S_2, and the information loss from T to T' is minimized.

By a reduction from the k-anonymization problem which has been shown NP-hard [4], we have the following result.

Theorem 3 (Complexity): The problem of k-anonymization using butterflies is NP-hard. ■

Now, we develop a heuristic algorithm to anonymize a table using butterflies on two QIDs.

1) General Idea: An EC of at least k tuples on $S_1 \cup S_2$ is a special type of k-butterfly. In the naïve union QID method, we can anonymize table T using ECs of size at least k on $S_1 \cup S_2$. Alternatively, we can construct large butterflies which are not k-anonymous with respect to $S_1 \cup S_2$. An extreme is that we generalize all tuples to the same on attributes $S_1 \cap S_2$, and then we can conduct k-anonymization on $S_1 - S_2$ and $S_2 - S_1$ independently.

Essentially, there is a tradeoff between using small butterflies and using large butterflies in information loss on different attributes. The advantage of a large butterfly is that it allows less information loss on attributes in $S_1 - S_2$ and $S_2 - S_1$ since tuples do not need to take the same values on those attributes. The disadvantage is that, since all tuples in a butterfly have the same values on attributes in $S_1 \cap S_2$, a large butterfly may lead to heavy information loss on those attributes.

Therefore, to reduce the information loss using butterflies, we need to balance the gain on the attributes in $S_1 \cup S_2 - S_1 \cap S_2$ and the loss on the attributes in $S_1 \cap S_2$.

The general idea of our method to anonymize a table T is in two steps.

First, we anonymize T on the union QID $S_1 \cup S_2$, and form a binary hierarchy (i.e., a binary tree) of ECs. Each internal node in the hierarchy is a set of ECs.

Second, we examine the nodes in the hierarchy of ECs bottom-up to check whether reorganizing the tuples at a node into a butterfly may potentially reduce the information loss. If so, we apply a butterfly construction algorithm on the set of tuples. If the butterfly constructed as such reduces the information loss, it is used to anonymize the tuples.

2) Step 1: Building a Binary Hierarchy of ECs: First, we build a binary hierarchy of ECs on $S_1 \cup S_2$. This hierarchy is a natural product of some generalization algorithms such as *Mondrian* [5]. If the ECs are computed by other algorithms, the hierarchy can be easily constructed through a "binary clustering" of all the ECs. To illustrate, assume that the generalized form of each EC is a rectangle in the space formed by the attributes of $S_1 \cup S_2$. Then, we only need to create a binary R-tree on the ECs, which is already a good hierarchy.

3) Estimating Reduction of Information Loss: In a binary hierarchy of ECs, *the set of tuples at a node* N, denoted by $T(N)$, are the tuples in the ECs that are descendants of N. Now, our task is to try to organize the tuples in $T(N)$ into butterflies to reduce information loss.

For a node N in the binary hierarchy of ECs, in order to efficiently check whether reorganizing the tuples in $T(N)$ into a butterfly may reduce information loss, we derive a lower bound of the information loss in such a butterfly. The computation of the lower bound does not require to construct the butterfly. Thus, we can first check whether the lower bound indicates a potential reduction of information loss before we construct the butterfly.

Recall that we adopt the uncertainty penalty (Definition 1) to measure information loss. We use the iNN distance to establish a lower bound.

Definition 3 (iNN distance): Let E be an equivalence class, $t \in E$ be a tuple in E, and A be a set of attributes. For i $(0 \leq i \leq \|E\|)$, **the i-th nearest neighbor distance** (**iNN distance** for short) of t on A is $NNDist_A(t,i,E) = dist(t, NN(t,i,E))$, where $NN(t,i,E)$ is the i-th nearest neighbor of t in E, and $dist(t_1,t_2) = \sum_{A \in S} \frac{\|t_1[A]-t_2[A]\|}{\|A\|}$ is the minimum uncertainty penalty needed to generalize t_1 and t_2 into the same EC with respect to QID S. ■

We have the following lower bound of information loss.

