A Robust Reversible Watermarking Scheme for Relational Data

Ruitao Hou[1,2], Hequn Xian[1,2(✉)], Xiao Wang[3], and Jing Li[1,2]

[1] College of Computer Science and Technology, Qingdao University,
Qingdao 266071, China
xianhq@126.com

[2] State Key Laboratory of Integrated Services Networks, Xidian University,
Xi'an 710071, China

[3] Qingdao Ocean Shipping Mariners College, Qingdao 266071, China

Abstract. Reversible watermarking is widely used in copyright protection of relational data. It allows recovering the original data besides claiming copyright. In current schemes, watermarked data are either completely restored to the original version or kept unchanged. We present a robust and reversible watermark which allows arbitrary portion of the watermark to be removed. Experiments show the robust of the proposed algorithm is robust.

Keywords: Watermark · Reversible · Copyright · Relational data

1 Introduction

Reversible watermarking technique for relational data is an effective method to protect copyright which is developed from traditional watermarking technique for relational data [1, 2]. It allows the inversion of watermark embedding to recover the original data. However, existing reversible watermarking schemes for relational data still have some problems. Suppose Alice is the owner of some relational data, and embeds a watermark into her data with a reversible watermarking technique before distributing them to user Bob. If Bob finds out that the usability of the data does not meet his requirements, he can purchase relevant keys from Alice to perform the inverse operation of watermark embedding. Thus, Bob obtains the original data. After that, if Bob sells the recovered data to others without Alice's consent, Alice will not be able to claim copyright because there is no longer a watermark in the recovered data. In the above scenario, although usability can be enhanced via recovering the original data, the owner loses the ability to claim the copyright permanently. This problem mainly stems from the facts that existing schemes cannot control the extent of data recovery and all watermarks are removed during the data recovery process. Therefore, enhancing the usability of data while simultaneously preserving the copyright claim has become a new research focus.

In this paper, a robust reversible watermarking scheme for relational data named GRW is proposed. Quality grade is defined to describe the impact of watermark embedding on the usability of data. Four fundamental algorithms are designed to facilitate the processes of watermark embedding, data quality grade detection, watermark

J. Li et al. (Eds.): SPNCE 2019, LNICST 284, pp. 545–550, 2019.
https://doi.org/10.1007/978-3-030-21373-2_44

detection, and data quality grade enhancement. Reversibility can be achieved by upgrading watermarked data from a low data quality grade to higher grades.

2 Related Work

The first reversible watermarking scheme for relational data were proposed by Zhang et al. [3]. In this scheme, a histogram with difference values is used to achieve watermark reversibility. Gupta and Pieprzyk proposed a reversible watermarking scheme (DEW) [4]. They used difference expansions to achieve the reversibility of the watermark. However, this scheme is less robust to tuple alteration attacks. Combining genetic algorithms and difference expansion, Jawad and Khan considered the watermarking problem as a constrained optimization problem and applied difference expansion to achieve reversibility (GADEW) [5]. Franco-Contreras proposed a robust reversible watermarking scheme based on circular histogram transforms [6]. In this scheme, a circular histogram transform was constructed, and relative angular positions of some attribute values were changed to implement watermark embedding. Iftikhar et al. utilized genetic algorithms and a data analysis method in information theory to deal with the watermarking problem (RRW) [7]. They used genetic algorithms to generate the optimal watermark to minimize data distortion. However, the generation of an optimal watermark requires heavy computation, and the efficiency is unsatisfactory when processing very large volumes of data. Farfoura et al. converted a recognizable image into a bit stream, which was embedded into the least significant bits of attribute values (PEEW) [8]. They utilized prediction error expansion of integers to achieve reversibility of the watermark. Imamoglu designed a reversible watermarking method using the firefly algorithm and difference expansion [9]. The firefly algorithm was used to select the optimal attribute pairs, which were then embedded as the watermark. Jiang et al. divided relational data into blocks, and a watermark was embedded into the wavelet domain of these data blocks. The wavelet transform was used to implement watermark reversibility [10]. Although data recovery can be achieved using the above schemes, they do not allow control of the extent of data recovery. GRW is proposed in this work to solve this problem.

3 GRW Scheme

GRW consists of four procedures: (1) watermark partition embedding; (2) data quality grade detection; (3) watermark detection; and (4) data quality grade enhancement.

