A k-Anonymity-Based Robust Watermarking Scheme for Relational Database

Jing Yu[1,2], Shuguang Yuan[1,2], Yulin Yuan[1,2], Yafan Li[1,2], and Chi Chen[1,2(✉)]

[1] State Key Laboratory of Information Security,
Institute of Information Engineering, Chinese Academy of Science,
Beijing 100093, China
{yujing,yuanshuguang,yuanyulin,liyafan,chenchi}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences,
Beijing 101400, China

Abstract. In the era of big data, secure and controlled data publishing becomes increasingly vital. When data holders publish dataset to data demanders, data holders often (1) protect the copyright of the published dataset and (2) anonymize user's data by k-anonymity for privacy purpose. Hence, there is a realistic demand of watermarked k-anonymity dataset for ownership. However, there are two important challenges to be addressed: the lack of primary key and the narrow bandwidth channel for watermarked k-anonymity dataset. In this paper, we try to address above challenges by proposing a k-anonymity-based robust watermarking scheme in anonymized dataset by an "one-time" way to achieve both protection of privacy and copyright. This scheme is primary key independent and meets the requirement of keeping the same form with k-anonymity. Experimental studies prove the robustness of watermarking scheme against subset deletion and subset addition attacks.

Keywords: Database watermarking · Ownership · k-anonymity

1 Introduction

In the era of big data, with the development of technologies such as data mining and information sharing, data often need to be published to other organizations for use, analysis and research. Consider a scenario that merchants (called "data holders") publish dataset to buyers (called "data demanders"). Some dishonest data demanders (called "traitors") may collect, share or sell published dataset for profits without permission from the data holders. The data holders may use a watermarking scheme to embed a specific mark into their dataset for asserting ownership. In relational database, many robust watermarking schemes [5,6,19,20,23,26] for ownership have been proposed.

For privacy purpose, privacy preserving data publishing schemes including k-anonymity [16,22] have developed. Data holders could anonymize user's data

C. Su et al. (Eds.): SciSec 2022, LNCS 13580, pp. 557–573, 2022.
https://doi.org/10.1007/978-3-031-17551-0_36

before publishing dataset. Hence, there is a realistic demand of copyright identification of leaked anonymized data in that contained sensitive information such as medical health data. To the best of our knowledge, some works [13,17,18] aim to fingerprint anonymized dataset. By applying different anonymization patterns, they achieved traitor-tracing. And the [8] implements sanitization and fingerprinting by adding and removing tuples from the anonymized dataset. These works achieved the goal of traitor-tracing. But the goal of identification of dataset copyright can't reach due to the lack of identifiable watermarks. However, we hope to protect privacy while asserting data ownership through k-anonymity and watermarking techniques.

However, there are two important challenges of watermarking in k-anonymity dataset need be addressed. The first challenge is that the lack of primary key. In k-anonymity scenario, the primary key is defined as explicit identifiers which must be deleted. For most watermarking schemes [1,2,5,10,26], the primary key is used to locate watermarks. Hence, types of classical watermarking schemes can't adapt k-anonymity dataset. The second challenge is that narrow bandwidth channel. The structure of k-anonymity is classified into four types: explicit identifiers (EIs), quasi-identifiers (QIs), sensitive attributes (SAs) and non-sensitive attributes (NSAs). The EIs will be deleted. And any modifications in QIs and SAs are intolerant for privacy objective of k-anonymity. NSAs are often considered as data that are not important. The fact is that NSAs are more likely to be destroyed with a higher priority. Hence, the places and bandwidth of watermark are limited in k-anonymity dataset.

1.1 Contribution and Paper Organization

To address above challenges, we analyze the existing three types of watermarking strategies in k-anonymity in detail in Sect. 2: Watermark then Anonymize (WA), Anonymize then Watermark (AW) and Integrated Strategy, and propose a k-anonymity-based watermarking scheme for relational database. The main contributions of our work include:

1. We discuss and analyze the operable watermarking possibilities in anonymization scenario, and try to find available bandwidth in QIs for watermarking.
2. We propose a novel watermarking algorithm by an "one-time" way to achieve both protection of privacy and copyright. It's primary key independent and meets the requirement keeping the same form with k-anonymity.
3. Experimental studies prove the robustness of the watermarking algorithm against subset deletion and subset addition attacks.

Organization: Sect. 2 introduces background and motivation. Section 3 presents the proposed watermark scheme. In Sect. 4, data-driven experiments are demonstrated. Section 5 demonstrates related works. Section 6 concludes this paper.

2 Background and Motivation

In this section, we first introduce the background knowledge related to anonymization, then describe our motivation for studying k-anonymity-based watermarking scheme.

2.1 Background

Researchers have proposed many studies targeting anonymized data. Most of the current consider that the data to be anonymized in the form of explicit identifiers (EIs), quasi-identifiers (QIs), sensitive attributes (SAs) and non-sensitive attributes (NSAs), where EIs are attributes that explicitly identify tuple owners (e.g. name and ID), QIs are attributes that could be linked to external tables to identify the tuple owner (e.g. age and zipcode), SAs include sensitive information about individuals such as illness, salary, etc., and NSAs contain all attributes except for the previous three types. In fact, NSAs are often considered as data that are not important for research. Most works assume that the four sets of attributes are disjoint. To protect individuals privacy, EIs will be removed before the data is published. Thus the data holders publish an anonymized dataset, including QIs′, SAs and NSAs, among which QIs′ are the results of QIs being anonymized.

