Cross-Bucket Generalization for Information and Privacy Preservation

Boyu Li, Yanheng Liu, Xu Han, and Jindong Zhang

Abstract—Generalization is an effective technique for protecting confidential information of individuals, and has been studied by proposing numerous algorithms. However, the previous works do not separate the protection against identity disclosure and sensitive disclosure. Thus, when the requirement of attribute protection is higher than that of identity protection, generalization for l-diversity causes overprotection for identity and large mounts of information utility loss. This paper presents a novel approach, called cross-bucket generalization, as a solution to meet the problem. The rationale is to divide microdata into equivalence groups and buckets. First, it provides separate protection for identity and sensitive values, and the level of protection can be flexibly adjusted based on actual demands. Second, the sizes of equivalence groups and buckets are minimized as far as possible by only satisfying the protection requirements, which avoid the overprotection for identity and reduce information loss. The experiments we conducted illustrate the effectiveness of our solution.

Index Terms—Privacy preservation, data publication, generalization, bucketization, k-anonymity, l-diversity

1 Introduction

In the past several years, microdata release has posed threats to individual privacy and organizational confidentiality. According to Sweeney (see [1]), 87 percent of the population in the United States had reported characteristics that likely made them unique based on particular attributes. Access to these data need to be safeguarded for the safety and security of the people. An adverse effect could be the unwarranted use of microdata.

For example, suppose that an adversary has access to the voter registration list (Fig. 1a), and obtains the microdata released by a hospital (Fig. 1b). Knowing that Helen went to the hospital before and combining her sex, age, and zip code values, the adversary can determine that: (1) her record is with ID "106" in the microdata; and (2) her disease was pneumonia.

The goal of preventing of such privacy disclosures has resulted in the development of many anonymization techniques (see surveys [2], [23]), such as two popular ones, namely, generalization and bucketization. The attributes are further partitioned into three categories: (1) Explicit-Identifier, such as name and security number, which can uniquely or mostly identify the record owner and must be removed from the published table; (2) Quasi-Identifier (QI), such as age, sex, and zip code, which can reveal the identity of a record owner when taken together; and (3) Sensitive attribute that contains the confidential information of individuals, such as salary and disease.

Generalization [6], [10], [12], or transforming the QI values into more general forms, divides the tuples into equivalence groups in which the values of each QI attribute are the same. Thus, the records in the same equivalence group are indistinguishable. In particular, a generalized table is k-anonymous [1] when the size of each equivalence group is at least k. Fig. 2a shows a 2-anonymous generalized table of Fig. 1b, and the probability that the identity disclosure of any tuple is exposed is at most $1/2$.

Bucketization [5], [7], [8] partitions the tuples into buckets in which the relation between the sensitive attributes and QI attributes is broken, such that each record in the bucketized table corresponds to multiple sensitive values. The bucketized table always complies with l-diversity [4] principle, i.e., for each tuple, the probability that the sensitive value is exposed is at most $1/l$. For example, Fig. 2b provides the bucketized table of Fig. 1b that satisfies the 4-diversity condition. Any tuple associated with a sensitive value inside its bucket has an equal probability that is no more than $1/4$.

1.1 Motivation

Although generalization for k-anonymity can effectively prevent identity disclosure, this technique cannot provide sufficient protection for sensitive values. For instance, in Fig. 2a, the sensitive values in the second equivalence group are both bronchitis; thus, an adversary can infer that the tuples with IDs "104" and "105" both have bronchitis. One feasible solution is to let generalization satisfy the l-diversity principle. Fig. 3 illustrates the generalized table that

- *B. Li, X. Han, and J. Zhang are with the College of Computer Science and Technology, Jilin University, Changchun 130012, China. E-mail: {afterslby, 18204316731, zhangjindong_100}@163.com.*
- *Y. Liu is with the College of Computer Science and Technology and Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China. E-mail: yhliu@jlu.edu.cn.*

Manuscript received 26 Oct. 2016; revised 30 Aug. 2017; accepted 6 Nov. 2017. Date of publication 14 Nov. 2017; date of current version 2 Feb. 2018. (Corresponding author: Jindong Zhang.)
Recommended for acceptance by R. Meo.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2017.2773069

Name	Sex	Age	Zip code
Tina	Female	26	22105
Helen	Female	31	43312
Louis	Male	42	01274
Rachel	Female	34	02732

(a)

ID	Age	Sex	Zip code	Disease
101	16	Female	43307	Flu
102	22	Male	43302	Dyspepsia
103	24	Female	43306	Hepatitis
104	26	Male	43307	Bronchitis
105	29	Male	43309	Bronchitis
106	31	Female	43312	Pneumonia
107	34	Female	43312	Gastritis
108	35	Male	43309	Dyspepsia

(b)

Fig. 1. (a) The voter registration list and (b) the microdata.

ID	Age	Sex	Zip code	Disease
101	[16-24]	*	[43302-43307]	Flu
102	[16-24]	*	[43302-43307]	Dyspepsia
103	[16-24]	*	[43302-43307]	Hepatitis
104	[26-29]	Male	[43307-43309]	Bronchitis
105	[26-29]	Male	[43307-43309]	Bronchitis
106	[31-35]	*	[43309-43312]	Pneumonia
107	[31-35]	*	[43309-43312]	Gastritis
108	[31-35]	*	[43309-43312]	Dyspepsia

(a)

ID	Age	Sex	Zip code	Disease
101	16	Female	43307	Dyspepsia
102	22	Male	43302	Flu
103	24	Female	43306	Bronchitis
104	26	Male	43307	Hepatitis
105	29	Male	43309	Gastritis
106	31	Female	43312	Dyspepsia
107	34	Female	43312	Pneumonia
108	35	Male	43309	Bronchitis

(b)

Fig. 2. (a) The generalized table and (b) the bucketized table.

complies with the 4-diversity condition, which provides the same protection over sensitive values as Fig. 2b. The requirement for the protection for sensitive values also increases the level of the protection for identity (e.g., the generalized table shown in Fig. 3 is compelled to satisfy 4-anonymity). The results are overprotection for identity and more information loss. When the generalization approach has a stringent requirement on the protection of sensitive values, large amounts of information utility losses occur.

Although bucketization for l-diversity limits the boundary of attribute disclosure under a constant value, the technique preserves all original QI values for excellent information utility, even with the high requirements of attribute protection. The major shortcoming of a bucketized table is that it cannot defend against identity disclosure. And if the adversary has not determined the information, he can hardly be prevented from learning that the record of a person is included in the bucketized table. For example, in Fig. 2b, the adversary is likely to recognize the record of Helen in the bucketized table, then infer that Helen previously fell ill and went to the hospital.

A stronger identity protection of generalization for k-anonymity is ensured when a bigger parameter k is deployed. However, a larger equivalence group always needs the larger ranges of the generalized QI values that decrease the information utility. Similarly, for bucketization that satisfies l-diversity, the attribute protection is enhanced when parameter l increases. A bigger bucket size means less association of a record with its original sensitive value, which also causes more information loss. Therefore, a possible solution to reduce information loss is minimizing the sizes of equivalence groups or buckets on the objective of satisfying the principle requirement.