3.1 Quality Grade

Definition 1. Quality grade, QD, is the quantified value of data usability under the impact of watermark embedding. $QD \in [0, \lambda - 1], QD \in N$, where λ indicates the number of data partitions.

According to Definition 1, the owner can divide the data into λ partitions. When *QD* equals 0, all data partitions contain watermarked tuple. When *QD* is equal to $\lambda - 1$, there is no watermark. When the value *QD* is *q*, the number of partitions containing watermarked tuple is $\lambda - 1 - q$.

3.2 Watermark Partition Embedding

Before the watermark is embedded into the original data, the watermark and an auxiliary string *S* need to be created. The data owner can convert some identification information into a binary sequence. *S* is created randomly, but the length of it is equal to the length of the watermark. *S* has auxiliary roles in data quality grade detection, watermark detection, and data quality grade enhancement.

The watermark partition embedding algorithm is shown as Algorithm 1.

Algorithm 1. Watermark partition embedding

Input: original data, data partition key, watermark embedding key, watermark, auxiliary string

Output: watermarked data, auxiliary data

```
for each tuple  in the original data do
  Calculate the data partition of the tuple
  if (the tuple is supposed to contain watermark bit) then
      Calculate the location the watermarked  bit;
      Calculate the watermark bit $W[l]$
      Calculate the corresponding bit in auxiliary string $S[l]$;
      Calculate  the auxiliary data bit;
      Store he auxiliary data bit in storage structure of auxiliary data;
      Obtain the watermark bit with $W[l]$ xor $S[l]$;
      Calculate  the original bit in the tuple;
      Update the  tuple and set the value of wat
      Update the original bit in the tuple with $W[l]$ xor $S[l]$
  end if
end for
Return the data watermarked and the auxiliary data
```

3.3 Quality Grade Detection

Quality grade detection is the preliminary procedure of watermark detection and data quality grade enhancement. Its purpose is to confirm data partitions watermarked. The data quality grade detection algorithm is shown in Algorithm 2.

Algorithm 2. Quality grade detection

Input: watermarked data, storage structure of auxiliary data, data partition key, watermark embedding key, auxiliary string

Output: quality grade

```
Initialize data quality grade $QD$
for each tuple in the watermarked data do
  Determine which data partition the tuple belongs to;
  if (the tuple contains watermark bit) then
      Locate the watermarked bit;
      Select a bit from the corresponding auxiliary string;
      Calculate the auxiliary data bit;
      Store the auxiliary data bit in a temporary data structure
  end if
end for
for each data partition do
  if(the bit in temporary data structure matches the bit in auxiliary data for current data partition then $QD$ ++;
Return $QoD$;
```

3.4 Watermark Detection

The watermark detection algorithm is shown as Algorithm 3. We use the majority voting mechanism [2] to improve the accuracy of detection.

Algorithm 3. Watermark detection

Input: watermarked data, storage structure of auxiliary data, data partition key, watermark embedding key, auxiliary string , quality grade

Output: watermarked data

```
Determine which data partitions are supposed contain the watermark according to the value of quality grade;
for each tuple in the watermarked data do
  Determine which data partition the tuple belongs to;
  if (data partition is supposed to have watermark) then
    if (the tuple is supposed to carry watermark) then
      Locate the watermarked bit;
      Select a bit from the corresponding auxiliary string;
      Get the auxiliary data bit;
      Calculate the watermark bit $W_D[l]$
      Use majority voting mechanism to obtain the watermark bit;
  end if
end for
Count the voting result for the intended watermark
Return the detection result $W_D$
```

3.5 Quality Grade Enhancement

The purpose of quality grade enhancement is to reverse the watermark embedded in some data partitions. The algorithm is shown as Algorithm 4.

Algorithm 4. Data quality grade enhancement

```
Input: watermarked data, storage structure of auxiliary data, part of data partition key,
Watermark embedding keys, quality grade
Output: relational data after enhancing data quality grade
for each tuple in watermarked data do
  if (the tuple belongs to a watermarked partition) then
    if (the tuple contains watermark bit) then
        Locate the watermarked bit;
        Select a bit from the corresponding auxiliary string;
        Calculate the auxiliary data bit;
        Obtain the watermark bit with xor calculation;
        Calculate the original bit in the tuple;
        Update the tuple and set the value of watermark bit to the original bit
    end if
Return the data after quality grade enhancement
```

4 Experiments

To verify the robustness of GRW, we simulated tuple deletion attack on watermarked data. An adversary randomly deletes some tuples from watermarked data with intention to destroy the watermark. The setting of the experiments are as follows: 200000 tuples, 20 data partition, 1 water marked tuple in 4, 53 attributes capable for watermark. We compared GRW with DEW [4], GADEW [5], RRW [7], and PEEW [8].

