Research Article

Reversible Fragile Watermarking Scheme for Relational Database Based on Prediction-Error Expansion

Ali Hamadou ¹, Lanciné Camara,² Abdoul Aziz Issaka Hassane,¹ and Harouna Naroua³

¹Department of Mathematics, Dan Dicko Dankoulodo University of Maradi, Maradi, Niger

²Social Science and Management University of Bamako, Bamako, Mali

³Department of Mathematics and Computer Science, Abdou Moumouni University, Niamey, Niger

Correspondence should be addressed to Ali Hamadou; alihamadou@yahoo.fr

Received 29 January 2020; Revised 2 April 2020; Accepted 15 April 2020; Published 5 May 2020

Academic Editor: Haipeng Peng

Copyright © 2020 Ali Hamadou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The protection of database systems content using digital watermarking is nowadays an emerging research direction in information security. In the literature, many solutions have been proposed either for copyright protection and ownership proofing or integrity checking and tamper localization. Nevertheless, most of them are distortion embedding based as they introduce permanent errors into the cover data during the encoding process, which inevitably affect data quality and usability. Since such distortions are not tolerated in many applications, including banking, medical, and military data, reversible watermarking, primarily designed for multimedia content, has been extended to relational databases. In this article, we propose a novel prediction-error expansion based on reversible watermarking strategy, which not only detects and localizes malicious modifications but also recovers back the original data at watermark detection. The effectiveness of the proposed method is proved through rigorous theoretical analysis and detailed experiments.

1. Introduction

With the rapid evolution of computer and Internet technology, the use of databases has enormously increased. Generally, database relations contain critical information about organizations and individuals that need to be protected from illegal copying and forgery. The piracy problem has become one of the most devastating threats to networking systems and electronic business. While demand for the use of databases is growing, pirated copying has become a severe threat to such databases due to the low cost of copying and the high values of the target databases. In that context, the security of published or shared data has become a great concern for data owners as the creation of database involves intellectual and financial effort [1].

In the last few years, digital watermarking had been proposed as a promising way of protecting digital data from illegal copying and manipulation. Digital watermarking deploys information hiding to conceal a “piece of secret information” (watermark) inside a digital data for the

purpose of ownership proofing or data integrity control. The soundness of such a method relies on the assumption that altering the digital data in the process of hiding the watermark does not destroy the data usefulness. However, the errors introduced to data during the watermark embedding process inevitably reduce data quality. In addition, permanent distortions in many applications (e.g., military, law-enforced, and medical data) are not tolerated. Most of the distortion-embedding watermarking schemes are irreversible. That is, they do not allow original data recovering from the watermarked relation [1–4]. Therefore, although some distortion-free schemes exist in the literature [1–9], there is a need to develop a new generation of watermarking techniques that could efficiently address the abovementioned issues. To achieve that, reversible or lossless digital watermarking has been extended from multimedia assets to relational data to allow original data recovery at watermark detection. Reversible watermarking controls the distortions introduced by watermark embedding, and it ensures data recovery along with ownership protection or tamper

proofing. Several reversible methods are available in the literature for watermarking relational databases [5]. Nevertheless, most of the proposed solutions are robust as they have been proposed for copyright protection and ownership proofing.

In this paper, we propose a novel fragile reversible watermarking technique for integrity control of relational data using prediction-error expansion. The main characteristics of our proposed method can be summarized as follows:

Reversibility: distortions introduced to the underlying data are not permanent since the original data can be fully restored at watermark detection. As a result, the scheme is especially suitable for applications that require zero permanent distortion.

Imperceptibility: the watermark is encoded in a way it should not be unnoticeable and undetectable using conventional techniques.

Fragility: any modification maliciously made to the watermarked relation can be detected and localized with high probability.

Blindness: the original database relation is not required for watermark detection.

Usability: distortions caused by watermark insertion do not comprise data usefulness.

Security: both embedding and detection stages of the watermark are governed by the use of a secret key known only to the data owner.