In this study, we presume that the requirement of attribute protection is higher than that of identity protection and propose cross-bucket generalization as a solution. This anonymization technique prevents both identity and attribute disclosure while preserving significant information utility requires stringent measures for the protection of sensitive values. It separates the protection for identity and sensitive values. Cross-bucket generalization partitions the tuples into equivalence groups that satisfies the requirement of identity protection, then divides the generalized tuples into buckets to break their linkages between QI values and sensitive values. The detailed formalization and analysis of cross-bucket generalization are presented in Section 2. Fig. 4 shows a possible result of Fig. 1b by cross-bucket generalization.

The advantages of cross-bucket generalization are as follows. First, it provides separate protection for identity and sensitive values by establishing different sets of requirements for identity protection and attribute protection. The level of protection can be flexibly adjusted based on actual demands. For instance, the cross-bucket generalized table shown in Fig. 4 complies with 2-anonymity and 4-diversity, which is more flexible than the generalized table shown in Fig. 3.

Cross-bucket generalization reduces the sizes of equivalence groups and buckets as far as possible by satisfying the protection requirements only. For example, the size of each equivalence group in the cross-bucket generalized table shown in Fig. 4 is 2, which avoids identity overprotection in the generalized table shown in Fig. 3. The size of each bucket is likewise 2, but the power of the protection

ID	Age	Sex	Zip code	Disease
101	[16-26]	*	[43302-43307]	Flu
102	[16-26]	*	[43302-43307]	Dyspepsia
103	[16-26]	*	[43302-43307]	Hepatitis
104	[16-26]	*	[43302-43307]	Bronchitis
105	[29-35]	*	[43309-43312]	Bronchitis
106	[29-35]	*	[43309-43312]	Pneumonia
107	[29-35]	*	[43309-43312]	Gastritis
108	[29-35]	*	[43309-43312]	Dyspepsia

Fig. 3. The generalized table for 4-diversity.

ID	Age	Sex	Zip code	Disease
101	[16-24]	Female	[43306-43307]	Flu
102	[22-26]	Male	[43302-43307]	Dyspepsia
103	[16-24]	Female	[43306-43307]	Hepatitis
104	[22-26]	Male	[43302-43307]	Bronchitis
105	[29-35]	Male	43309	Bronchitis
106	[31-34]	Female	43312	Pneumonia
107	[31-34]	Female	43312	Gastritis
108	[29-35]	Male	43309	Dyspepsia

Fig. 4. The cross-bucket generalized table.

for sensitive values is the same as the bucketized table shown in Fig. 2b.

1.2 Contributions

Our contributions are as follows. First, a cross-bucket generalization approach for privacy-preserving data publication is presented. Cross-bucket generalization partitions tuples into equivalence groups and buckets to prevent both identity and attribute disclosure with significantly less information loss than generalization. The primary consideration is the high protection requirement for sensitive values.

Second, we illustrate the effective protection of cross-bucket generalization for identity and sensitive values based on the principles of k-anonymity and l-diversity. The study introduces the (k, l)-anonymous cross-bucket generalization concept, such that for each tuple, the probabilities of identity disclosure and attribute disclosure are, at most, $1/k$ and $1/l$, respectively.

Third, an algorithm combining generalization and bucketization approaches to achieve (k, l)-anonymous cross-bucket generalization is presented. The sizes of equivalence groups and buckets are minimized to meet the protection requirements and reduce information loss. Subsequently, the range of the generalized values in each equivalence group is narrowed to the maximum possible digits.

Finally, experiments were conducted to evaluate the cross-bucket generalization algorithm vis-à-vis generalization and bucketization in two aspects: (1) cross-bucket generalization provides better protection for sensitive values than the other techniques; and (2) cross-bucket generalization preserves data utility more effectively than generalization, as shown by the results of the discernibility penalty and query answerability. The effect of adjusting the parameter k when the parameter l is fixed is also studied.

The rest of this paper is organized as follows: Section 2 proposes the formalization of the cross-bucket generalization technique, and defines (k, l)-anonymous cross-bucket generalization for privacy preservation. Section 3 presents an algorithm to achieve (k, l)-anonymous cross-bucket generalization. Section 4 shows the results and analysis of the experiments. Section 5 describes related studies. Section 6 concludes the paper and proposes directions for future studies.

2 Cross-Bucket Generalization

Suppose that a microdata T consists of d QI attributes $A_1^{QI}, A_2^{QI}, \ldots, A_d^{QI}$ and a sensitive attribute A^{SA}. Each attribute can be either numerical or categorical, and $D[A_i]$ denotes the domain of attribute A_i. For any tuple $t \in T$, $t[A_i]$ represents its value on attribute A_i.

2.1 Concepts

The formalization of the cross-bucket generalization technique requires a prior review of certain basic concepts.

Definition 1 (Partition and QI Group). *A partition consists of several subsets of T, such that each tuple belongs to exactly one subset. Each subset of tuples is called a QI group. Specifically, let there be m QI groups $\{G_1, G_2, \ldots, G_m\}$, then $\bigcup_{i=1}^{m} G_i = T$, and for any $1 \le i_1 \ne i_2 \le m$, $G_{i_1} \cap G_{i_2} = \emptyset$.*

Depending on different anonymization approaches, each QI group has different performances. In generalization, the values of each QI attribute are simplified to the same form inside each QI group. In bucketization, each QI group is divided into two sub-tables, each containing QI values and sensitive values, respectively.

Definition 2 (Equivalence Group). *Given a partition of T with m QI groups, and each QI group is called an equivalence group, if for any tuple $t \in T$, a generalized table of T contains the tuple t of the form*

$$(G_j[1], G_j[2], \ldots, G_j[d], t[A^{SA}]),$$

where $G_j (1 \le j \le m)$ is the unique QI group including t, and $G_j[i] (1 \le i \le d)$ is the value of the attribute A_i^{QI} for all the tuples in G_j.

Definition 3 (Bucket). *Given a partition of T with m QI groups, and each QI group is called a bucket, such that each QI group is represented as the form*

$$QIT(QI, GID) \text{ and } SAT(SA, GID),$$

where QI and SA are the QI values and sensitive values of the tuples in the QI group, respectively, and GID denotes the group id.

A data table is considered a privacy preservation table if the data table satisfies the anonymization principle. The basic condition for a generalized table is k-anonymity.

Definition 4 (k-Anonymity). *Table T complies with k-anonymity if the values of $t[A^{QI}]$ appear at least k times in T for each $t \in T$.*

However, k-anonymity cannot prevent attribute disclosure when homogeneity and background knowledge are used in alleged adversary attacks. Machanavajjhala et al. proposed that for l-diversity to improve k-anonymity, the sensitive values in the QI group should be l "well-represented", and the probability of attribute disclosure can be limited under a constant value.

Definition 5 (Breach Probability). *For any tuple $t \in T$, its breach probability is set as $p(t, v)$, which is equal to the probability that the value v of t is exposed.*

Definition 6 (l-Diversity). *Table T complies with l-diversity if each $t \in T$ satisfies*

$$p(t, s) \le 1/l,$$

where s is the sensitive value of t.