The k-anonymity model is one of the most widely anonymization methods. K-anonymity was first proposed by *Sweeney* and *Samarati* [16,22]. The k-anonymity model requires that any tuple in an anonymized dataset is indistinguishable from other k-1 tuples in QIs′. The set of tuples in the anonymized dataset containing the same QI values are defined as equivalence classes (ECs). That is, the size of all ECs in the anonymized dataset after k-anonymity is not smaller than k.

Table 1. Example dataset

(a) An original table						(b) 3-anonymity table				
EI	QIs		SAs		NSA	QIs′		SAs		NSA
Name	Age	Zipcode	Disease	Salary	Score	Age	Zipcode	Disease	Salary	Score
Alex	24	53712	Heart disease	5000	98	[24,32]	[53712–53713]	Heart disease	5000	98
Beth	25	53711	Heart disease	6000	87	[25,30]	53711	Heart disease	6000	87
Carl	30	53711	Flu	10000	80	[25,30]	53711	Flu	10000	80
Ellen	30	53711	Cancer	5000	92	[25,30]	53711	Cancer	5000	92
Glen	32	53712	Heart disease	4000	79	[24,32]	[53712–53713]	Heart disease	4000	79
Helen	32	53713	Cancer	8000	96	[24,32]	[53712–53713]	Cancer	8000	96

For example, Table 1 (a) is an original dataset, where Name is an EI, <*Age*, *Zipcode*> are considered as QIs, Disease, Salary are considered as SAs, and Score is a NSA. Table 1 (b) shows the anonymized result obtained from Table 1 (a) after 3-anonymity. Tuples 2, 3 and 4 in Table 1 (b) form an equivalence class with respect to quasi-identifiers <*Age*, *Zipcode*>. Even if the data demanders know Beth's QI values, it is difficult to tell which of the three tuples he is.

The implementation of k-anonymity is divided into two methods depending on how the data is transformed: global recoding and local recoding. Global recording means that the same QIs values must be mapped to the same values or ranges in all tuples. Typical global recording algorithms include Incognito [14], Binary search [16] and Datafly [21]. Local recoding allows the same QIs values to be mapped to different values or ranges. Top-Down Specialization [7], Mondrian [15] and Bottom-Up Generalization [24] are typical local recoding algorithms.

2.2 Motivation

To protect privacy while achieving copyright protection, there are three solutions for data holders to use. We analyze the three solutions and thus illustrate the motivation of our research.

(1) Watermark then Anonymize (WA).

The meaning of WA is to embed watermark in the original dataset first and then anonymize the watermarked dataset. As we mentioned above, the original dataset usually consists of EIs, QIs, SAs and NSAs. However, since EIs can often correspond to unique individuals, they are usually removed when published. Therefore, in the following discussion, we only discuss scenarios where QIs, SAs and NASs are processed separately. In addition, due to the difficulty of embedding watermarks in categorical attributes, the attributes we discuss in the following are all numerical attributes.

- **Watermark on QIs.** First consider embedding watermark in QIs. We assume that the data holder embeds watermark in numerical QIs and then anonymizes the dataset. However, since the values of QIs are likely to change after k-anonymity, which may cause the watermark information to be rewritten, resulting in the watermark information not being detected properly.
- **Watermark on SAs.** We assume that the data holder embeds watermark in numerical SAs. Although SAs do not change after k-anonymity, and the anonymization process does not affect the watermark embedded in SAs. However, as important parameters for data usage and analysis, the accuracy of SAs after embedding watermarks can be affected.
- **Watermark on NSAs.** We consider the case of embedding watermark in numerical NSAs. Although watermark embedded in NSAs are similarly unaffected by the subsequent anonymization process, there are still some problems. Firstly, compared to other attributes, the number of NSAs is usually relatively small. This results in less space for watermarking information to be embedded. More importantly, since NSAs are often considered unimportant for research, they may be restricted from publishing, which results in the absence of watermark information in the published dataset.

(2) Anonymize then Watermark (AW).

AW means to anonymize the original dataset first and then embed watermark in the anonymized dataset. Since k-anonymity is only processed for QIs, we focus

on the case of embedding watermark in QIs′ in this section. And the cases of embedding watermark in SAs and NSAs are the same as the case in WA.

– **Watermark on QIs′.** We assume that the data holder performs k-anonymity on the original dataset first and then embeds watermark in QIs′. We know that for k-anonymity achieved by generalization or clustering methods, the value of QIs′ in the anonymized dataset is presented as a range, such as [24,32] in Table 1 (b). Therefore, watermark information cannot be embedded in QIs′ presented in the form of a range. Moreover, for k-anonymity achieved by microaggregation methods, although the values of QIs after anonymization are accurate values, the watermarks are still not suitable to be embedded in QIs′. This is because k-anonymity requires that the QI values in each equivalence class are the same, and the data holder must still ensure that the QI values in the equivalence classes are the same after embedding watermark. This will result in a significant reduction in the number of watermarks that can be embedded in anonymized dataset.

(3) Integrated Strategy.

Integrated strategy means that data holders combine the privacy protection and copyright protection into one integrated mechanism during dataset publishing. Only a few works belong to this type. The works [13,17,18] propose a fingerprinting method based on k-anonymity. The method distributes datasets with different generalization patterns to multiple data demanders, thus enabling tracing back to the traitors who caused the privacy disclosure. However, the method does not assert data ownership. In addition, the authors did not experimentally validate the method. The method proposed in [8] implements sanitization and fingerprinting by adding and removing tuples from the dataset. The method protects the privacy of individuals, while allowing traitors to be tracked in the event of illegal redistribution. However, the method also cannot assert data ownership. In addition, there is no experimental validation of the method in the article. The methods above combine privacy protection and fingerprinting, and are able to protect data privacy while tracing traitors. However, none of these methods can assert the ownership of the data.