2. Previous Work

Several recent surveys have provided detailed discussion on relational database watermarking [1–5]. Robust reversible watermarking has also been investigated in many recent works [10–13]. In this section, we mainly focus on the work done on lossless watermarking for database tamper proofing and authentication. Notice that most of the available schemes have been adapted from techniques proposed for multimedia objects [14].

In [15], Zhang et al. proposed the first well-known reversible watermarking scheme for relational databases. This method is designed to achieve lossless and exact authentication of numeric relational data using expansion on data error histogram. This scheme is claimed to possess the ability of perfect restoration of the original attribute data from the nontampered watermarked relational databases, thus guaranteeing a “clear and exact” tampered-or-not authentication without causing any permanent distortion to the database. The disadvantage of this method is because consecutive tuples in database relation are not necessarily correlated as pixels in an image, which may seriously reduce the embedding capacity. As a result, the scheme is not resilient to heavy attacks.

Coatrieux et al. [16] described a lossless watermarking technique for integrity control of categorical data in medical databases. This technique, before watermark embedding, organizes the relation tuples into secret groups using a one-

way hash function. The watermark is encoded in groups based on histogram shifting modulation to form a digital signature that will be used to verify the integrity of the associated group.

In [17], the authors introduced a method, which can not only detect malicious modifications but also recover the true data from the modified locations in database relations. This scheme follows the same algorithmic steps as in [18], at the difference that unlike in [17], the XOR operator is used as hash function for generating the attribute watermark. This operator has heredity property, which enables to recover back the true data from the watermarked database table.

Franco-Contreras et al. [19, 20] proposed a lossless watermarking based on circular interpretation of bijective modulations for integrity verification of medical relational databases. The proposed technique modulates the relative angular position of the circular histogram center of mass of one numeric attribute in the relation. This scheme can be used for copyright protection, integrity control, or traitor tracing.

In a recent work, Khandudja and Chakraverty [21] have introduced a fragile, blind, and reversible method for tamper detection in decision systems. The scheme, based on the rough set theory, first prepares secure signature by encoding the information on reducts, rules, and their support values and then securely embeds it into a dataset. In case of alteration in reducts, the proposed technique can recover the original value.

Recently, in [22], a reversible watermarking technique for tamper detection and original data recovery in relational databases was presented. In the proposed work, two optimization techniques, orthogonal learning particle swarm optimization (OLPSO) and firefly algorithm (FA), are used to find the optimal locations in the database for embedding the watermark. In case of tampering, the method has the ability to restore the database back to the original state.

In their work, Chang and Wu [23] presented a reversible technique for tamper detection of numeric attributes. The foregoing scheme is based on difference expansion and support regression vector (SVR) techniques. Difference expansion is a well-known lossless watermarking modulation for message embedding. It performs arithmetic operations on numeric features causing modifications by expanding the differences between original and predicted values adding one virtual least significant bit (LSB) that is used for message embedding. However, it is weak against update attacks.

In [24], a tamper recovery fragile watermarking technique for relational databases was proposed. This scheme is group based, and the watermark is embedded and verified group-by-group using Reed–Solomon coding technique.

In [25], Gupta and Pieprzyk proposed a robust reversible watermarking technique, which extended the Agrawal and Kierman method [26]. They used difference expansion on integers and achieved reversibility of the original data.

In [27], Farfoura and Horng proposed a blind and reversible method which improves the scheme in [25] by reducing data distortion. The proposed method is based on prediction-error expansion (originally proposed for the

image [28]) instead of difference expansion, and a single attribute is used to carry a watermarking bit. For watermark insertion, an identification image is converted into a bit stream. It is then embedded into fractional part of numeric attributes to represent copyright watermark information [27].