By Definitions 2 and 3, we define cross-bucket generalization technique as:

Definition 7 (Cross-Bucket Generalization). *Given a microdata T, a cross-bucket generalization of T is given by the partitions of equivalence groups and buckets, and each tuple belongs to exactly one equivalence group and one bucket. Specifically, let there be m equivalence groups $\{EG_1, EG_2, \ldots, EG_m\}$ and n buckets $\{B_1, B_2, \ldots, B_n\}$, then $\bigcup_{i=1}^{m} EG_i = T$ and $\bigcup_{j=1}^{n} B_j = T$, and for any $1 \le i_1 \ne i_2 \le m$, $EG_{i_1} \cap EG_{i_2} = \emptyset$ and for any $1 \le j_1 \ne j_2 \le n$, $B_{j_1} \cap B_{j_2} = \emptyset$.*

A key concept of cross-bucket generalization is that of matching bucket.

Definition 8 (Matching Bucket). *Given an anonymized table T by cross-bucket generalization, and for any tuple $t \in T$, MB is a matching bucket of t in T if $t[A^{QI}] \in MB[A^{QI}]$, where $t[A^{QI}]$ is the QI values of t and $MB[A^{QI}]$ is the set of QI values in MB.*

To illustrate how a cross-bucket generalized table prevents privacy disclosures, we assume that an adversary already knows the QI values of Helen from Fig. 1a, and attempts to acquire her disease value from Fig. 4. First, by matching QI values, the adversary can determine that the tuple with ID "106" or "107" is Helen, then infer that the matching buckets of Helen are the third bucket and forth bucket. Because the sensitive values in the matching buckets have equal possibilities to be correct, then her possible disease values are bronchitis, pneumonia, gastritis, and dyspepsia.

2.2 Privacy Preservation

In this section, we analyze in detail the protection of cross-bucket generalization against identity disclosure and attribute disclosure, then introduce the notion of (k, l)-anonymous cross-bucket generalization.

We first consider the protection against identity disclosure in the cross-bucket generalized table, and have the following lemma and corollary.

Lemma 1. *Given a cross-bucket generalized table, for any tuple $t \in T$, the probability of identity disclosure is at most $1/|G(t)|$, where $G(t)$ is the equivalence group containing t.*

Proof. According to Definition 7, the cross-bucket generalized table is divided into equivalence groups, and each tuple belongs to exactly one equivalence group. The adversary can obtain at least $|G(t)|$ possible records by matching the QI values of t because the QI values of t are generalized with other records in $G(t)$. Therefore, the probability of identity disclosure of t is at most $1/|G(t)|$. □

Corollary 1. *A cross-bucket generalized table can satisfy the k-anonymity principle by restricting the least size of the equivalence groups at k.*

Proof. From Lemma 1, the probability of identity disclosure for any tuple in the cross-bucket generalized table corresponds to the size of the equivalence group containing the tuple. Therefore, if the least size of the equivalence groups is k, such that every tuple is indistinguishable with at least other $k-1$ tuples, then the cross-bucket generalized table must satisfy the k-anonymity principle. □

Next, the protection against attribute disclosure is discussed. According to Definition 7, the cross-bucket generalized table is also partitioned into buckets, and the tuples in the same equivalence group may be assigned into different buckets. Therefore, the adversary first needs to determine the buckets where the target tuple is located, that is, matching buckets, through the QI values of the target tuple.

Definition 9 (Bucket Location Probability). *For any tuple $t \in T$, its bucket location probability, denoted as $p(t, B)$, equals the probability that t is in the bucket B.*

Lemma 2. *Given a cross-bucket generalized table, for any tuple $t \in T$, the probability that the sensitive value s of t is exposed is as follows:*

$$p(t,s) \leq \sum_{MB} p(t, MB) \frac{|MB(s')|}{|MB|},$$

where $|MB(s')|$ is the number of the most occurrence sensitive value s' in the matching bucket MB, and $|MB|$ is the size of MB.

Proof. To acquire the sensitive value s of the target tuple $t \in T$, the adversary calculates the probability of t in each bucket location, and calculate the probability that t carries the sensitive value s in each bucket. Thus the adversary has

$$p(t,s) = \sum_{B} p(t, B) p(s|t, B),$$

where $p(s|t, B)$ denotes the probability that t takes a sensitive value s, given that t is in bucket B. The adversary can eliminate the buckets that does not contain QI values of t, expressed as follows:

$$p(t, B) = 0, \text{if } t[A^{QI}] \notin B[A^{QI}].$$

According to Definition 8, we have

$$p(t,s) = \sum_{MB} p(t, MB) p(s|t, MB),$$

where MB is the matching bucket of t. The highest occurring sensitive value s' in MB is expressed as

$$|MB(s)| \leq |MB(s')|.$$

Thus

$$p(s|t, MB) = \frac{|MB(s)|}{|MB|} \leq \frac{|MB(s')|}{|MB|},$$

then

$$\begin{aligned} p(t,s) &= \sum_{MB} p(t, MB) p(s|t, MB) \\ &\leq \sum_{MB} p(t, MB) \frac{|MB(s')|}{|MB|}. \end{aligned}$$

□

Corollary 2. *A cross-bucket generalized table can satisfy the l-diversity principle by confining each tuple as follows: (1) each sensitive value appears only once in the matching buckets of the tuple; and (2) the matching bucket that contains the correct sensitive value satisfies the following expression:*

$$\frac{p(t, MB)}{|MB|} \leq \frac{1}{l}.$$

Proof. According to Lemma 2, for any tuple $t \in T$

$$p(t,s) \leq \sum_{MB} p(t, MB) \frac{|MB(s')|}{|MB|}.$$

We confine each sensitive value that appears only in the matching buckets of t. Therefore, only one matching bucket MB' contains the sensitive value s of t, such that

$$\begin{aligned} p(t,s) &= \sum_{MB} p(t, MB) p(s|t, MB) \\ &= p(t, MB') p(s|t, MB'), \end{aligned}$$

and for any matching bucket MB of t, we have

$$\frac{|MB(s')|}{|MB|} = \frac{1}{|MB|}.$$

Then

$$p(t,s) \leq \sum_{MB} p(t,MB)\frac{|MB(s')|}{|MB|} = \frac{p(t,MB')}{|MB'|}.$$

The second condition is expressed as

$$p(t,s) \leq \frac{p(t,MB')}{|MB'|} \leq 1/l.$$

Thus, the cross-bucket generalized table complies with l-diversity under the following conditions. □

Corollaries 1 and 2 present the theoretical basis that cross-bucket generalization can be limited to satisfy k-anonymity and l-diversity. Therefore, we define (k, l)-anonymous cross-bucket generalization as follows:

Definition 10 ((k, l)-Anonymous Cross-Bucket Generalization). *A cross-bucket generalized table is (k, l)-anonymous if for any tuple $t \in T$, the probability of the identity disclosure of t is at most $1/k$, and*

$$p(t,s) \leq 1/l,$$

where s is the sensitive value of t.

