mathematics

MDPI

Article

A Robust and Reversible Watermarking Algorithm for a Relational Database Based on Continuous Columns in Histogram

Yan Li [1,2,*], Junwei Wang [1] and Hongyong Jia [2]

[1] PLA Strategic Support Force Information Engineering University, Zhengzhou 450001, China; wjwly79@163.com
[2] Institute of Software, Zhengzhou University, Zhengzhou 450001, China; hyjia@zzu.edu.cn
* Correspondence: ly79@zzu.edu.cn; Tel.: +86-155-1618-1121

Received: 30 July 2020; Accepted: 5 November 2020; Published: 8 November 2020

Abstract: Due to the discreteness of integer data, there are a large number of gaps and continuous columns in the histogram based on integer data. Aiming at the characteristics, this paper presents a robust and reversible watermarking algorithm for a relational database based on continuous columns in histogram. Firstly, it groups the database tuples according to the watermark length and the grouping key. Secondly, it calculates the prediction errors and uses the absolute values of the prediction errors to construct the histogram. Thirdly, it traverses the histogram to find all the continuous columns and in turn, computes the sum of the height of each continuous column and selects the group of continuous columns that has the largest sum as the positions to embed the watermarks. FCTD (Forest cover type data set) is utilized for experimental verification. A large amount of experimental data shows that the method is effective and robust. Not only does the data distortion caused by shifting histogram columns not exist, but the robustness of the watermark is also greatly improved.

Keywords: reversible database; robust watermarking algorithm; continuous columns; low data distortion; high robustness

1. Introduction

The rapid development of information technology not only brings great changes to people's daily life and production mode, but also brings explosive growth of data volume [1]. Faced with the increasing digital information, how to ensure the security and reliability of this digital information in a database has become an important research direction in the field of information security [2].

Watermarking technology is an effective method to solve the problem of digital information security protection [3–5], which has attracted extensive attention from researchers. It has four characteristics: 1. Robustness: watermark can withstand various normal data operations and the ability to resist various attacks; 2. Provability: the ability of an extracted watermark to provide copyright information to prove copyright ownership; 3. Security: non-authorized persons cannot operate on the watermark information; 4. Confidentiality: the data carrier will not cause significant data distortion due to embedding watermark. While providing value to the public with national security information such as personal privacy, medical images, trade secrets, and even military maps, big data also faces great security challenges in network communication [6–9]. For example, the Matrix data set created by 77m company uses data from RoS Land Values, but 77m's behavior is not approved by RoS, which constitutes an infringement of the rights of RoS commercial databases. In 2002, digital watermarking technology was first introduced into a relational database by the IBM Almaden research center and applied to

copyright protection of the database [10]. Since then, a variety of database watermarking technologies has appeared—for example, relational database watermarking [11,12] based on special tag tuple, digital fingerprint watermarking [13] based on a multimedia watermarking block watermarking idea, and watermarking of the relational database based on optimization technology [14]. For databases containing large amounts of information, malicious attackers copy, alter, and disseminate copyrighted information and distribute it to unauthorized users for reuse without the permission of the owner, so as to gain considerable economic benefits. Such malicious attacks not only seriously damage the rights of copyright holders, but also harm the interests of data and information units, hinder the safe development of digital information industry, and may also seriously damage economic development and endanger national security.

In view of this situation, the reversible database watermarking method emerges. It is an important technology to protect database security [15–22]. This technique can hide the watermark information in the original carrier data and recover the original carrier data without loss after extracting the watermark. When it is attacked deliberately, the reversible database watermarking technology is used to extract the watermark and recover the data under the condition of less distortion. Therefore, it is of practical significance to focus on reversible database watermarking technology and explore reversible database watermarking methods with low distortion and a high embedding amount to promote military technology progress, national information security, and social stability.

In [15], the authors first proposed a histogram column shifting method for reversible database watermarking. In [16], they combined genetic algorithm (GA) and difference expansion based reversible watermarking (DEW) and proposed a robust reversible database watermarking solution based on the GADEW method, which improved DEW capacity while maintaining certain distortion. In [17], a firefly algorithm was combined with DEW and a reversible database watermarking solution based on the FFADEW method was proposed, which improved DEW capacity and reduced data distortion. In [18], they proposed a prediction error extended watermarking (PEEW) reversible database watermarking method, which has a better anti-attack effect. In [19], the authors proposed GAHSW (Genetic Algorithm and Histogram Shifting Watermarking), a reversible database watermarking method based on numerical relational data distortion control. This method combines GA with histogram column shifting prediction error watermarking (HSW) to improve the robustness of database watermarking with minimal distortion. A low distortion reversible watermarking method HGW (Histogram-Gaps based Watermarking) is proposed in reference [20]. This method improves the traditional column shifting translation method, reduces the number of translation columns, and reduces the distortion of data. In [21], a reversible database watermarking method Non-Redundancy Shifting based Histogram-Gaps which is named Non-Redundancy Shifting Based Histogram-Gaps (NSHGW) is proposed. This method optimizes the histogram column shifting method. When embedding the watermark, there is no data distortion caused by the redundant translation of the histogram column and the change of the carrier data is greatly reduced so that there is no redundant translation distortion. However, the data distortion generated during embedding the watermark still exists.