In [29], the authors extended their work [27] to handle the situation when multiple owners claim the ownership of watermarked data. A time-stamping authentication protocol is designed to ensure ownership protection for relational databases. The motivation behind watermark embedding in the fractional portion is to reduce the distortion in the underlying data. If the numeric feature has no fractional part, no watermark will be embedded; hence, the watermark capacity decreases. In addition, although this method is robust against common database manipulations, the watermark will not survive a simple rounding integer operation.

In [30], another blind and reversible technique based on PE is proposed. The intention behind the design of this technique is to provide proof of ownership and tries to overcome the drawbacks of Farfoura et al. watermarking technique [29]. During watermark embedding phase, the last two digits from all numeric attributes of the selected tuples are used. Although the robustness of the technique has been analyzed against malicious attacks and benign updates, it cannot be used for tamper detection and localization.

3. Prediction-Error Expansion Overview

The embedding and extraction procedures of the conventional prediction-error expansion (PE) proposed by Thodi and Rodriguez in [28] are briefly introduced here. In PE, the prediction-error between the original image pixel value and the estimated (predicted) value is utilized for watermark embedding. Moreover, authors suggest incorporating expansion embedding with histogram shifting to ensure the reversibility.

The concrete embedding procedure of watermarking of the PE approach is described as follows: let x be the intensity of a pixel in a gray-scale image. The predicted value $\hat{x}$ of the pixel x is computed as follows:

$$\hat{x} = \begin{cases} \max(a, b), & \text{if } c \leq \min(a, b), \\ \min(a, b), & \text{if } c \geq \max(a, b), \\ a + b - c, & \text{otherwise,} \end{cases} \quad (1)$$

where a , b , and c are the right, lower, and diagonal neighbors of the pixel x (see Figure 1).

The prediction error e is defined by

$$e = x - \hat{x}. \quad (2)$$

A watermark bit i is embedded by expanding the prediction error as follows. The binary representation of the prediction error e is shifted left by one bit to create a vacant LSB, into which the marking bit i is inserted. As a result, the modified (expanded) prediction error e_w is given by

$$e_w = 2e + i. \quad (3)$$

FIGURE 1: Neighborhood of pixel x .

The watermarked pixel value x_w can then be computed using the expanded prediction error and the pixel estimated value $\hat{x}$:

$$x_w = \hat{x} + e_w = x + e + i. \quad (4)$$

At the decoder, extraction is done by first calculating the prediction error e_w . The encoded bit is extracted from the LSB of e_w . The original prediction error e is then computed as

$$e = \lfloor \frac{e_w}{2} \rfloor. \quad (5)$$

Finally, the original pixel value is restored as

$$x = x_w - e - i. \quad (6)$$

For example, assume $x = 223$, $a = 208$, $b = 213$, and $c = 213$. Then, the prediction error is given by $e = 223 - 208 = 15$. Let $i = 1$ be the embedding watermark bit. The expanded prediction error and the watermarked pixel values can be computed as follows: $e_w = 2 * 15 + 1 = 31$ and $x_w = 208 + 31 = 239$. At the detection phase, the concealed bit and the original prediction error are given by $i = \text{LSB}(e_w) = 1$ and $e = \lfloor 31/2 \rfloor = 15$. Lastly, the original pixel value can be restored as follows: $x = 239 - 15 - 1 = 223$.

4. Proposed Method

As discussed in Section 2, in the robust method proposed by Farfoura and Horng [27], the watermark embedding is not possible if the selected attributes for marking do not have fractional part. To overcome this limitation, in our scheme, instead of the fractional portion, we assume that the original pixel x is the least significant digit (LSD) of any numerical attributes available for watermarking, and the predicted intensity $\hat{x}$ is any value that is known by the data owner at both watermark embedding and detection stages (e.g., the tuple primary key hash value). Our reversible scheme consists of four phases: (i) preprocessing; (ii) watermark embedding; (iii) integrity verification; and (iv) data recovery. Table 1 summarizes the notations used in the rest of this paper, and the data partitioning procedure is shown in Algorithm 1.