3 Cross-Bucket Generalization Algorithm

This section presents an algorithm to implement (k, l)-anonymous cross-bucket generalization. We aim to achieve the following two goals. First, the size of each equivalence group and bucket should be minimized to satisfy only the principle requirement. Second, the ranges of the generalized QI values should be minimized. The procedure is presented in Algorithm 1.

Algorithm 1. Cross-Bucket Generalization(T, k, l)

```
1: T_ori = T
2: T_anony = φ
3: tuple_count = |T|
4: diversity_num = cal_diversity(k, l)
5: while tuple_count > 0 do
6:    S_set, loop_num = cal_sen_set(T_ori, k, diversity_num)
7:    while loop_num > 0 do
8:       T_gen = pick_tuples(T_ori, S_set)
9:       bucket_set = divide_tuples(T_gen, k)
10:      T_anony = T_anony ∪ bucket_set
11:      T_ori = T_ori − T_gen
12:      tuple_count = tuple_count − |T_gen|
13:      loop_num = loop_num − 1
14:   end while
15: end while
16: return T_anony
```

The data structures T_{ori} and T_{anony} store the original data and the anonymized result, respectively (lines 1 and 2). The variable $tuple\_count$ (line 3) denotes the number of the tuples that has not been generalized. The function $cal\_diversity(k,l)$ returns a variable $diversity\_num$ (line 4), which will be the parameter to calculate the sensitive set (line 6). In each iteration (lines 5 to 15), the algorithm first calculates a set of sensitive values S_{set} and a number of loops $loop\_num$ (line 6). Then, in each iteration (lines 7 to 14), the algorithm selects eligible tuples T_{gen} from T_{ori}, based on S_{set} (line 8). Subsequently, the algorithm generalizes the selected tuples T_{gen} and divides them into buckets (line 9), then adds the output $bucket\_set$ to T_{anony} (line 10). The algorithm eliminates the generalized tuples T_{gen} from T_{ori} (line 11) and updates the values of $tuple\_count$ and $loop\_num$ (lines 12 and 13). Finally, the algorithm returns the anonymized result T_{anony} after all the tuples have been generalized (line 16).

The algorithm comprises three main parts: sensitive set calculation (line 6), tuple selection (line 8), and tuple division (line 9). We elaborate each part in the rest of this section.

3.1 Sensitive Set Calculation

In this phase, the algorithm calculates a set of sensitive values and a number of loops. The sensitive set will be the input for picking tuples in the phase of tuple selection until the number of loops is reduced to zero. We modify and use the assignment algorithm of m-Invariance [18] to implement the function $cal\_sen\_set(T,k,diversity\_num)$ in Algorithm 1. Algorithm 2 presents the description.

Algorithm 2. cal_sen_set(T, k, $diversity\_num$)

```
1: sen_domain = {value, count | value ∈ T[A^SA], order by count desc}
2: if |sen_domain| < 2 * diversity_num then
3:    β = |sen_domain|
4:    α = the least count in sen_domain
5: else
6:    β = diversity_num
7:    α = calculate_alpha(sen_domain, β)
8:    while α does not exist do
9:       β = β + k
10:      if |sen_domain| − β < 2l then
11:         β = |sen_domain|
12:         α = the least count in sen_domain
13:         break
14:      end if
15:      α = calculate_alpha(sen_domain, β)
16:   end while
17: end if
18: S_set = {the first β values in sen_domain}
19: return S_set, α
```

The meanings of symbols α and β, and their computation methods are the same as discussed in m-Invariance. The input $diversity\_num$ is obtained by the function $cal\_diversity(k,l)$ in line 4 of Algorithm 1, which is the smallest number larger than or equal to l and divisible by k. The condition in line 2 checks whether it is the last call of $cal\_sen\_set(T,k,diversity\_num)$ in Algorithm 1. If yes, the algorithm returns the rest of distinct sensitive values as the result (line 3). Otherwise, β is initialed as $diversity\_num$ (line 6), and the step size is k (line 9). The condition in line 10 aims to maintain the size of S_{set} in the last iteration in Algorithm 1 as larger than $2l$.

According to Algorithm 2, S_{set} has the following properties.

Property 1. *The sensitive values in S_{set} are unique.*

Property 2. *The size of S_{set} is not less than l if $|S_{set}|$ is divisible by k; else, $|S_{set}|$ is larger than $2l$.*

3.2 Tuple Selection

The algorithm selects the proper tuples from T_{ori} based on S_{set} achieved by the previous phase. The selected tuples should have a one-to-one correspondence to the sensitive values in S_{set}, and the interval of their QI values should be minimized. We modify the Mondrian algorithm [3] to achieve the function $pick\_tuples(T, S_{set})$ in Algorithm 1. The description is expressed as Algorithm 3.

Algorithm 3. pick_tuples(T, S_{set})

1: $attri\_QI\_set = \{attri | attri \in set\ of\ A^{QI}\}$
2: **while** $attri\_QI\_set \neq \emptyset$ **do**
3: $attribute \leftarrow choose\_attri(T, attri\_QI\_set)$
4: $median \leftarrow cal\_median(T, attribute)$
5: $T_l \leftarrow \{t \in T : attribute(t) \leq median\}$
6: $T_r \leftarrow \{t \in T : attribute(t) > median\}$
7: **if** $check\_condition(T_l, S_{set})$ **then**
8: **pick_tuples**(T_l, S_{set})
9: **break**
10: **else if** $check\_condition(T_r, S_{set})$ **then**
11: **pick_tuples**(T_r, S_{set})
12: **break**
13: **else**
14: $attri\_QI\_set = attri\_QI\_set - attribute$
15: **end if**
16: **end while**
17: **if** $attri\_QI\_set = \emptyset$ **then**
18: $T_{gen} = select\_tuples(T, S_{set})$
19: **return** T_{gen}
20: **end if**

The data structure $attri\_QI\_set$ is initialized as the set of QI attributes (line 1). In each iteration (lines 2 to 16), the algorithm chooses an attribute (line 3), counts its median value (line 4), then divides T into two smaller parts (lines 5 and 6). The function $check\_condition(T, S_{set})$ checks whether T contains all the sensitive values in S_{set}. The branch statement (lines 7 to 15) checks either of the smaller part T_l or T_r that satisfies the condition. If the condition is satisfied, the algorithm recursively calls $pick\_tuples(T_l, S_{set})$ (line 8) or $pick\_tuples(T_r, S_{set})$ (line 11). Otherwise, the chosen attribute is eliminated from $attri\_QI\_set$ (line 14). If $attri\_QI\_set$ is empty, not a single attribute can be used to divide T into the smaller table that contains the entire sensitive values in S_{set} (line 17). The function $select\_tuples(T, S_{set})$ returns a set of tuples from T in which each sensitive value in S_{set} is carried by a tuple (line 18). Finally, the algorithm returns the set of the selected tuples T_{gen} (line 19), which will be generalized and divided into buckets in the next phase to satisfy the (k, l)-anonymity requirement.