To solve the above problems, we propose a robust and reversible watermarking algorithm for a relational database based on continuous columns in histogram (RCW). This method takes advantage of the large number of continuous columns in the histogram and improves the original histogram translation method. It finds all the continuous columns in the histogram in turn and calculates the sum of the height of each continuous columns and selects the group of continuous columns that has the largest sum as the positions to embed watermarks. The robustness of the embedded watermark using this method has been greatly improved.

The main body of the paper is as follows: in part 2, a brief overview of the relevant research is provided by highlighting the direction of our preliminary work. In part 3, the proposed reversible database watermarking scheme is described in detail. In part 4, the reliability of the experimental results is verified by Forest cover type data set (FCTD) open data set. Finally, the conclusion is given in Section 5.

2. Related Work

This section describes the HGW and NSHGW reversible database watermarking methods in detail.

2.1. HGW Reversible Database Watermarking Method

In [20], a low distortion reversible database watermarking method (HGW, Histogram-Gaps based Watermarking) was proposed. This method improves the traditional histogram shifting method, reduces the number of shifting columns, and reduces the distortion of data. The specific steps of watermark embedding are as follows:

Step 1 determines the embedded point and shifting direction. For any group i, find the highest column in the histogram, that is, the peak position as the embedding point, and mark the peak position as p_i. Search from p_i to the left and right and find the position with the first frequency of 0. These two positions are denoted as p_{iL} and p_{iR}, respectively. Calculate the sum of the heights of all the columns between p_{iL} and p_i, denoted as hs_{iL}, calculate the sum of the heights of all the columns between p_i and p_{iR}, denoted as hs_{iR}, then calculate the number of all columns between p_i, p_{iL}, and p_{iR}, respectively, denoted as d_{iL} and d_{iR}. Compare the magnitude of hs_{iL} and hs_{iR} with the smaller side as the direction of the shift.

Step 2 Shifting embed. The following two histogram shifting and watermark embedding formulas can be obtained according to the size relation of hs_{iL} and hs_{iR} values, as shown in Equations (1) and (2).

if $hs_{iL} \geq hs_{iR}$,

$$p'_h = \begin{cases} p_h + 1, & p < p_h < p_i + d_{iR} \\ p_h + w, & p_h = p_i \\ p_h, & otherwise \end{cases} \quad (1)$$

if $hs_{iL} < hs_{iR}$,

$$p'_h = \begin{cases} p_h - 1, & p_i - d_{iL} < p_h < p_i \\ p_h - w, & p_h = p_i \\ p_h, & otherwise \end{cases} . \quad (2)$$

According to the known reference [20], when histogram shifting and watermark embedding, not only the conditional comparison of the sum of the heights should be considered, but also the two cases of $p_e \geq 0$ and $p_e < 0$ should be distinguished: when $p_e \geq 0$, $p_h = p_e$; when $p_e < 0$, $p_h = -p_e$. In the HGW method, although the number of histogram columns shifting has been reduced on the original basis, there will still be a lot of redundant shifting distortion.

2.2. NSHGW Reversible Database Watermarking Method

In [21], a reversible database watermarking method without shifting distortion is proposed, which is summarized as NSHGW (Non-Redundancy Shifting Based Histogram-Gaps). This method improves the histogram column shifting method during watermark embedding and finds out all columns with space on the right side and then selects the highest column to embed on the basis of satisfying the conditions so that there is no shifting distortion. The specific steps of watermark embedding are as follows:

Step 1 Determine embedding point

For group i, start from the leftmost side of the histogram and search to the right to find each column HC_j that meets the following conditions.

If $HC_j > 0$, $HC_{j-1} = 0$ or $HC_{j+1} = 0$, HC_j is the height of the column j in the histogram.

From all qualified HC_j, choose the highest column as HC_i, take HC_i as the watermark embedding point, mark the prediction error value of the corresponding position of HC_i as p_i and mark the relationship between the blank position HC_i and GP_i. If the blank position is on the left of HC_i, let $GP_i = -1$; if the blank position is on the right of HC_i, let $GP_i = 1$.

Step 2 Watermark embedding

According to the relative relationship between HC_i and its adjacent blank position, it can be embedded as per the two following formulas:

When $GP_i = -1$,

$$p'_h = \begin{cases} p_h - w, & p_h = p_i \\ p_h, & otherwise \end{cases} \quad (3)$$

When $GP_i = 1$,

$$p'_h = \begin{cases} p_h + w, & p_h = p_i \\ p_h, & otherwise \end{cases} . \quad (4)$$

According to $p_h = |p_e|$, when $p_e \geq 0$, $p_h = p_e$; when $p_e < 0$, $p_h = -p_e$; In the case of $p_e \geq 0$ and $p_e < 0$, substitute $p_h = p_e$ and $p_h = -p_e$ into Equations (3) and (4), respectively. The watermark is embedded according to the characteristic of the histogram gap and it will be processed according to the formula of distinguishing the situation.