3 Proposed Watermark Scheme

In this section, we introduce the k-anonymity-based watermarking scheme, including bandwidth channel, watermark architecture and algorithms.

3.1 Bandwidth Channel

According to above discussion, EIs, SAs attributes are excluded from watermarking. NSA attributes may be destroyed with a higher priority. Hence, we explore suitable places and bandwidth on QI attributes. This section we will discuss the bandwidth channel for watermarking in QIs attributes. Most watermark methods work under a general assumption that the original dataset can

tolerate a certain degree of quality degradation. The tolerance is closely related to the bandwidth for watermark. The anonymized dataset consists of many ECs, each of which is guaranteed to contain at least k tuples. And the optimal upper reference bound of an EC is $2k-1$, which has been strictly proved in reference [15]. Hence, there is a natural interval $[k, 2k-1]$ for the number of tuples contained in each EC. Thus we advocate that the each EC in an anonymized dataset can actually tolerance degree change of tuple number in the interval $[k, 2k-1]$, thereby providing the desired bandwidth channel for watermarking.

Example. Figure 1 shows a bandwidth channel for watermarking in a 3-anonymity table. We take the attribute $<Age>$ as QI to 3-anonymize and use a histogram to represent the 3-anonymity table. The abscissa represents the generalization interval of attribute $<Age>$, and the ordinate represents the tuple number in an EC, denoted by $|EC_i|$. Thus the bandwidth channel for watermarking in the 3-anonymity table is the interval of [3, 5].

Based on the bandwidth channel which is unique to anonymized dataset, we propose a new watermarking architecture, under which we can generate ECs in which the number of tuples are controlled. The controllable tuple numbers in these special ECs can be as watermarks.

Age	Zipcode	Salary
[21-25]	53711	5000
[21-25]	53711	4000
[21-25]	53711	7000
[21-25]	53711	3000
[26-30]	53712	6000
[26-30]	53712	7000
[26-30]	53712	5000
[31-35]	53713	4000
[31-35]	53713	6000
[31-35]	53713	8000
[31-35]	53713	7000
[31-35]	53713	5000

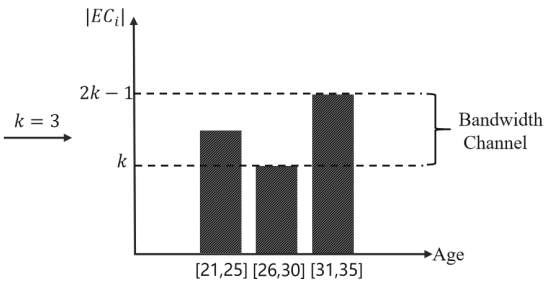

Fig. 1. The bandwidth channel for watermarking

3.2 Watermark Architecture and Algorithm

This section we will discuss the proposed watermark architecture of relational data that meet the dual requirements of privacy and copyright. The main architecture is presented in Fig. 2. The architecture includes the following three major phases: 1. Watermark Embedding, 2. k-anonymity and 3. Watermark Detection. Phase 1 and phase 2 belong to the data publishing stage, which must satisfy the k-anonymity specification in order to preserve the same data format. For ease of reference, we list notations that will be used in this paper in Table 2.

k-Anonymity Specification. The k-anonymity specification is used to regulate the watermark embedding function and the anonymity process, as in Fig. 2. It

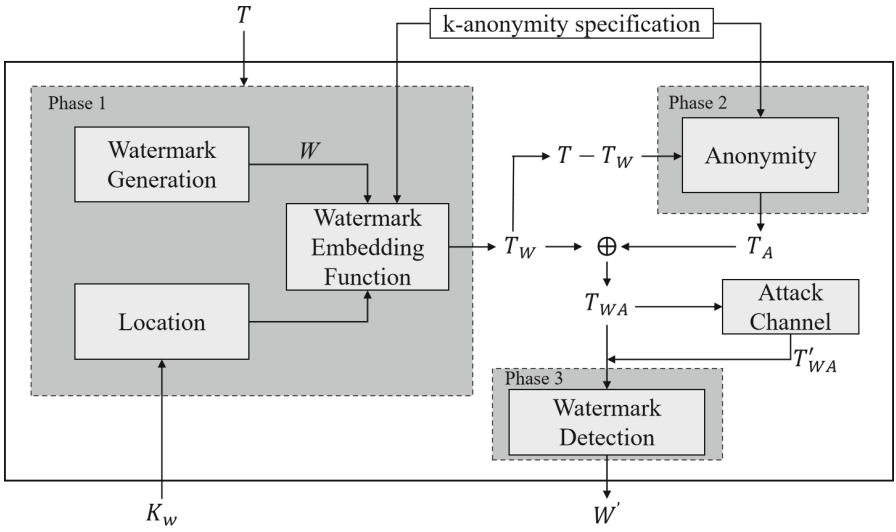

Fig. 2. Main architecture of the proposed watermarking scheme

requires that when executing the above two processes, the same set of QIs must be processed, while satisfying the anonymization parameter k.