The analysis of the time complexity of Algorithm 3 states that in the worst case, its recursion is equal to that of Mondrian. Given that the height of kd-tree is $logn$, the time complexity of Algorithm 3 is $O(log|T|)$, where $|T|$ is the size of T.

3.3 Tuple Division

The function $divide\_tuples(T_{gen}, k)$ in Algorithm 1 contains two parts. The first part generalizes T_{gen} to comply with k-anonymity. Algorithm 4 describes the generalization process.

Algorithm 4. generalize_tuples(T_{gen}, k)

1: **if** $|T_{gen}|$ *is divisible by* k **then**
2: $index = 0$
3: **while** $index < |T_{gen}|$ **do**
4: $generalize\_oper(T_{gen}, index, k)$
5: $index = index + k$
6: **end while**
7: **else**
8: $group\_num = |T_{gen}|/k$
9: $remainder\_num = |T_{set}|\%k$
10: $per\_base\_num = remainder\_num/group\_num$
11: $per\_remainder\_num = remainder\_num\%group\_num$
12: $index = 0$
13: **for** $i = 0$ to $per\_remainder\_num$ **do**
14: $generalize\_oper(T_{gen}, index, k + per\_base\_num + 1)$
15: $index = index + k + per\_base\_num + 1$
16: **end for**
17: **for** $i = per\_remainder\_num$ to $group\_num$ **do**
18: $generalize\_oper(T_{gen}, index, k + per\_base\_num)$
19: $index = index + k + per\_base\_num$
20: **end for**
21: **end if**
22: **return** T_{gen}

The condition in line 1 checks whether the size of T_{gen} is divisible by k. If yes, T_{gen} can be divided into equivalence groups (lines 2 to 6), where the function $generalize\_oper(T_{gen}, index, step)$ generalizes the tuples in T_{gen}, starting from $index$th with the step at $step$. The algorithm calculates the number of equivalence groups (line 8) and remainder tuples (line 9). The average number of the remainder tuples is distributed in each equivalence group (lines 10 and 11) and evenly partitioned into equivalence groups (lines 12 to 20).

Proposition 1. *After the generalization phase, T_{gen} complies with k-anonymity.*

Proof. According to Algorithm 4, each call is expressed as: $generalize\_oper(T_{gen}, index, step)$, in which the value of $step$ is not less than k. Thus, the size of every equivalence group in T_{gen} is at least k. Due to Corollary 1, T_{gen} complies with k-anonymity. □

Next, T_{gen} is divided into buckets that break the relations between QI values and sensitive values to satisfy l-diversity. Algorithm 5 presents the partition process. The data structure $bucket\_set$ is used to store the result buckets (line 1). In each iteration (lines 2 to 10), the algorithm generates an empty bucket (line 3) and adds the jth tuple from T_{gen} into the bucket with the step at k through the while loop (lines 5 to 8). The algorithm then adds the bucket into $bucket\_set$ (line 9). Finally, the algorithm returns $bucket\_set$ as a result (line 11).

Proposition 2. *After the partition phase, T_{gen} complies with l-diversity.*

Proof. The sensitive values in T_{gen} have the same properties as S_{set} because the sensitive values carried by T_{gen} corresponds with S_{set}. According to Property 1, the sensitive values are unique. Therefore, each tuple satisfies the first condition in Corollary 2 because all matching buckets are obtained inside T_{gen}.

TABLE 1
Description of the Attributes

	Attribute	Type	Size
1	Sex	Categorical	2
2	Age	Continuous	73
3	Relationship	Categorical	13
4	Marital status	Categorical	6
5	Race	Categorical	9
6	Education	Categorical	11
7	Hours per week	Continuous	90
8	Occupation	Categorical	216
9	Salary	Continuous	702

We then show how each tuple satisfies the second condition in Corollary 2. We consider two cases of T_{gen}. First, the size of T_{gen} is divisible by k. According to Algorithm 4, every k tuple comprises an equivalence group, starting from the first group in T_{gen}. In Algorithm 5, every new bucket only contains one tuple from each equivalence group after each loop. Given that the sizes of equivalence groups are all k, then

$$p(t, MB) = \frac{1}{k},$$

and

$$|MB| = \frac{|T_{gen}|}{k}.$$

According to Property 2, $|T_{gen}|$ is not less than l, thus

$$\frac{p(t, MB)}{|MB|} = \frac{1}{k|MB|} = \frac{1}{|T_{gen}|} \leq \frac{1}{l}.$$

Next we consider the scenario that $|T_{gen}|$ is not divisible by k. According to Algorithm 4, two different sizes of equivalence groups are identified: $k + per\_base\_num + 1$ and $k + per\_base\_num$. Given that $remainder\_num$ is smaller than k, these two values must be smaller than $2k$. In Algorithm 5, the generated bucket contains at most two tuples from the same equivalence group, expressed as

$$p(t, MB) \leq \frac{2}{k},$$

and

$$|MB| \geq \frac{|T_{gen}|}{k}.$$

According to Property 2, $|T_{gen}|$ is bigger than $2l$, expressed as

$$\frac{p(t, MB)}{|MB|} \leq \frac{2}{k|MB|} \leq \frac{2}{|T_{gen}|} \leq \frac{1}{l}.$$

Thus, each tuple must satisfy the conditions in Corollary 2 for T_{gen} to comply with l-diversity. □

4 Experiments

This section evaluates the efficiency of the cross-bucket generalization algorithm proposed in Section 3. We use the real US Census data [9], eliminate the tuples with missing values, and randomly select 22,517 tuples with nine attributes. The QI attributes include sex, age, relationship, marital status, race, education, hours per week, and occupation. Salary is assigned as the sensitive attribute. Table 1 presents the attributes in detail.

Algorithm 5. partition_tuples(T_{gen})

```
1: bucket_set = ∅
2: for i = 0 to k do
3:    bucket = ∅
4:    j = i
5:    while j < |T_gen| do
6:       add T_gen[j] to bucket
7:       j = j + k
8:    end while
9:    bucket_set = bucket_set ∪ bucket
10: end for
11: return bucket_set
```

We implement the Mondrian Algorithm [3] and Anatomy Algorithm [7] to compare cross-bucket generalization in two aspects, namely, privacy protection and information preservation. Mondrian Algorithm partitions the microdata table into non-overlapping equivalence groups by the domains of QI attributes. While Anatomy Algorithm divides the microdata table into buckets in which the sensitive values are unique and keep all the raw QI values. Additionally, we experiment on the effect of the parameter k for (k, l)-anonymous cross-bucket generalization when the parameter l is fixed.