Theorem 4 (Information loss): Let $E_1, \ldots, E_m$ $(m \geq 1)$ be ECs on $S_1 \cup S_2$ in a table T, and $G = \cup_{i=1}^{m} E_i$ be the set of tuples in those ECs. If a k-butterfly with respect to QIDs S_1 and S_2 is constructed using all tuples in G, then the information loss in the k-butterfly is at least $L(G) = \|G\|(\lambda_{S_1-S_2} + \lambda_{S_2-S_1}) + Loss(S_1 \cap S_2)$, where $Loss(S_1 \cap S_2)$ is the information loss due to the generalization of all tuples in G to the same on $S_1 \cap S_2$, and for $A = S_1 - S_2$ or $S_2 - S_1$, $\lambda_A = \max\{\min_{t \in E \subset G}\{NNDist_A(t, \lceil \frac{k}{m} \rceil - 1, E)\}, \min_{t \in E \subset G}\{NNDist_A(t, k-1, T)\}\}$. ■

4) Step 2: Applying Butterflies: In the second step, we scan in the bottom-up manner the binary hierarchy of ECs built in the first step. For each node N, the tuples in $T(N)$ are anonymized by the children of N using either ECs or butterflies. Thus, the total information loss can be calculated by summing up the loss of all children of N. We compare this loss with the lower bound of information loss using one butterfly on all tuples in $T(N)$ given by Theorem 4. If the lower bound given by Theorem 4 is less, then we construct a butterfly on N.

If all tuples in $T(N)$ are identical on attributes in $S_1 \cap S_2$, then we construct a butterfly on the node. In such a case, the information loss $Loss(S_1 \cap S_2) = 0$ in the butterfly. Moreover, the butterfly allows that the tuples do not need to agree with each other on $S_1 - S_2$ and $S_2 - S_1$, and thus is expected to have lower information loss.

The detailed algorithm is presented in Fig. 2.

```
Input: a binary hierarchy $H$ of ECs, QIDs $S_1$ and $S_2$;
Output: a binary hierarchy $H$ using ECs and butterflies
    to reduce information loss;
Method:
$l = 0$; // $l$ is the total information loss in the current $H$
check all nodes in $H$ in the bottom-up manner,
for each node $N$ do
  if $N$ is a leaf node then
    for each tuple $t \in E$ where $E$ is the EC in $N$ do {
      compute $NNDist_{S_1-S_2}(t,i,E)$ and
      $NNDist_{S_2-S_1}(t,i,E)$ for $i = 1, \ldots, k$;
      calculate the information loss of $N$;
      $l = l+$ the information loss of $N$; }
  else { // $N$ is not a leaf node
    $G =$ the set of tuples in the ECs that are
    descendants of $N$;
    if all tuples in $G$ take the same value on $S_1 \cap S_2$
    then call $butterfly(N)$;
    else {
      let $Loss(G)$ be the sum of information loss in
        children of $N$;
      if $Loss(G) > L(G)$ in Theorem 4
      then call $butterfly(N)$;
      if $Loss(S_1 \cap S_2) \geq l$ at $N$
      then prune all ancestors of $N$;
    }  }
```

Fig. 2. Applying butterflies to reduce information loss.

5) Building Butterflies: Let G be the set of tuples on which we want to construct a k-butterfly. We conduct the following two steps. First, we generalize all tuples in G to the same on attributes in $S_1 \cap S_2$ and calculate the information loss $Loss(S_1 \cap S_2)$ by one scan of all tuples in G. Second, we use a k-anonymization algorithm to anonymize tuples in G on QID $S_1 - S_2$. This step constructs the "left wings" of the butterfly. Similarly, we apply the anonymization algorithm to anonymize tuples in G on QID $S_2 - S_1$ for the "right wings" of the butterfly.

Once the butterfly is computed at node N, we calculate the information loss in the butterfly. If it is smaller than the sum of information loss of the children of N, then the butterfly replaces the children of N, and the information of N is updated. Consequently, we also update the total information loss of the whole data set (i.e., variable l in Fig. 2) accordingly.

In general, our approach can be extended to handle cases where there are more than two QIDs. In addition, we can provide privacy preservation when users may collude. Limited by space, we have to omit the details.

Acknowledgements

Jian Pei's research was supported in part by an NSERC Discovery grant and an NSERC Discovery Accelerator Supplements grant. The research of Yufei Tao, Jiexing Li and Xiaokui Xiao was supported by grants CUHK 4161/07 and CUHK 1202/06 from HKRGC. All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] J. Pei *et al.* Privacy preserving publishing on multiple quasi-identifiers. *Technical Report TR 2008-18*, School of Comping Science, Simon Fraser University, November 2008.

[2] J. Xu *et al.* Utility-based anonymization using local recoding. In *KDD'06*.

[3] G. Ghinita *et al.* Fast data anonymization with low information loss. In *VLDB'07*.

[4] A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In *PODS'04*.

[5] K. LeFevre *et al.* Mondrian multidimensional k-anonymity. In *ICDE'06*.