Table 2. Notations

Notation	Description	Notation	Description
T	Original dataset	T_W	Watermarked dataset
T_A	Anonymized Dataset	T_{WA}, t_{WA}	Published Dataset and a tuple in T_{WA}
T'_{WA}	Published dataset that have been attacked	k	The system parameter for k-anonymity
W, W_g	Watermarks and the watermark in g-th EC	K_w	Watermark secret key
γ	A density control parameter	Δ	A redundant space for watermark
$QI_1, ..., QI_p$	QI attributes	s	A minimal step
t, N	A tuple and total number of tuples in T	n	Total number of watermarks
η	The proportion of successful watermark detection	τ	The watermark detection threshold

Phase1: Watermark Embedding. The main focus of watermark embedding phase is to embed watermark in such a way that it does not affect the privacy objective of k-anonymity. The original dataset T is transformed into watermarked dataset T_W. The original dataset T, watermark W, secret key K_w and k-anonymity specification are inputs. Watermark W and secret key K_w are only known to the data holders. And no one can detect the embedded watermark W without the secret key K_w.

Phase2: k-anonymity. In this phase, the main work is to anonymize the unmarked dataset $T - T_W$ to satisfy the k-anonymity specification. Thus, the published dataset can preserve the same data format with T_W and protect individual privacy. The output of this phase is the anonymized dataset T_A. By

aggregating the watermarked dataset T_W and anonymized dataset T_A, publishing dataset T_{WA} is generated.

Attack Channel. After watermarking, the dataset is released to the data demanders over a communication channel or attacker channel. The published dataset T_{WA} may undergo different types of attacks in the attacker channel. The T'_{WA} denotes dataset under attack. The attacker dose not know the real positions of watermarks in that lack of secret key. The attacker may conduct two forms of attacks: **Subset Deletion Attacks.** The attacker may delete subset of watermarked dataset T_{WA} to destroy the watermarks. **Subset Addition Attacks.** The attacker may insert a number of duplicate tuples or randomly generated fake tuples into the watermarked dataset T_{WA}. In this particular type of attack, the insertion of new tuples by the attacker did not harm the data quality and watermark information, but it may decrease the detection ratio of watermark.

Phase3: Watermark Detection. In the watermark detection phase, the watermark W' in the published dataset T_{WA} will be extracted using the secret key K_w.

3.2.1 Watermark Embedding Phase. In the watermark embedding phase, three important tasks are accomplished: watermark generation, watermark location and watermark embedding.

(1) Watermark Generation. The watermark generation is based on the bandwidth channel, which is the interval $[k, 2k-1]$ in ECs.

Definition 1. Watermark Based on Bandwidth. The watermark W is composed of several variable intervals W_g, which controls the tuple numbers of the marked ECs.

$$W = (W_1, W_2, ...W_g, ..., W_n), 0 \leq |W_g| \leq k-1, 1 \leq g \leq n \tag{1}$$

$$W_g = [a, b], 0 \leq a < b \leq k-1, |b-a| = \Delta \tag{2}$$

where n represents the number of watermarks, g represents the g-th marked EC. Based on the bandwidth channel, the $|EC_g| = k + W_g$ and $k \leq |EC_g| \leq 2k-1$. Thus $0 \leq |W_g| \leq k-1$. The a and b are the positive integer, and $|b-a|$ relative to k-1 is a relatively smaller positive integer Δ, which provides a redundant space for watermark.

(2) Watermark Location. The potential ECs are selected to embed the watermark W. For the anonymized dataset, each QI attribute of a tuple t in an EC has an interval $[QI^{min}, QI^{max}]$. Thus, locating $t.QI^{min}$ is the key to form a special ECs. We use an one-way hash function (e.g.SHA1 or MD5) H to locate $t.QI^{min}$. The location function is as follow:

$$H(K_w|t.QI) = H(K_w|t.QI_1|...|t.QI_p) \tag{3}$$

where K_w represents a secret key known only to the data holders, | represents concatenation. The $t.QI$ is the concatenation of the value for each QI attribute in tuple t. Then let H randomizes the values of K_w and the $t.QI$:

$$H(K_w|t.QI) \ mod \ \gamma = 0 \tag{4}$$

When return value of formula (4) is 0, the tuples having value $t.QI_1, ..., t.QI_p$ will be selected, which can be as the start point of interval. Then the end point needs to be determined, making the number of tuples in the interval [start point, end point] equal to $|W_g|$. The γ is a control parameter that determines the density of watermarks. Note that, for ease of understanding and illustration, we represent tuples as points in space.

(3) Watermark Embedding. When the start point $t.QI^{min}$ is determined, we need to determine the end point to form EC that meet k-anonymity and watermark W requirement. However, there are many tuple sets satisfy watermarking requirement. Then the problem is how to choose the best one from the tuple sets which has minimal information loss. The normalized certainty penalty (NCP) [25] measure the information loss of each QI attribute in ECs. After the start point is located, we rank priority to candidate QIs with NCP for finding an end point as $t.QI^{max}$ that satisfy watermark W. From start point to end point of interval, tuples are selected to be grouped, which satisfies watermark W and has minimal information loss. Next, we generalize the selected tuple sets to form marked ECs. Finally, we take out the marked ECs to form an anonymized dataset with watermark sequence W.

$$T_W = EC_{W_1} \cup EC_{W_2} \cup ... \cup EC_{W_n} \tag{5}$$

Example. Figure 3 showns the procedure of watermark embedding, which contains three basic steps: Select the start points, Select tuple set and Generalization. Consider that dataset T has QIs $(QI_1, ..., QI_p)$, and assume that there is a order for each QI dimension. The tuples of T on $QI_1, ..., QI_p$ can then be represented as points in p-dimensional space. The Fig. 3 (a) shows a two-dimensional representation of a dataset, which Age and Weight are QI attributes. Each white dot represent a tuple. The overlapping points represent that these tuples have the same age-weight values.