4.1 Privacy Protection

In this experiment, we check the disclosure probability of the sensitive value for each tuple in the original table and calculate the average results for comparison. Suppose the adversary already knows the QI values of all the tuples and their existence in the anonymized tables, then attempts to acquire the sensitive values of the tuples by matching their QI values. For a tuple t in the anonymized table that is not divided into buckets, we count the number of matching tuples of t as $Num_{tuples}(t)$, and the number of the matching tuples that carries the sensitive value s of t as $Num_{sen}(t)$, expressed as

$$p(t, s) = Num_{sen}(t)/Num_{tuples}(t).$$

While the anonymized table partitioned into buckets is expressed as

$$p(t, s) = \sum_{MB} p(t, MB)p(s|t, MB),$$

where MB is the matching bucket of t. We also count $Num_{tuples}(t)$, the number of the matching tuples in MB is denoted as $Num(t, MB)$, and the number of the sensitive value s of t in MB is denoted as $Num(s, MB)$. Thus

$$p(t, s) = \sum_{MB} \frac{Num(t, MB) * Num(s, MB)}{Num_{tuples}(t) * |MB|},$$

where $|MB|$ is the size of MB.

Mondrian and Anatomy comply with l-diversity and fix k at 3 for cross-bucket generalization. Fig. 5 shows the results of the proportion to which the salary values are exposed. Anatomy performs its expected protection that abides by l-diversity. Mondrian is more effective than Anatomy

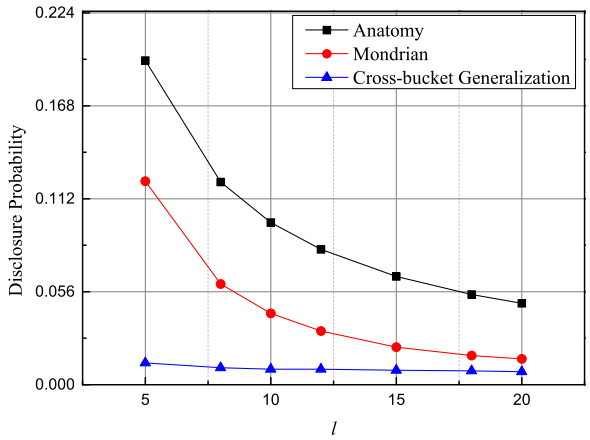

Fig. 5. Salary breach probabilities.

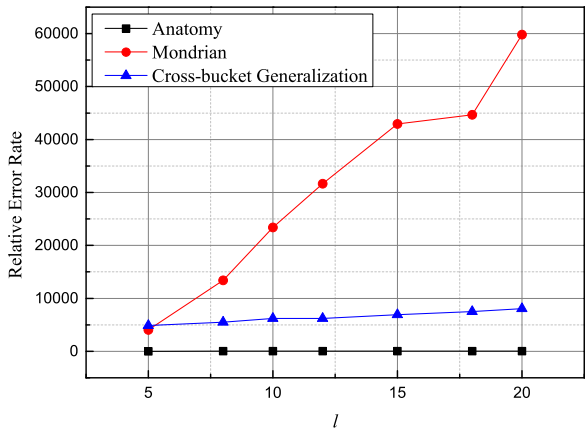

Fig. 7. Relative error results.

because the sizes of equivalence groups are always larger than those of buckets, which hinder the adversary from obtaining the accurate salary values. However, the disclosure probability of cross-bucket generalization is significantly lower than that of other techniques. In the cross-bucket generalized table, the tuples in the same equivalence group are assigned into different buckets, and the equivalence groups likely tend to overlap. Therefore, the adversary must acquire several matching buckets by matching QI values.

4.2 Information Preservation

This experiment compares the information quality between cross-bucket generalization and Mondrian by two approaches. The first one uses the discernibility metric measurement [19], denoted as C_{DM}, which is given by the equation

$$C_{DM} = \sum_{EG} |EG|^2,$$

where EG is the equivalence group. A smaller value of C_{DM} means less generalization and perturbation in the anonymization process, whereas a bigger value means more information loss in the generalized table.

In this experiment, Mondrian complies with l-diversity, and the k is fixed at 3 for cross-bucket generalization, which uses the same configuration as the previous experiment. Fig. 6 presents the results of the discernibility metric. The

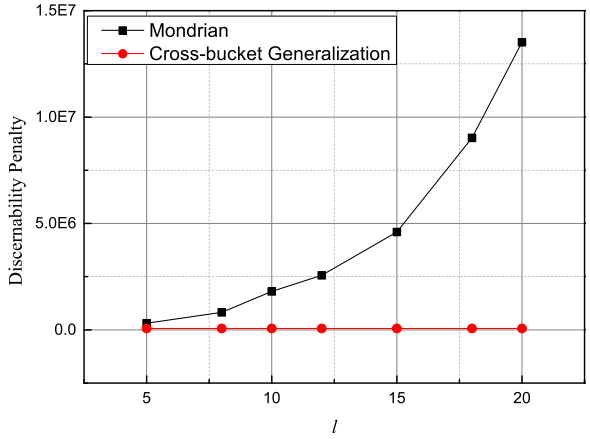

Fig. 6. Discernibility metric results.

magnitude of the difference between Mondrian and cross-bucket generalization is pronounced. Thus, the sizes of equivalence groups in the cross-bucket generalized tables are smaller than those in the anonymized tables by Mondrian. In addition, the values of cross-bucket generalization are the same in Fig. 6, which means the sizes of equivalence groups are hardly affected by l. Results indicate that the identity protection of cross-bucket generalization is barely affected by improving attribute protection, which avoids the overprotection shown in Fig. 3.

Next, we use the approach of aggregate query answering [5] to check information utility. We randomly generate 1,000 queries and calculate the average relative error for each anonymized table. The sequence of the queries is expressed as the form

SELECT SUM(salary) FROM **Microdata**
WHERE $pred(A_1^{QI})$ **AND** $pred(A_2^{QI})$ **AND** $pred(A_3^{QI})$
AND $pred(A_4^{QI})$.

Specifically, the query condition contains four random QI attributes, and the sum of the salaries is the result for comparison. For the categorical QI attributes, the predicate $pred(A^{QI})$ has the following form

$$(A^{QI} = v_1 \ or \ A^{QI} = v_2 \ or \ \ldots \ or \ A^{QI} = v_m),$$

where $v_i (1 \leq i \leq m)$ is a random value from $D[A^{QI}]$, whereas for the numerical QI attributes, the predicate $pred(A^{QI})$ has the following form:

$$(A^{QI} > v) \ or \ (A^{QI} < v) \ or \ (A^{QI} = v)$$
$$or \ (A^{QI} \geq v) \ or \ (A^{QI} \leq v) \ or \ (A^{QI} \neq v),$$

where v is a random value from $D[A^{QI}]$.

The relative error of a query equals $(Sum_{upper} - Sum_{lower})/Sum_{act}$, where Sum_{upper} and Sum_{lower} are the upper and lower bounds of the sum of the salaries, respectively, and Sum_{act} is the actual sum of the salaries. Fig. 7 presents the experiment results of the aggregate query answering.