Watermark Detection. As shown in Fig. 1, the detection accuracy of GRW and RRW remained at 100% when up to 90% tuples were attacked. When a large proportion of tuples were deleted, the detection accuracy of other schemes decreased.

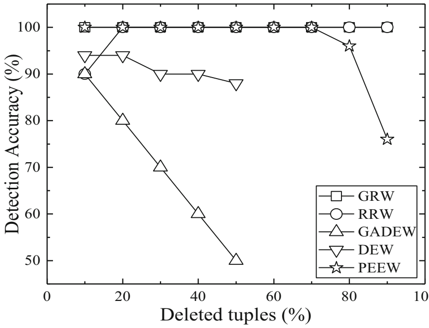

Fig. 1. Watermark detection accuracy after tuple deletion attacks

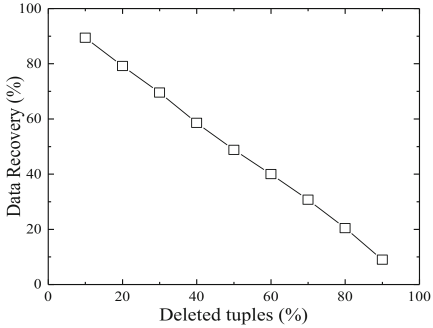

Fig. 2. Data recovery accuracy after tuple deletion attacks

Data Recovery. As shown in Fig. 2, when 50% of tuples were deleted, 48.81% watermarked tuples could be recovered to the original data as 51.19% watermarked tuples were deleted by the adversary. All remaining watermarked tuples could be successfully recovered.

5 Conclusions

Existing reversible watermarking schemes remove all watermarks in the data and do not allow control of the extent of data recovery. Once the original data is recovered, the owner loses the protection of copyright. A robust reversible watermarking scheme for relational data is proposed in this paper. The quality grade can be enhanced incrementally by removing arbitrary portions of the watermark. Experiments showed that the proposed scheme has high robustness against tuple deletion attack.

Acknowledgment. This research is supported by the Open Project Program of The State Key Laboratory of Integrated Services Networks (ISN19-14), Hequn Xian is the corresponding author.

References

1. Agrawal, R., Kiernan, J.: Watermarking relational databases. In: Proceedings of the 28th International Conference on Very Large Data Bases, pp. 155–166. VLDB Endowment (2002)
2. Sion, R., Atallah, M., Prabhakar, S.: Rights protection for categorical data. IEEE Trans. Knowl. Data Eng. **17**(7), 912–926 (2005)
3. Zhang, Y., Yang, B., Niu, X.-M.: Reversible watermarking for relational database authentication. J. Comput. **2**(17), 59–65 (2006)
4. Gupta, G., Pieprzyk, J.: Reversible and blind database watermarking using difference expansion. In: Proceedings of the 1st International Conference on Forensic Applications and Techniques in Telecommunications, Information, and Multimedia and Workshop, pp. 24–29. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering) (2008)
5. Jawad, K., Khan, A.: Genetic algorithm and difference expansion based reversible watermarking for relational databases. J. Syst. Softw. **11**(86), 2742–2753 (2013)
6. Franco-Contreras, J., Coatrieux, G., Cuppens, F., et al.: Robust lossless watermarking of relational databases based on circular histogram modulation. IEEE Trans. Inf. Forensics Secur. **9**(3), 397–410 (2014)
7. Iftikhar, S., Kamran, M., Anwar, Z.: RRW—a robust and reversible watermarking technique for relational data. IEEE Trans. Knowl. Data Eng. **4**(27), 1132–1145 (2015)
8. Farfoura, M.E., Horng, S.J.: A novel blind reversible method for watermarking relational databases. J. Chin. Inst. Eng. **1**(36), 87–97 (2013)
9. Imamoglu, M.B., Ulutas, M., Ulutas, G.: A new reversible database watermarking approach with firefly optimization algorithm. Math. Probl. Eng. **2017**(2), 1–14 (2017)
10. Jiang, C.X., Cheng, X.H., Xu, X.L., et al.: Reversible database watermark based on integer wavelet transform. J. Guilin Univ. Technol. **37**(1), 191–195 (2017)