Step 1: Select the Start Point. First, we should search the start point according to formula (3), (4). Take Fig. 3 (b) as a example, K_w is the watermark key. When $Age = a_2$, $Weight = w_2$, the value of formula (4) is 0. These two tuples are selected and one of them is used for subsequent calculations. This point is marked as black.

Step 2: Select Tuple Set. Next, we select the tuple set that satisfy k-anonymity and watermarking requirements. As shown in Fig. 3 (c), the anonymity parameter $k = 3$, the watermark $W_g = [a,b](a = 0, b = 1)$. We choose $W_g = 1$, and two tuples has been selected in the step1. There are two alternative ways to select tuple set as shown by the dotted line in the figure. In order to select the better tuple set

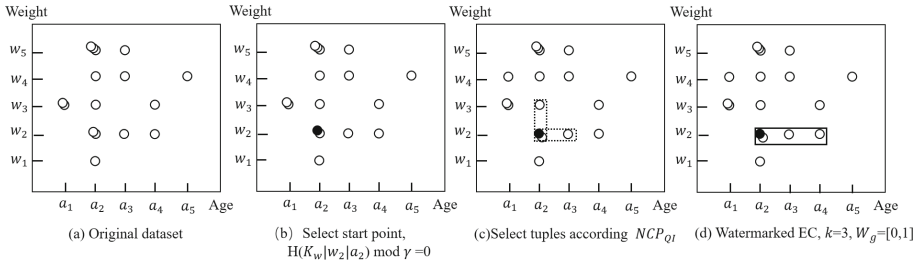

Fig. 3. Spatial representation of Age-Weight and selections based on k and W

which has less information loss, we calculate $NCP_{QI_{age}}$ and $NCP_{QI_{weight}}$ on these selected tuples in generalized data format. After calculation, the $NCP_{QI_{age}}$ is minimum. Thus, we select tuples according the Age attribute, and the result is shown in Fig. 3 (d). Note that, this step may repeat many times until the tuple number in the EC is greater than or equal to $k + W_g$.

Step 3: Generalization. Finally, we generalize these tuple to form marked ECs. As shown in Fig. 3 (d), we select the tuples including (a_2, w_2), (a_2, w_2), (a_3, w_2), (a_4, w_2). After generalization, the four tuples will be replaced by range $([a_2, a_4], w_2)$. These four tuples generate an marked EC.

3.2.2 *k*-anonymity Phase. In the anonymity phase, the most important work is to anonymize the unmarked dataset $T - T_W$ to satisfy the k-anonymity specification. The k-anonymity algorithms using local recoding (LR) can be used to process unmarked dataset in our scheme. The generalization intervals generated by GR do not overlap, while the generalization intervals generated by the LR might potentially overlap. In other words, the LR allows the same QI values to be generalized into different intervals. If the GR is used in the anonymity phase, strict non-overlapping generalization intervals will be generated. It is easier for an attacker to identify the marked ECs. The LR will produce overlapping generalization intervals, thus, the attacker can not easily identify the marked ECs. Therefore, only the LR is suitable for our scheme. In anonymity phase, the processing of the unmarked dataset is as follows:

$$T_A = k - anonymity_{LR}(T - T_W) \tag{6}$$

where T_A is the anonymized dataset of $T - T_W$, the $k - anonymity_{LR}$ is the k-anonymity algorithms using local recoding. After phase 1 and phase 2, the published dataset T_{WA} is made (including marked EC_{W_g} and unmarked ECs), which adequately protects both individual privacy and data copyright (Fig. 4).

The basic algorithm is given in Algorithm 1, which describes the watermark embedding phase and the k-anonymity phase. Lines 2–8 determine all the starting points that will be marked when embedding the watermarks. Lines 10–25 implement the embedding of the watermarks. First, add one of the selected starting points to the *group* (lines 11–13). If the size of *group* is within the range

Algorithm 1 Watermark Embedding

Input: T,K_w,k, W,γ
Output: T_{WA}

1: $Node \leftarrow [\,]$, $T_W \leftarrow [\,]$
2: **for** each t in T **do**
3: **if** H($K_w|t.QI_1|...|t.QI_p$) mod γ =0 **then**
4: $node \leftarrow$ *tuples have same QIs value with* t
5: $Node \leftarrow Node \cup node$
6: $T \leftarrow T - node$
7: **end if**
8: **end for**
9: $i \leftarrow 1$
10: **for** each $node$ in $Node$ **do**
11: $t \in node$
12: $group \leftarrow$ *tuples in the space*$(node)$
13: $number \leftarrow Count(group)$
14: **if** $number = k + W_i$ **then**
15: $EC_{node} \leftarrow Generalize(group)$
16: $T_W = T_W \cup EC_{node}$
17: **else if** $number < k + W_i$ **then**
18: $j = \arg\min_p NCP(t.QI_1 + s), ..., NCP(t.QI_p + s)$
19: $node \leftarrow updateNode(node, QI_j, s)$
20: $i = (i + 1)\ mod\ len(W)$
21: jump to Line 11
22: **else**
23: continue
24: **end if**
25: **end for**
26: $T_A \leftarrow\ k - anonymity(T - T_W)$
27: $T_{WA} \leftarrow T_W\ \cup\ T_A$
28: **return** T_{WA}