In the NSHGW method, the number of redundant columns in the shifting histogram has been avoided on the basis of the original method, which reduces the distortion and ensures the data quality, but there will still be some data distortion when the watermark is embedded. Meanwhile, the main focus of the NSHGW method is to reduce distortion, but its robustness can be further improved. Therefore, we propose a robust reversible watermarking algorithm for relational databases based on continuous bar graphs in part 3.

3. The Proposed Method

The RCW reversible database watermarking method discussed in this section greatly improves the robustness of watermarking while maintaining low distortion. The main architecture of the RCW approach is shown in Figure 1.

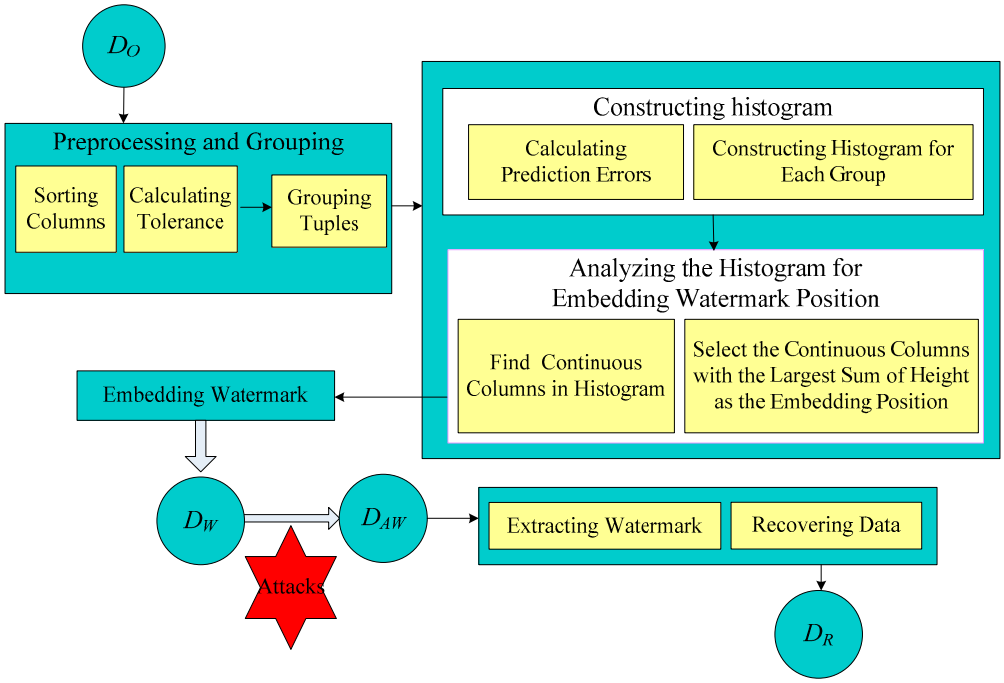


Figure 1. Schematic diagram of the relational database based on continuous columns in histogram (RCW) method.

In the schematic diagram of the RCW method, D_O is the original database, D_W is the database embedded with watermark, D_{AW} is the database after being deliberately attacked, and D_R is the original database recovered. The method includes the following four main stages: (1) preprocessing

and grouping; (2) histogram construction; (3) watermark embedding; (4) watermark extraction and data recovery.

3.1. Preprocessing and Grouping

In the preprocessing and grouping stage, two important tasks are accomplished: (1) selecting the appropriate attribute column for embedding the watermark and preprocessing the selected attribute column. First, select the desired integer attribute column in the database, then sort the attribute column by attribute name (alphabetically) and finally, determine the semantic distortion range of each attribute column. For example, select attribute columns such as score and age in the student database and then sort them into age, score, etc. in alphabetical order of the attribute column names and finally, determine the maximum and minimum values of each attribute column selected. (2) Get the block key. A random algorithm is used to obtain key information that will be used for grouping when embedding and extracting watermarks.

Step 1. Search the relational database for multiple integer property columns with identity significance and then sort the column names of the selected property columns in ascending order.

Step 2. According to the value of the attribute column, the semantic distortion range of attribute j is defined as $[min[j], max[j]]$ and the tolerance of prediction error of attribute j is calculated, of which its calculation is given in Equation (5):

$$\hat{y} = \lfloor (max[j] + min[j])/2 \rfloor, \tag{5}$$

where $max[j]$ and $min[j]$ are the maximum and minimum values of the attributes in column j, respectively.