We observe that the performance of cross-bucket generalization is more effective than Mondrian. The relative error rate of cross-bucket generalization is only a little affected by l compared with Mondrian. Based on the above

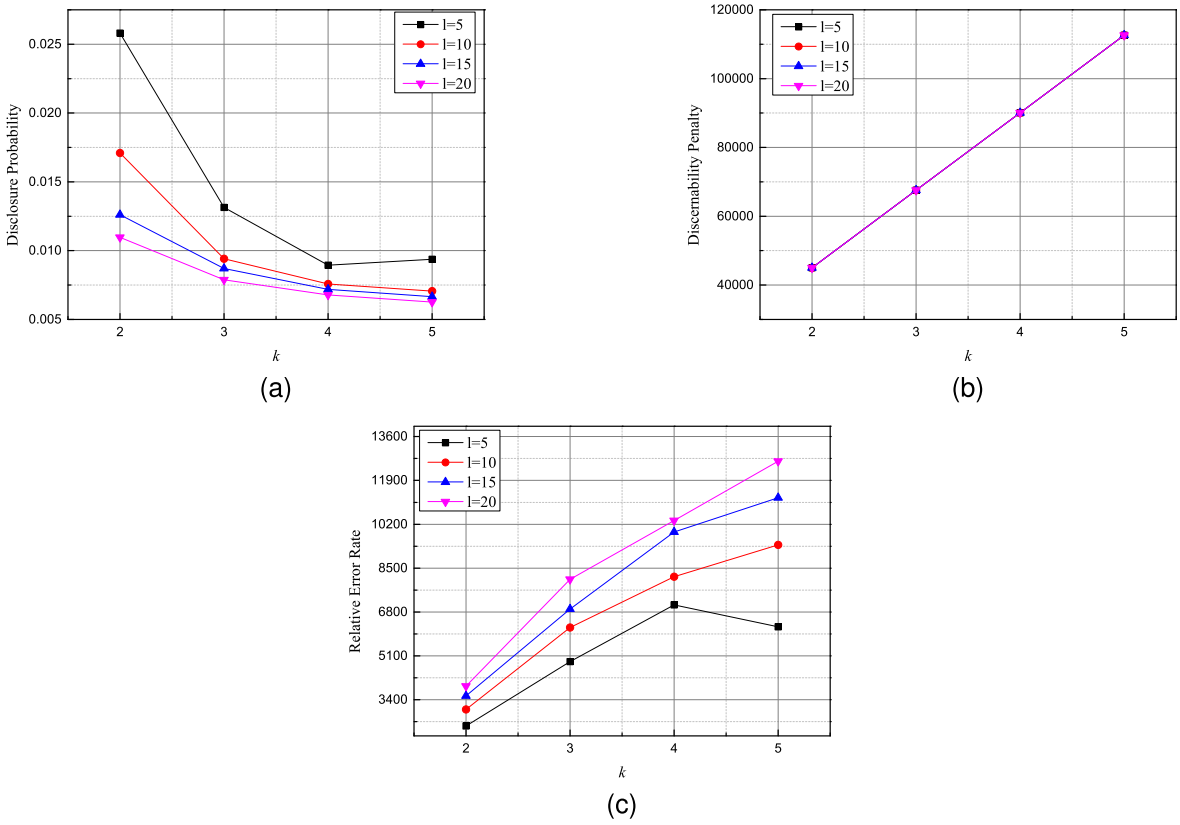

Fig. 8. Varied k values.

experiments, when identity protection is necessary (which is the case for most scenarios), and the requirement of attribute protection is higher than that of identity protection, cross-bucket generalization is the better choice than Mondrian for protecting privacy and saving information.

4.3 Effect of k

The above discussed experiments showed results when k is fixed at 3. In this next experiment, we evaluate the effect of k on attribute protection and information utility. We fix l at 5, 10, 15, and 20, respectively, and use the same configuration as previous experiments. The results are shown in Fig. 8.

Results from Fig. 8a show that the increase of k effectively improves the attribute protection when the disclosure probability is greater than about 0.01. The lines in Fig. 8b completely coincide, which means the sizes of equivalence groups are almost controlled by k. Fig. 8c illustrates that the relative error rate of aggregate query answering can be more affected by the increase of k when the value of l is larger.

5 Related Studies

Other than the protection for confidential information in different scenarios, major research on privacy-preserving data publishing focuses on the trade-off between privacy protection and information preservation [16], [22]. Even if the privacy of individuals can be perfectly protected, the published table is useless when it cannot be used for analysis and research. The generalization approach has been well-studied by proposing numerous algorithms, dividing it into three schemes: (1) global recoding, (2) local recoding, and (3) multidimensional recoding. Global recoding [20] replaces the same values in the original table by the same generalized value, whereas local recoding [21] allows the same values in the original table to be transformed into different generalized values. In the multidimensional recoding [3], tuples are partitioned into equivalence groups by their domain values, and either of two equivalence groups are non-intersected. However, previous generalization algorithms do not separate the protection against identity disclosure and attribute disclosure. Thus, these algorithms face the dilemma of always causing overprotection for identity when addressing a high requirement of attribute protection.

Certain anonymization approaches, such as bucketization, prevent attribute disclosure and preserve all the original QI values. Most of these approaches cannot prevent identity disclosure, and the QI values of individuals can be easily leaked to the adversary. The existence of individuals in the published table is likely to be exposed as a privacy violation [29]. On the other hand, the acquired QI values increase the background information that can be obtained by the adversary that can help the adversary to attack other data tables [30]. Therefore, the generalization procedure is always necessary for anonymizing microdata.

In addition, De Capitani di Vimercati et al. propose a potential fragmentation technique [28] based on constraints even the assumption is different from this paper. It supports the specification of confidentiality constraints, generically capturing privacy needs as sensitive attributes, or sensitive associations among them, which need to be protected. Their goal is to protect sensitive associations as defined in the confidentiality constraints by trying to integrate k-anonymity and l-diversity which is similar to this paper to some extent.

The recent closest study is the generalized bucketization scheme [17] that attempted to reduce information loss by minimizing the sizes of buckets. Wang et al. studied that each sensitive value has its own privacy setting, and buckets of different sizes can be formed. The goal is to find the optimal bucket setting, such that the minimum bucket size for each bucket that is necessary to provide the specified privacy settings for all the sensitive values is within the bucket. However, its serious drawback is that cannot prevent identity disclosure. Re-identification attacks can occur. By contrast, the proposed approach in this present study also protects the identity of individuals, thus reducing the risk of confidential information disclosure.

Differential privacy [11], [13], [14], as proposed by Dwork, is a powerful technique for providing privacy guarantees and aggregate data analysis, and has recently received attention. Differing from other anonymization approaches, this technique does not release the data points but answers statistical queries, injecting random noise to prevent the table linkage of individuals. Similarly, differential privacy also faces the contradiction between privacy protection and data analysis [24], [25]. For example, when the parameter ϵ of ϵ-differential privacy is large, the approach provides good information utility, but it is likely to cause privacy disclosure. A small leads to strong privacy protection but a poor data utility.

6 Conclusion and Future Studies

This study presents a novel approach called cross-bucket generalization that is useful for privacy-preserving data publication. The technique meets the requirements by providing a higher attribute protection than identity protection. The concept is to separate the protection against identity disclosure and attribute disclosure and avoid the overprotection for identity, given that the requirements of the protection for identity and sensitive values can be different.