Algorithm 2 Watermark Detection

Input: T_{WA},K_w,W,k,γ
Output: *Detection accuracy*

1: $Node \leftarrow [\,]$, $W' \leftarrow [\,]$
2: **for** each EC_{WA} in T_{WA} **do**
3: $t_{WA} \leftarrow$ *a tuple in* EC_{WA}
4: **if** (H($K_w|t_{WA}.QI_1^{min}|...|t_{WA}.QI_p^{min}$) mod γ =0) **then**
5: $Node \leftarrow Node \cup EC_{WA}$ *having QI value* ($K_w|t_{WA}.QI_1^{min}|...|t_{WA}.QI_p^{min}$)
6: **end if**
7: **end for**
8: **for** each EC_{WA} in $Node$ **do**
9: $number \leftarrow Count(tuples\ in\ EC_{WA})$
10: $W'.add(EC_{WA}, number - k)$
11: **end for**
12: **return** $\frac{match(W,W')}{|W|} * 100\%$

Fig. 4. Watermark embedding and detection algorithm

$k + W_i$, the *group* is generalized and added to T_W (lines 14–16). If the size of *group* is less than the minimum of this range, we uses NCP to expand the number of selected tuples on QI attributes (lines 17–21). Otherwise, these points are not processed (lines 22–24). Finally, *k-anonymity* on T-T_W does remaining work. The upper bound on the time complexity of the watermark embedding algorithm is $O(N * 2k)$.

3.2.3 Watermark Detection Phase. In the watermark detection process, the first step is to locate the start points in the marked ECs using the secret parameters K_w and γ according to formula (3), (4). The next step is to calculate the tuple numbers in the marked ECs where the start points are located. The tuple numbers in each marked ECs minus the parameter k is the detected watermark W'. The last step is to compare W and W'. In our scheme, we can judge whether the watermark detection is successful according to formula (7), (8). First, we judge whether the watermark detection is successful in a marked EC according to formula (7). If the ${W_g}' \in [a, b]$, the watermark detection is successful. Then, we calculate the proportion η of ECs with successful watermark detection in the all marked ECs according to formula (8). We set the detection threshold τ, if $\eta \geq \tau$, the watermark detection for the T'_{WA} is successful.

$$f(EC_{W_g{}'}) = \begin{cases} 1, a \leq {W_g}' \leq b, 0 \leq g \leq n; \\ 0, otherwise. \end{cases} \tag{7}$$

$$\eta = \frac{\sum_{g=0}^{n} f(EC_{W_g{}'})}{n} * 100\% \tag{8}$$

The watermark detection algorithm is shown in Algorithm 2. The purpose of lines 2–7 is to find all equivalent classes EC_{WA} that contain watermark information. Then, the algorithm calculates the size of these EC_{WA} and records them in W' (lines 8–11). Finally, the detected W' is compared to W. The watermark detection is successful if the match proportion η is more than detection threshold τ. The time complexity of the watermark detection algorithm is $O(|EC_{WA}|)$. $|EC_{WA}|$ is the number of equivalence classes in T_{WA}.

4 Experimental Studies

In this section, we present the experimental studies of our practical algorithms in terms of watermarking robustness, data utility and efficiency. Our algorithms are k-anonymity-based robust watermarking algorithms, thus we named them k-RWA in short. In the later experiments, we use k-RWA to represent our proposed algorithms.

4.1 Experimental Setup

Our experimental setup includes the Dataset, Experimental Environment and Experimental Parameters.

Dataset. We evaluated our proposed scheme in the publicly available dataset INFORMS[1]. The dataset includes 102578 records and 18 attributes. And 9 numerical attributes in the INFORMS dataset were used in our experiment as QIs.

Experimental Environment. The experiments were conducted on a machine equipped with a 3.0 GHz Intel(R) Core(TM) i5 processor with 16 GB RAM. The operating system on the machine was Microsoft Windows 10. The programming language we used is Python in version 3.7.3.

Experimental Parameters. In our experiments, we choose the local recording algorithms Mondrian [15] and Top-Down Specialization [7] as k-anonymity algorithms in k-RWA. We use $k = 20$, QI = 3, $|W_g| = [0,10]$, $\gamma = 25$. Our experiments were repeated 10 times and the average of the results was calculated as the final result for each trial.

4.2 Robustness

In this section, we did experiments on the robustness of k-RWA with two types of attacks, "Subset Deletion" and "Subset Addition".

Subset Deletion. We randomly delete subset of tuples from published dataset T_{WA} with the ratio from 10% to 90%. The detection threshold is $\tau = 50\%$ with

[1] https://sites.google.com/site/informsdataminingcontest/.

red dotted line. In Fig. 5(a), experiments show the result of k-RWA (Mondrian) and k-RWA (Top-down), and even when the ratio of deleted tuples reach 90%, the watermark detection is successful.

Subset Addition. We add new tuples in the published dataset T_{WA} with the ratio from 20% to 200%. The source of new tuples for addition is from original dataset. The detection threshold is $\tau = 50\%$ too. Figure 5 (b) shows the experiment results of k-RWA (Mondrian) and k-RWA (Top-down) by adding tuples with the different ratios, and even when the ratio of addition tuples reach 200%, the watermark detection is successful.

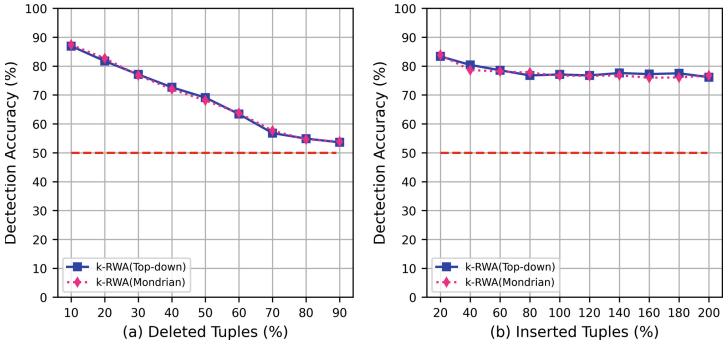

Fig. 5. The resilience to attack for k-RWA

Through the experiments above, k-RWA has good robustness against subset deletion and subset addition attacks.