Step 3 tuple grouping. The packet key Ks is set based on the attribute column and the tuples in the database are divided into several non-overlapping groups $\{G_i\} i = i, 2, \ldots\ldots, N_g$ by random method. The value of N_g is determined by the length of the watermark to be embedded. Use Equation (6) to determine the grouping of each tuple.

$$n_u = H(K\mathrm{s}|H(K\mathrm{s}|t_u.PK)) \mathrm{mod} N_g \tag{6}$$

In the formula, n_u represents the packet number, "|" represents the connection operation, $H()$ is a hash function, and the key Ks of the packet and the primary key $t_u.PK$ of the tuple are the parameters.

3.2. Construction of Histogram

Calculate with Equations (7) and (8) the value of p_h:

$$p_e = y - \hat{y} \tag{7}$$

$$p_h = \left| p_e \right|. \tag{8}$$

Construct a histogram for each group with p_h. y is the attribute value of a unary group in column j and p_e represents the corresponding prediction error value y, $p_h \geq 0$.

Step 1. Determine the insertion point

Step 1.1. For group i, start from the far left of the histogram and search to the right to find all consecutive column sets that meet the following conditions and calculate the sum of the heights of all columns in each column set.

(1) The leftmost column and the rightmost column of the set are blank positions.

(2) The column on the far left is the column of which its prediction error is 0 and the column on the far right is blank.

Step 1.2. Choose the column height and the largest column set from all the column sets, denoted as VC_i, and the position of the leftmost column in the set is denoted as p_i, the number of columns is

denoted as r_i, and the sum of the heights of all columns is denoted as h_i. For example, as shown in Figure 2 below, $p_i = j,\ r_i = j + 4$.

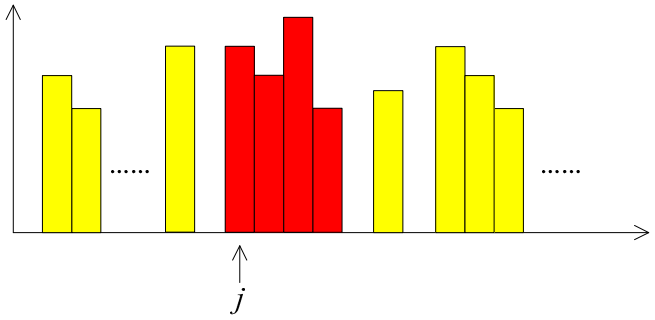


Figure 2. Continuum of histogram column gaps.

3.3. Watermark Embedding

The watermark can be embedded in VC_i according to the following formula.

$$p'_h = \begin{cases} p_h + w, & p_i \le p_h = p_i + r_i \\ p_h, & otherwise \end{cases} . \tag{9}$$

According to Equation (9), when $p_e \ge 0$, $p_h = p_e$; when $p_e < 0$, $p_h = -p_e$; Distinguish the two cases of $p_e \ge 0$ and $p_e < 0$ and substitute $p_h = p_e$ and $p_h = -p_e$ into Equation (9), respectively, then the following formula can be obtained:

$$p'_e = \begin{cases} p_e + w, & p_i \le p_e < p_i + r_i \\ p_e, & otherwise \end{cases} \tag{10}$$

$$p'_e = \begin{cases} p_e - w, & -(p_i + r_i) < p_e \le -p_i \\ p_e, & otherwise \end{cases} . \tag{11}$$

For the watermark embedding mentioned above, the case is finally treated according to Equations (10) and (11). p_i, r_i, and h_i in each grouping histogram are stored using array *pa*. Different from the existing technologies, this method uses the gaps in the histogram and the continuous columns between the gaps to calculate the total height value of each group of continuous columns in the histogram and selects the group that has the largest value of total height as embedment points. In this way, when the watermark information is embedded in the database, not only can the amount of shifting columns be reduced and the amount of distorted data, but after the watermark is embedded in the database, the data with embedded watermark information can still be used normally and the robustness of the watermark is also greatly improved.

3.4. Watermark Extraction and Data Recovery

After the watermark embedding is completed, the database embedded with the watermark is released, which may be subject to some deliberate attacks (common types of attacks include insertion, alteration, and deletion, etc.). These deliberate attacks destroy the watermark information by modifying the data in the database. In the process of watermark extraction and data recovery, it is very important to extract watermark information from a relational database and reconstruct the original database using a histogram. The specific steps are as follows:

First, the watermark embedded attribute columns are selected from the database and sorted in the same way as in the pre-processing. Using the block key generated in the watermark embedding phase,

the tuples are divided into N_g non-overlapping groups by Equation (6). The prediction error was used to construct the histogram for each group and the prediction error could be calculated by Equation (12)

$$p_e' = y' - \hat{y}, \tag{12}$$

where y' is any attribute value in D_W, $\hat{y}$ is the tolerance of the attribute column of y', which can be calculated from Equation (7), and p_e' is the prediction error corresponding to y'. After histogram construction, scan each p_e' one by one, judge the position of p_e' in the histogram, extract the watermark, and restore the database.

Step 1. Watermark extraction. The watermark is extracted using p_i, r_i, and h_i in each grouping histogram stored in array *pa*.