In particular, the (k, l)-anonymous cross-bucket generalization algorithm is also proposed, combining the generalization and bucketization approaches. Once the data is masked, the possibilities of identity disclosure and attribute disclosure are limited under $1/k$ and $1/l$, respectively. Moreover, the sizes of equivalence groups and buckets are minimized to only satisfy the protection requirements for saving information utility.

The framework of cross-bucket generalization can be applied to other scenarios for protecting confidential information, such as multiple release publication [26], continuous data publication [27], and personalized privacy preservation [15]. These issues are recommended to be studied in accordance with the cross-bucket generalization technique.

Acknowledgments

The authors would like to thank the anonymous referees for their many valuable suggestions and comments. This work was supported in part by the National Nature Science Foundation of China under Grant No. 61373123, the National Key Research and Development Program of China No. 2017YFB0102500, and the Science and Technology Development Foundation of Jilin Province No. 20150414004GH, 20160204041GX, 20170101133JC.

References

[1] L. Sweeney, "k-anonymity: A model for protecting privacy," *Int. J. Uncertainty Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, 2002.

[2] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," *ACM Comput. Surveys*, vol. 42, 2010, Art. no. 14.

[3] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Mondrian multidimensional k-anonymity," in *Proc. IEEE Int. Conf. Data Eng.*, 2006, pp. 25–25.

[4] A. Machanavajjhala, J. Gehrke, and D. Kifer, "L-diversity: Privacy beyond k-anonymity," in *Proc. IEEE Int. Conf. Data Eng.*, 2006, pp. 24–24.

[5] Q. Zhang, N. Koudas, D. Srivastava, and T. Yu, "Aggregate query answering on anonymized tables," in *Proc. Int. Conf. Data Eng.*, 2007, pp. 116–125.

[6] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *Int. J. Uncertainty Fuzziness Knowl.-Based Syst.*, vol. 10, no. 05, pp. 571–588, 2002.

[7] X. Xiao and Y. Tao, "Anatomy: Simple and effective privacy preservation," in *Proc. 32nd Int. Conf. Very Large Data Bases*, 2006, pp. 139–150.

[8] T. Li, N. Li, J. Zhang, and I. Molloy, "Slicing: A new approach for privacy preserving data publishing," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 3, pp. 561–574, Mar. 2012.

[9] S. Ruggles, K. Genadek, R. Goeken, J. Grover, and M. Sobek, Integrated Public Use Microdata Series, Univ. Minnesota, 2015. [Online]. Available: https://usa.ipums.org/usa/index.shtml

[10] K. Wang, P. S. Yu, and S. Chakraborty, "Bottom-up generalization: A data mining solution to privacy protection," in *Proc. IEEE Int. Conf. Data Mining*, Nov. 2004, pp. 249–256.

[11] C. Dwork, "Differential privacy," in *Proc. Int. Conf. Automata Languages Program.*, 2006, pp. 1–12.

[12] B. C. M. Fung, K. Wang, and P. S. Yu, "Top-down specialization for information and privacy preservation," in *Proc. IEEE Int. Conf. Data Eng.*, Apr. 2005, pp. 205–216.

[13] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. Theory Cryptography Conf.*, 2006, pp. 265–284.

[14] X. Xiao, G. Wang, and J. Gehrke, "Differential privacy via wavelet transforms," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 8, pp. 1200–1214, Aug. 2011.

[15] X. Xiao and Y. Tao, "Personalized privacy preservation," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2006, pp. 229–240.

[16] T. Li and N. Li, "On the tradeoff between privacy and utility in data publishing," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 517–526.

[17] K. Wang, P. Wang, A. W. Fu, and R. C. W. Wong, "Generalized bucketization scheme for flexible privacy settings," *Inf. Sci.*, vol. 348, pp. 377–393, 2016.

[18] X. Xiao and Y. Tao, "M-invariance: Towards privacy preserving re-publication of dynamic datasets," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2007, pp. 689–700.

[19] R. Bayardo and R. Agrawal, "Data privacy through optimal k-anonymization," in *Proc. IEEE Int. Conf. Data Eng.*, 2005, pp. 217–228.

[20] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Incognito: Efficient full-domain k-anonymity," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2005, pp. 49–60.

[21] J. Xu, W. Wang, J. Pei, X. Wang, and B. Shi, "Utility-based anonymization using local recoding," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 785–790.

[22] C. Aggarwal, "On k-anonymity and the curse of dimensionality," in *Proc. 31st Int. Conf. Very Large Data Bases*, 2005, pp. 901–909.

[23] B. Chen, D. Kifer, K. LeFevre, and A. Machanavajjhala, "Privacy-preserving data publishing," *Found. Trends Databases*, vol. 2, no. 1/2, pp. 1–167, 2009.

[24] K. Wang, C. Han, A. W. C. Fu, R. C. W. Wong, and S. Y. Philip, "Reconstruction privacy: Enabling statistical learning," in *Proc. Int. Conf. Extending Database Technol.*, 2015, pp. 469–480.

[25] C. Han and K. Wang, "Sensitive disclosures under differential privacy guarantees," in *Proc. IEEE Int. Congr. Big Data*, 2015, pp. 110–117.

[26] C. Yao, X. S. Wang, and S. Jajodia, "Checking for k-anonymity violation by views," in *Proc. 31st Int. Conf. Very Large Data Bases*, 2005, pp. 910–921.

[27] J. W. Byun, Y. Sohn, E. Bertino, and N. Li, "Secure anonymization for incremental datasets," in *Proc. Workshop Secure Data Manage.*, 2006, pp. 48–63.
[28] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "Fragments and loose associations: Respecting privacy in data publishing," *Proc. VLDB Endowment*, vol. 3, no. 1/2, pp. 1370–1381, 2010.
[29] M. E. Nergiz, M. Atzori, and C. Clifton, "Hiding the presence of individuals from shared databases," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2007, pp. 665–676.
[30] K. Wang and B. Fung, "Anonymizing sequential releases," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 414–423.

Boyu Li received the MS degree from Jilin University, in 2014. He is currently working toward the PhD degree in the College of Computer Science and Technology, Jilin University. His research interests include the technique of privacy-preserving data publishing and machine learning.

Yanheng Liu received the MSc and PhD degrees in computer science from Jilin University, People's Republic of China. He is currently a professor at Jilin University. His primary research interests include network security, network management, mobile computing network theory and applications, etc. He has co-authored more than 200 research publications in peer reviewed journals and international conference proceedings of which one has won "best paper" awards. Prior to joining Jilin University, he was a visiting scholar with the University of Hull, England, University of British Columbia, Canada, and Alberta University, Canada.

Xu Han received the BE degree in software engineering from Jilin University, People's Republic of China, in 2015. She is working toward the master's degree in the College of Software, Jilin University. Her research interests include privacy protection in ad hoc wireless and vehicular networks.

Jindong Zhang received the PhD degree in computer science and technology from Jilin University, in 2009. His research interests mainly cover intelligent control and automotive electronic technology. He is on the faculty of Jilin University as a teacher of computer science and technology, Key Laboratory of Symbol Computation and Knowledge Engineering of the Ministry of Education, and State Key Laboratory of Automobile Simulation and Control, Jilin University.