4.3 Utility Evaluation

In this section, we report the experimental results of the k-RWA in terms of data utility. If the original dataset is processed only by Mondrian or Top-down, it will bring information loss to the original dataset, we called it I_k. If the original dataset is processed by k-RWA(Mondrian) or k-RWA(Top-down), it will also bring information loss to the original dataset, we called it I_{k-RWA}. In this experiments, we compare I_{k-RWA} with I_k using the GCP metric [9]. First, we compare I_{k-RWA} with I_k with different k and QI. We vary k from 20 to 100, and vary QI from 2 to 9. The results as shown in Fig. 6 (a) and (b) indicate that the information loss of k-RWA is higher than Mondrian or Top-down in most cases. This is due to the fact that the equivalence class chosen by k-RWA in order to embed the watermark may not be optimal for anonymization. The trade-off between privacy objective and watermarking objective, the latter must be achieved with high priority, which may causes extra information loss.

Then we analyze the impact of parameter γ on data utility. We vary γ from 25 to 200. The result as shown in Fig. 6 (c) demonstrates that GCP decreases

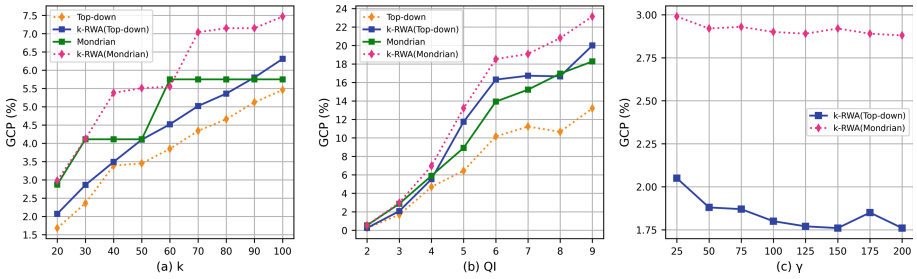

Fig. 6. Comparison of algorithms for the GCP with different value of k,QI and γ

gradually with increased γ for k-RWA(Top-down) and k-RWA(Mondrian), which proved that the less ECs selected for embedding watermark, the less information loss. Therefore, for data holders, appropriate parameters γ should be selected according to the degree of tolerance for information loss.

4.4 Efficiency

We also measured the execution time of the k-RWA embedding algorithm and the k-RWA detection algorithm under different dataset size and γ value.

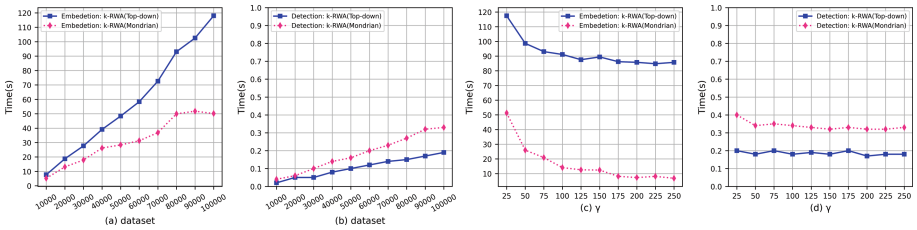

Fig. 7. Computation time for k-RWA

In Fig. 7 (a) and (b), we vary the dataset from 10000 tuples to 100000 tuples. The results show that the execution time of embedding and detection algorithms increases along with the size of embedded dataset. In Fig. 7 (c) and (d), we vary parameter γ from 25 to 250. The results show that the execution time of embedding and detection algorithms decreases along with the value increases of γ. Besides, the detection execution time shown in Fig. 7 (b) and (d) is much lower than the embedding execution time. The reason is that the information loss of each selected QI needs to be calculated when embedding watermarks, which has time consumption.

5 Related Work

Copyright protection is one of the most important issues in relational database for data holders. In [12], the database watermarking techniques are classified into three categories: Bit-Resetting Techniques (BRT), Data Statistics-Modifying Techniques (DSMT) and Constrained Data Content-Modifying Techniques (CDCMT). For BRT, selected bits are reset by a systematic process. Agrawal *et al.* [1,2] published the first relational databases watermarking scheme, which utilizes bits as watermarks. Following their research, a lot of watermarking models are proposed, such as [5,10,23,26]. For DSMT, data statistics such as mean, variance or distribution are used as watermarks. In [20], Sion *et al.* proposed a method that encoding of the watermark bit relies on altering the size of the "positive violators" set. Shehab *et al.* [19] formulated the database watermarking schemes as a constrained optimization problem. CDCMT schemes are based on modifying the contents of the data. For example, the schemes based on the ordering of the tuples (such as [4]) and insertion of extra spaces in attribute values (such as [3]). Such that watermarked data still remains useful. The zero-watermarking schemes [11] is also under this category. Our k-anonymity-based watermarking scheme belongs to the CDCMT.

6 Conclusion

In the current situation, data holders should not only protect data copyright, but also protect the privacy of users. To achieve dual protection goals, we proposed a database watermarking scheme based on k-anonymity. In our scheme, we propose an efficient watermark embedding algorithm and a watermark detection algorithm. Experimental results show that robustness, utility and efficiency of our scheme are good.