Step 1.1. For group i, calculate the sum of the heights of all columns within the range $[p_i, p_i + r_i)$, denoted as h'_{il}, and then calculate the sum of the heights of all columns within the range $(p_i, p_i + r_i]$, denoted as h'_{ir}.

Step 1.2. Compare h'_{il} and h'_{ir} to h_i stored in array *pa*, respectively, and record the result, of which the calculation is given in Equations (13) and (14).

$$dh_{il} = \left|h_i - h'_{il}\right| \tag{13}$$

$$dh_{ir} = \left|h_i - h'_{ir}\right|. \tag{14}$$

Step 1.3. Perform the following detection operation and record the number of all watermark bits 0 and 1 detected in this group, then use majority voting mechanism to determine the final watermark information of this group and regard the watermark bits with a large number as the final detected watermark bits. When $dh_{il} < dh_{ir}$, the watermark bit detected is 0, when $dh_{il} > dh_{ir}$, the watermark bit detected is 1. When $\left|p_e'\right| = p_i$, the property value is not modified and the watermark bit is 0. If $p_e' \geq 0$ and $p_e' = p_i + r_i$, the watermark bit detected is 1; if $p_e' < 0$ and $p_e' = -p_i - r_i$, the watermark bit detected is 1. Experiments show that the watermark extracted by the majority voting mechanism can also be used to prove the copyright owner.

Step 2. Data recovery. The watermark extraction and data recovery are shown in the schematic diagram of the RCW method in Figure 1. As shown in the formulas Equations (15)–(17), according to the watermark detection results, the following data were recovered:

When the detected watermark bit is 0

$$y^r = y', \tag{15}$$

where y^r is the attribute value after recovery. When the detected watermark bit is 1 and $p_e' \geq 0$,

$$y^r = \begin{cases} y' - 1, & p_i < p'_e \leq p_i + r_i \\ y', & otherwise \end{cases}. \tag{16}$$

When the detected watermark bit is 1 and $p_e' < 0$

$$y^r = \begin{cases} y' + 1, & -(p_i + r_i) \leq p'_e < p_i \\ y', & otherwise \end{cases}. \tag{17}$$

After the above steps, the watermark can be extracted and the copyright can be verified and the data with a watermark can be restored to its undistorted state to ensure the consistency of the data.

4. Experimental Results and Analysis

In order to experiment and evaluate the proposed watermarking information and verification technology, we tested the method. The test platform was selected on Intel Core i7 with 2.40ghz CPU and 8GB RAM for implementation and data testing.

The whole experiment was divided into two parts: the first part compared the RCW method with the data distortion rate experiment in [19–21]; the second part of the test simulated the robustness of the RCW method compared with other reversible database watermarking methods under deliberate attack. In order to make a better experimental comparison, we chose the same test database as GAHSW, HGW, and NSHGW mentioned in [19–21], which was the dataset of the Forest Cover Type dataset provided by the University of California. The dataset contains 581,012 tuples and 54 attributes. In this paper, 10 integer numeric attributes are selected for testing and compared with the existing reversible database watermarking methods. In order to make it comparable, this method, like the experimental environment of the reversible database watermarking method presented in [19–21], generates a synthetic column as the primary key, with the length (number of groups) of the watermark being 48 bits. All of the following experiments were embedded with watermarks under the same conditions.

4.1. Comparative Analysis of Data Distortion Effect

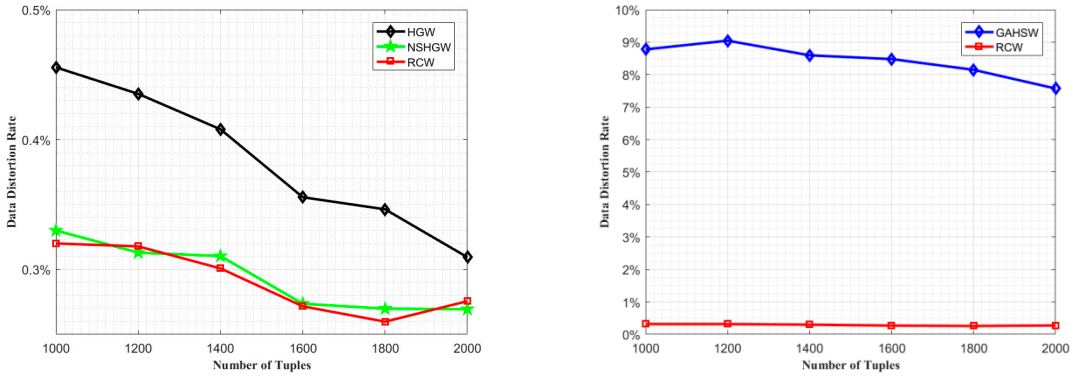
In this section, using the data distortion rate (data distortion rate, DDR) to assess the RCW, NSHGW, HGW, and GAHSW methods embed watermark after the distortion of the effects on the database. DDR is calculated, of which its calculation is given in Equation (18) as follows:

$$\mathrm{DDR} = \frac{T_{dis}}{TD}, \tag{18}$$

where T_{dis} is the total data of distortion and TD is the total data in the database. The larger the DDR, the greater the distortion of the data, and vice versa. Experimental results show that the distortion rate of RCW and NSHGW is similar, slightly lower than that of the HGW method, and much lower than that of the GAHSW method.