4.1. Preprocessing. Let $R(P_k, A_1, \dots, A_y)$ be the database relation to protect. $A_1, \dots, A_y$ are all numeric attributes, and P_k is the primary key attribute whose values never change or else can be recovered. In this phase, the relation R is securely partitioned into v groups based on the primary key hash values of tuples. The number of groups is a secret parameter known only to data owner as well as the secret key K_s . For the hashing purpose, a message authentication code (MAC) is

TABLE 1: Notations and parameters.

Symbol	Description
H	Number of tuples in the relation
Γ	Number of attributes in the relation
$t_i.A_j$	j^{th} attribute of the i^{th} tuple
$t_i.P_k$	Primary key of the i^{th} tuple
h_i	Hash value of the i^{th} tuple's primary key
H_j	Hash value of the j^{th} attribute
K_s	Secret key
V	Number of groups
μ	Average number of tuples in a group
G_j	j^{th} group
E	Prediction error
e_w	Expanded prediction error
W_j	Watermark embedded in attribute A_j

```

(1) For each tuple  $t_i \in R$  do
(2)    $h_i = \text{Hash}(K_s || t_i.P_k || K_s)$  //Primary key hash
(3)    $j = h_i \bmod v$  //group index
(4)   Insert tuple  $t_i$  into group  $G_j$ 
(5) End for
      Sort all tuples in  $G_j$  in increasing order of their primary key hash
(6) Return  $(G_1, G_2, \dots, G_v)$ 

```

ALGORITHM 1: Database relation partitioning.

computed on a cryptographic hash function concatenated with the secret key in order to prevent an unauthorized person to reproduce valid hash codes from a suspicious relation. The data partitioning procedure is that of Algorithm 1. Using the property that secure hash functions generate uniformly distributed message digests, the grouping technique used approximately assigns the same number $\mu = \eta/v$ of tuples to each group. Furthermore, it is difficult for an attacker to predict the tuples-to-group assignment without the knowledge of the secret parameters. Note that the grouping and sorting operations do not physically change the positions of the tuples in the database. However, since tuples in a database are independent, these operations are necessary for enforcing some relationship between them so that the embedded and extracted watermarks can be synchronized.

4.2. Watermark Embedding. Algorithms 2 and 3 show the watermark embedding process for a given group. This process can be summarized as follows: first, for each attribute, we calculate a hash value from which we generate the attribute watermark. The number of tuples in the group determines the size of the watermark. Next, in order to reduce the distortion caused by watermark insertion, the watermark bits are securely encoded by expanding the prediction-error of the LSD of numeric attributes. We assume that such errors will not affect data usability, as they are not permanent. The encoding process (Algorithm 3) first extracts the LSD of the attribute value and then computes its

prediction error, which is expanded with the corresponding watermark bit as shown by lines 1 through 4. Afterwards, the new value of the LSD is combined with the corresponding most significant digits (MSD) to calculate the value of the watermarked attribute.

4.3. Integrity Verification and Data Recovery. The protected database relation may be subject to various types of attacks aiming at maliciously changing the content of data. Therefore, for an attack to be successful, the pirate needs to maliciously modify data, while keeping unchanged the embedded watermark. However, since the embedded watermark is computed from data characteristics using cryptographic secure hash function, any modification made to data should be correctly detected and localized as tampering. Moreover, since the scheme is reversible, the original data can be restored. The detection process is fully blind; it only requires the knowledge of some parameters such as K_s , v , and γ . As in the embedding stage, the database relation is virtually divided into secret groups (Algorithm 1). The integrity of each group is verified independently as described in Algorithms 4 and 5. First, the concealed watermark bits are extracted from the expanded prediction errors of attributes least significant digits while restoring the original data. Then, the original watermark is computed from attribute hash values in the same way as in the embedding procedure. Finally, the extracted and computed watermarks are compared. If they match, then the associated attribute is authentic, otherwise it is tampered with.

```

(1)  $H_p = \text{HMAC}(K_s || h_1 || h_2 || \dots || h_\mu || K_s)$  //Hashing of primary key hash values
(2) for each attribute  $A_j$  do
(3)    $H_j = \text{HMAC}(H_p || t_{1.A_j} || t_{2.A_j} || \dots || t_{\mu.A_j} || K_s)$  //attribute hash
(4)   //Generate watermark bits
(5)    $W_j = \mu \text{ MSB}(H_j)$  //Assume length  $(H_j) \geq \mu$ 
(6)   //Encode bit
(7)   for each tuple  $t_i$  do
(8)      $\text{Encode\_Bit}(t_{i.A_j}, W_j[i], h_i)$  //See Algorithm 3
(9)   end for
(10) end for