Acknowledgment. This work was supported by National Science and Technology Major Project of China under the Grant No. 2016ZX05047003.

References

1. Agrawal, R., Haas, P.J., Kiernan, J.: Watermarking relational data: framework, algorithms and analysis. VLDB J. **12**(2), 157–169 (2003)
2. Agrawal, R., Kiernan, J.: Watermarking relational databases. In: VLDB 2002: Proceedings of the 28th International Conference on Very Large Databases, pp. 155–166. Elsevier (2002)
3. Al-Haj, A., Odeh, A.: Robust and blind watermarking of relational database systems (2008)
4. Bhattacharya, S., Cortesi, A.: A distortion free watermark framework for relational databases. In: ICSOFT (2), pp. 229–234. Citeseer (2009)
5. Cui, X., Qin, X., Sheng, G.: A weighted algorithm for watermarking relational databases. Wuhan Univ. J. Nat. Sci. **12**(1), 79–82 (2007)

6. Franco-Contreras, J., Coatrieux, G.: Robust watermarking of relational databases with ontology-guided distortion control. IEEE Trans. Inf. Forensics Secur. **10**(9), 1939–1952 (2015)
7. Fung, B.C., Wang, K., Yu, P.S.: Top-down specialization for information and privacy preservation. In: 21st International Conference on Data Engineering (ICDE 2005), pp. 205–216. IEEE (2005)
8. Gambs, S., Lolive, J., Robert, J.M.: Entwining sanitization and personalization on databases. In: Proceedings of the 2018 on Asia Conference on Computer and Communications Security, pp. 207–219 (2018)
9. Ghinita, G., Karras, P., Kalnis, P., Mamoulis, N.: Fast data anonymization with low information loss. In: Proceedings of the 33rd International Conference on Very Large Data Bases, pp. 758–769 (2007)
10. Guo, F., Wang, J., Li, D.: Fingerprinting relational databases. In: Proceedings of the 2006 ACM Symposium on Applied Computing, pp. 487–492 (2006)
11. Hamadou, A., Sun, X., Gao, L., Shah, S.A.: A fragile zero-watermarking technique for authentication of relational databases. Int. J. Digit. Content Technol. Appl. **5**(5) (2011)
12. Kamran, M., Farooq, M.: A comprehensive survey of watermarking relational databases research. arXiv preprint arXiv:1801.08271 (2018)
13. Kieseberg, P., Schrittwieser, S., Mulazzani, M., Echizen, I., Weippl, E.: An algorithm for collusion-resistant anonymization and fingerprinting of sensitive microdata. Electron. Mark. **24**(2), 113–124 (2014). https://doi.org/10.1007/s12525-014-0154-x
14. LeFevre, K., DeWitt, D.J., Ramakrishnan, R.: Incognito: efficient full-domain k-anonymity. In: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, pp. 49–60 (2005)
15. LeFevre, K., DeWitt, D.J., Ramakrishnan, R.: Mondrian multidimensional k-anonymity. In: 22nd International Conference on Data Engineering (ICDE 2006), pp. 25–25. IEEE (2006)
16. Samarati, P.: Protecting respondents identities in microdata release. IEEE Trans. Knowl. Data Eng. **13**(6), 1010–1027 (2001)
17. Schrittwieser, S., Kieseberg, P., Echizen, I., Wohlgemuth, S., Sonehara, N.: Using generalization patterns for fingerprinting sets of partially anonymized microdata in the course of disasters. In: 2011 Sixth International Conference on Availability, Reliability and Security, pp. 645–649. IEEE (2011)
18. Schrittwieser, S., Kieseberg, P., Echizen, I., Wohlgemuth, S., Sonehara, N., Weippl, E.: An algorithm for k-anonymity-based fingerprinting. In: Shi, Y.Q., Kim, H.-J., Perez-Gonzalez, F. (eds.) IWDW 2011. LNCS, vol. 7128, pp. 439–452. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32205-1_35
19. Shehab, M., Bertino, E., Ghafoor, A.: Watermarking relational databases using optimization-based techniques. IEEE Trans. Knowl. Data Eng. **20**(1), 116–129 (2007)
20. Sion, R., Atallah, M., Prabhakar, S.: Rights protection for relational data. IEEE Trans. Knowl. Data Eng. **16**(12), 1509–1525 (2004)
21. Sweeney, L.: Guaranteeing anonymity when sharing medical data, the datafly system. In: Proceedings of the AMIA Annual Fall Symposium, p. 51. American Medical Informatics Association (1997)
22. Sweeney, L.: Achieving k-anonymity privacy protection using generalization and suppression. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. **10**(05), 571–588 (2002)

23. Wang, H., Cui, X., Cao, Z.: A speech based algorithm for watermarking relational databases. In: 2008 International Symposiums on Information Processing, pp. 603–606. IEEE (2008)
24. Wang, K., Yu, P.S., Chakraborty, S.: Bottom-up generalization: a data mining solution to privacy protection. In: Fourth IEEE International Conference on Data Mining (ICDM 2004), pp. 249–256. IEEE (2004)
25. Xu, J., Wang, W., Pei, J., Wang, X., Shi, B., Fu, A.W.C.: Utility-based anonymization using local recoding. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–790 (2006)
26. Zhou, X., Huang, M., Peng, Z.: An additive-attack-proof watermarking mechanism for databases' copyrights protection using image. In: Proceedings of the 2007 ACM Symposium on Applied Computing, pp. 254–258 (2007)

Author Index