In experiments, tuples (rows) in a database are grouped according to the length of the watermark. When the watermark length is 48, it represents the number of values grouped in the database. RCW, NSHGW, HGW, and GAHSW were compared in turn and their distortion rates were verified when the total tuples were 1000, 1200, 1400, 1600, 1800, and 2000, respectively, with the same watermark. The experimental results were drawn as shown in Figure 3. In the figure, the vertical axis is DDR, which represents the ratio of data distortion; when DDR is zero, it represents the data of the database without distortion; and the horizontal axis represents the number of different tuples. The distortion rates of RCW, NSHGW, HGW, and GAHSW methods are the average of the results of 15 runs of the four methods, respectively. To show the distortion ratio more clearly, we will show the effect through Figure 3a,b.

It can be clearly seen from Figure 3a,b above that when embedding the same watermark information into the same database, the DDR generated by the RCW and NSHGW methods is roughly equivalent, which is slightly lower than HGW and far lower than GAHSW, both less than 0.3%.



(**a**) (**b**)

Figure 3. The comparison of data distortion rates caused by 48-bit watermarks. (**a**) The Comparison of distortion rates for RCW, Histogram-Gaps based Watermarking (HGW), and Non-Redundancy Shifting Based Histogram-Gaps (NSHGW); (**b**) The Comparison of distortion rates for RCW and Genetic Algorithm and Histogram Shifting Watermarking (GAHSW).

4.2. Watermark Robustness Analysis

In this section, the robustness of the RCW method under well-known database attacks is reported. For the convenience of comparison, the method in this paper is the same as the experiments of HGW and NSHGW in [20,21]. The watermark length is set to 48 for experimental testing. Through attack analysis, this paper verifies the robustness of the RCW method. This article focuses on testing three types of attacks: insert, delete, and alter. Robustness is evaluated by a bit error rate (BER), which is the ratio of the number of error-extracted bits to the number of embedded watermark bits. BER is calculated and its calculation is given as follows:

$$\mathrm{BER} = \frac{\sum_{i=1}^{N_g} w_i \oplus w_i^{det}}{N_g}, \tag{19}$$

where w_i is the embedded watermark bit and w_i^{det} is the detected watermark bit. We can see that the lower the BER value, the higher the watermark robustness. Experiments show that RCW is more robust than NSHGW and HGW. Next, we conduct the attack experiment in the best case and the worst case, respectively, to demonstrate the watermark detection and data recovery results and compare them.

We simulated the attacker trying to insert, delete, and alter 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, and 90% of the data, respectively. The results of deletion and modification were plotted in Figures 4 and 5, respectively. In the figure, the vertical axis is BER, indicating the rate of unsuccessful detection of the watermark. When BER approaches zero, the watermark is correctly detected from the database. The horizontal axis represents the percentage (%) of the tuple change in the database after the attack. Because RCW relies on randomness, the BER extracted from the watermark is the average of the 15 runs of the method.

RCW, NSHGW, and HGW have good robustness for insert attacks. No matter how many tuples are inserted into the database, the BER under the insertion attack is always 0, which will not affect the embedded watermark information. Since the watermark information is embedded in the original database, the original watermark information will not be destroyed by this attack after the insertion tuple is deliberately attacked, so the comparison diagram of the insertion attack will not be drawn here. As can be seen from Figures 4 and 5, RCW is more robust than NSHGW and HGW methods under deletion attack and modification attack. As the attack intensity increases, the BER of extracting watermark by these two methods also increases. Experiments also show that RCW has a better watermark detection rate than other methods.

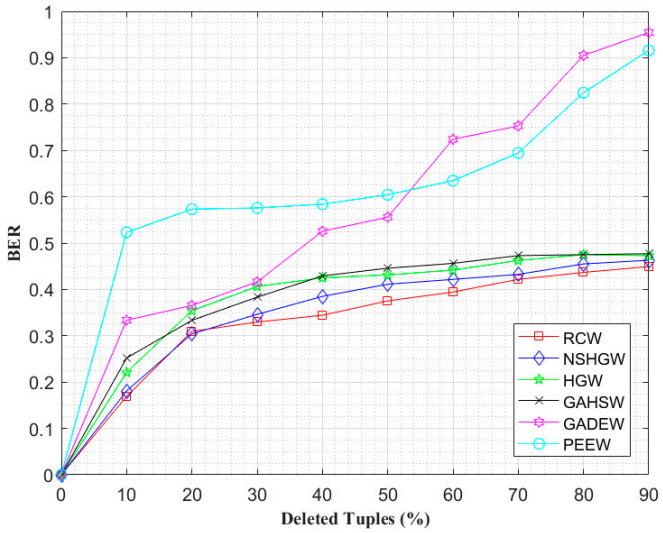


Figure 4. Deletion attacks bit error rate (BER) comparison for RCW, NSHGW, HGW, Genetic Algorithm and Histogram Shifting Watermarking (GAHSW), prediction error extended watermarking (PEEW), and GADEW algorithms.