```

ALGORITHM 2: Watermark embedding.

```

(1)  $\text{lsd} = \text{GetLSD}(t_{i.A_j})$ 
(2)  $T = \text{GetMSD}(t_{i.A_j})$  //Most significant digits of  $t_{i.A_j}$ 
(3)  $e = \text{lsd} - h_i$  //Prediction error
(4)  $e_w = 2 * e + W_j[i]$  //expanded prediction error
(5)  $\text{newval} = e_w + h_i$  //New value of the LSD
(6)  $\text{newval} = \text{To\_number}(T, \text{newval})$ 
(7) //Update the watermarked attribute value
(8)  $t_{i.A_j} = \text{newval}$ 

```

ALGORITHM 3: Encode_Bit

```

(1)  $H_p = \text{HMAC}(K_s || h_1 || h_2 || \dots || h_\mu || K_s)$  //Hashing of primary key hash values
(2) for each attribute  $A_j$  do
(3)   for each tuple  $t_i$  do
(4)      $\text{lsd} = \text{GetLSD}(t_{i.A_j})$ 
(5)      $T = \text{MSD}(t_{i.A_j})$ 
(6)      $e_w = \text{lsd} - h_i$  //watermarked prediction error
(7)      $W_j^*[i] = \text{LSB}(e_w)$  //recovering embedded bit
(8)      $t_{i.A_j} = \text{RecoverData}(T, \text{lsd}, e_w, W_j^*[i])$  //See Algorithm 5
(9)   end for
(10)  $H_j = \text{HMAC}(H_p || t_{1.A_j} || t_{2.A_j} || \dots || t_{\mu.A_j} || K_s)$  //attribute hash
(11) //Generate watermark bits
(12)  $W_j = \mu \text{ MSB}(H_j)$  //Assume length  $(H_j) \geq \mu$ 
(13) //Check integrity
(14) if  $W_j \neq W_j^*$  then
(15)   attribute  $A_j$  not authentic
(16) end if
(17) end for

```

ALGORITHM 4: Watermark detection.

```

(1)  $e = \lfloor 31/2 \rfloor$  //original prediction error
(2)  $\text{origval} = \text{lsd} - e - W_i^*[j]$  //original value of the LSD
(3)  $\text{origval} = \text{To\_number}(T, \text{origval})$  //attribute original value
(4) return origval

```

ALGORITHM 5: RecoverData.

5. Security Analysis

In this section, we analyze the security of our reversible fragile watermarking scheme in terms of failure probability in tamper detection. We consider the following attacks: attribute value alteration, tuple insertion, and tuple deletion.

5.1. Attribute Value Alteration Attack. In this kind of attack, the attacker can either maliciously alter a single value or multiple values. If a single value is modified, obviously, only one attribute in a single group will be affected. Due to the use of secure hash function in watermark computation, the modification will randomize the corresponding attribute hash and thus the generated watermark. After the attack, since each of the μ watermark bit has equal probability $p = 1/2$ to change or not, the probability that our scheme fails to detect this tampering is given by

$$P_f = \frac{1}{2^\mu}. \quad (7)$$

It is easy to see that this failure rate is monotonic decreasing with the group size μ . The larger the group size, the lower the failure probability, and the more secure is our scheme.