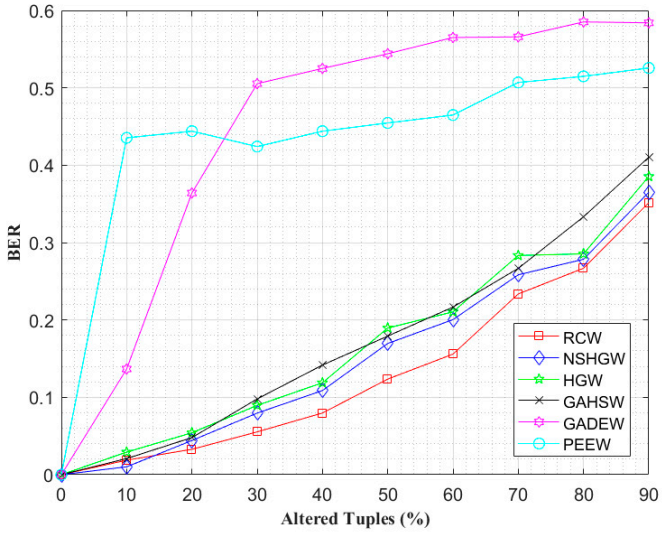


Figure 5. Alteration attacks BER comparison for RCW, NSHGW, HGW, GAHSW, PEEW, and GADEW algorithms.

Deletion attack is a random deletion of tuples to destroy the watermark. When the database is attacked by deletion, as the number of deleted tuples increases, the proportion of deleted tuples containing watermark information will also increase and the bit error rate of the extracting watermark will also increase. Figure 4 shows the BER of the watermark extracted by RCW, NSHGW, HGW, GAHSW, PEEW, and GADEW methods after removing the attack. For example, when the database is severely deleted, the proportion of tuple deleted in the database becomes 90%. NSHGW, HGW, GAHSW, GADEW, and PEEW extract the watermark with BER values of 0.462, 0.472, 0.477, 0.954, and 0.916, respectively. However, the BER extracted by the RCW watermark is 0.449. If you delete most of the watermark information contained in the tuple, you cannot recover the watermark. If it is only a little more than half, the RCW method can be restored. To sum up, deletion of the tuple attack has the greatest impact on the database watermarking information.

Alteration attack is a random modification of a database property. After the attacker makes the modification tuple attack on the database, we can restore the damaged database as long as most of

the data embedded in the watermark information are not modified. Figure 5 shows the BER of the watermark extracted by RCW, NSHGW, HGW, GAHSW, PEEW, and GADEW methods after modifying the attack. The experimental results show that the BER of the extracted watermark increases with the increase of data modification. The number of modified tuple in the database was changed to 90%. NSHGW, HGW, GAHSW, GADEW, and PEEW extracted watermarks with BER values of 0.365, 0.385, 0.410, 0.584, and 0.526, respectively. However, the BER extracted by the RCW watermark is 0.352. In a database, it is difficult to extract the watermark tuples affected by the attack from the remaining unaffected data if the majority of the watermark tuples are modified. If it is only about half of these tuples, it can be recovered.

In general, the RCW method produces a lower BER when embedding the same watermark into the same database than HGW and NSHGW.

5. Conclusions

In order to improve the robustness of the reversible database watermark and ensure the characteristics of low distortion of data, this paper presents a robust and reversible watermarking algorithm for relational database based on continuous columns in histogram (RCW). This method takes advantage of a large number of gaps in the histogram and the feature of the continuous vertical column and selects the continuous vertical column with the maximum height and as the embedding point to embed the watermark, thus greatly improving the robustness of the watermark. Experimental results show that the RCW method is similar to the NSHGW method in data distortion effect analysis and is slightly lower than the HGW method and far lower than the GAHSW method. In terms of robustness analysis, it is obviously superior to the existing NSHGW and HGW methods. When under deliberate attacks, the bit error rate of watermark extraction is much lower than NSHGW and HGW. In the future, the goal of my research is to find a better reversible database watermarking method, which can further improve watermark embedding capacity and efficiency while maintaining low distortion.

Author Contributions: Data curation, J.W.; Formal analysis, H.J.; Writing–original draft, Y.L. and J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The work presented in this paper is supported by Science and Technology Research Project in Henan Province (No.192102210115).

Conflicts of Interest: The authors declare no conflict of interest to this work. The authors declared that they do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

References

1. Gursale, N.; Mohanpurkar, A. A robust, Distortion minimization fingerprinting technique for relational database. *Int. J. Recent Innov. Trends Comput. Commun.* **2014**, *2*, 737–1741.
2. Shehab, M.; Bertino, E.; Ghafoor, A. Watermarking relational databases using optimization-based techniques. *IEEE Trans. Knowl. Data Eng.* **2008**, *20*, 116–129. [CrossRef]
3. Khanduja, V.; Chakraverty, S.; Verma, O. Enabling information recovery with ownership using robust multiple watermarks. *J. Inf. Secur. Appl.* **2016**, *29*, 80–92. [CrossRef]
4. Gupta, G.; Pieprzyk, J. Database relation watermarking resilient against secondary watermarking attacks. In *Information Systems Security: 5th International Conference, Proceedings of the Lecture Notes in Computer Science, Volume 5905, Kolkata, India, 14–18 December 2009*; Springer: Berlin/Heidelberg, Germany; pp. 222–236.
5. Bhattacharya, S.; Cortesi, A. A distortion free watermark framework for relational databases. In Proceedings of the ICSOFT 2009—4th International Conference on Software and Data Technologies, Sofia, Bulgaria, 26–29 July 2009; Volume 2, pp. 229–234.
6. Iftikhar, S.; Kamran, M.; Anwar, Z. RRW-a robust and reversible watermarking technique for relational data. *IEEE Trans. Knowl. Data Eng.* **2015**, *27*, 1132–1145. [CrossRef]

7. Mo, Q.; Yao, H.; Cao, F. Reversible data hiding in encrypted image based on block classification permutation. *Comput. Mater. Contin.* **2019**, *59*, 119–133. [CrossRef]
8. Wang, B.W.; Kong, W.W.; Li, W. A dual-chaining watermark scheme for data integrity protection in Internet of Things. *Comput. Mater. Contin.* **2019**, *58*, 679–695. [CrossRef]
9. Liu, J.; Li, J.B.; Cheng, J.R. A novel robust watermarking algorithm for encrypted medical image based on DTCWT-DCT and chaotic map. *Comput. Mater. Contin.* **2019**, *61*, 889–910. [CrossRef]
10. Agrawal, R.; Kiernan, J. Watermarking relational databases. In Proceedings of the 28th VLDB Conference, Hong Kong, China, 20–23 August 2002; pp. 155–166.
11. Sion, R. Proving ownership over categorical data. In Proceedings of the 20th International Conference on Data Engineering, Boston, MA, USA, 2 April 2004; pp. 584–596.
12. Sion, R.; Atallah, M.; Prabhakar, S. Rights protection for categorical data. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 912–926. [CrossRef]
13. Liu, S.Y.; Wang, S.H.; Dengetal, R. A block oriented fingerprinting scheme in relational database. In *Information Security and Cryptology—ICISC 2004, Proceedings of the 7th International Conference, Seoul, Korea, 2–3 December 2004*; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3506.
14. Chang, J.; Wu, H. Reversible fragile database watermarking technology using difference expansion based on SVR prediction. In Proceedings of the International Symposium on Computer, Consumer and Control, Taiwan, China, 4–6 June 2012; pp. 690–693.
15. Zhang, Y.; Yang, B.; Niu, X.M. Reversible watermarking for relational database authentication. *J. Comput.* **2006**, *17*, 59–65.
16. Jawad, K.; Khan, A. Genetic algorithm and difference expansion based reversible watermarking for relational databases. *J. Syst. Softw.* **2013**, *86*, 2742–2753. [CrossRef]
17. Imamoglu, M.B.; Ulutas, M.; Ulutas, G. A new reversible database watermarking approach with firefly optimization algorithm. *Math. Probl. Eng.* **2017**, *2017*, 1–14. [CrossRef]
18. Mahmoud, E.; Farfoura, H.; Wang, X. A novel blind reversible method for watermarking relational databases. *J. Chin. Inst. Eng.* **2013**, *36*, 87–97.
19. Hu, D.H.; Zhao, D.; Zheng, S.L. A new robust approach for reversible database watermarking with distortion control. *IEEE Trans. Knowl. Data Eng.* **2018**, *31*, 1024–1037. [CrossRef]
20. Li, Y.; Wang, J.W.; Ge, S.K. A reversible database watermarking method with low distortion. *Math. Biosci. Eng.* **2019**, *16*, 4053–4068. [CrossRef] [PubMed]
21. Li, Y.; Wang, J.W.; Luo, X.Y. A reversible database watermarking method non-redundancy shifting-based histogram gaps. *Int. J. Distrib. Sens. Netw.* **2020**, *16*, 155014772092176. [CrossRef]
22. Franco-Contreras, J.; Coatrieux, G.; Cuppens, F.; Cuppens-Boulahia, N.; Roux, C. Robust Lossless Watermarking of Relational Databases Based on Circular Histogram Modulation. *IEEE Trans. Inf. Forensics Secur.* **2017**, *9*, 397–410. [CrossRef]