Now suppose that the attacker alters multiple values. Since all values have equal probability to be modified, the attack could affect one or more groups. When all modified values are in a single group, one or more attribute watermarks could be randomized. If n ($n \leq \gamma$) attributes are affected, the probability that all watermarks in the group are correctly verified after this attack is given by

$$P_f = \frac{1}{2^{\mu n}}. \quad (8)$$

We can see that this failure rate decreases exponentially with the group size and the number of affected attributes.

5.2. Tuple Insertion Attack. The pirate can insert one or more tuples into the protected database relation. In case of a single insertion, this attack will increase the size of the group to which the new tuple is assigned to. As a result, all relevant attribute watermarks will be randomized. Since there are $\mu\gamma + \gamma$ bits in the group watermark, the error rate for this modification is

$$P_f = \frac{1}{2^{\gamma(\mu+1)}}. \quad (9)$$

Now, suppose that n tuples are inserted during this attack. The piracy could affect more than one group. If a single group is affected, the probability that our scheme fails to correctly detect the tampering is obtained by

$$P_f = \frac{1}{2^{\gamma(\mu+n)}}. \quad (10)$$

We can observe that these rates are monotonic decreasing with the group size μ , the number γ of attributes, and the number of inserted tuples n . The larger these parameters are, the lower are the failure rates in watermark

extraction, and the better is the security of the proposed scheme.

5.3. Tuple Deletion Attack. If a single tuple is deleted by the attacker, all watermarks in the relevant group will be randomized. Therefore, the probability for this piracy to succeed is

$$P_f = \frac{1}{2^{\gamma(\mu-1)}}. \quad (11)$$

Consider that n tuples are deleted from the watermarked relation. If all these tuples fall in a single group, the probability to correctly detect all relevant watermarks is given by

$$P_f = \frac{1}{2^{\gamma(\mu-n)}}. \quad (12)$$

We can easily see that the failure rate decreases exponentially with the group size and the number of attributes. However, it increases in case of massive deletion.

6. Experimental Results

In this section, the distinguished features of our scheme are discussed. We performed our algorithms in 4 Core Processors Intel Core i3-8100, 3.6 GHz with a 4 GB RAM computer running SQL Server 2012 and the **Microsoft Visual Studio C# IDE**. The operating system used in this experiment was Windows 7 Professional 64 Bit. We have used the forest cover type for 30×30 meter cells obtained from US Forest Service (USFS) Region 2 Resource Information System (RIS) data [31] in our experiment. It consists of 581,012 tuples and 54 attributes with numeric and nonnumeric data. For watermark insertion, we used the 10-first integer attributes and added a new attribute called *id* to serve as a primary key.

The experiment consists of two steps listed in the following: (i) imperceptibility and usefulness analysis and (ii) tamper detection and data recovering.

6.1. Imperceptibility and Data Usefulness. Table 2 shows the impact of watermark insertion on the mean and variance of marked attributes. The values are rounded to the nearest integer. We can see that the mean remains unchanged except for attribute 4, where it is increased by 1. Though the change in the variance is higher than in the mean, the errors are still minor. Furthermore, the distortion caused is not permanent since the scheme is reversible.

6.2. Tamper Detection and Data Recovering. For efficiency reasons, we used a sample of 1,000 tuples as an experimental dataset. We tested the capability of our scheme to detect maliciously modified groups and restore original data from nontampered groups. For each attack, we analyzed the relevant detection probability which is the complementary of the corresponding failure probability. Each test was performed 10 times, and the results were averaged.

We first observed that, in case of multiple tuple insertion attack, the affected groups were completely detected. This is

TABLE 2: Embedding impact in mean and variance of watermarked attributes.

Attribute	Before embedding		After embedding	
	Mean	Variance	Error in mean	Error in variance
1	2959	78391	0	+1
2	156	12525	0	0
3	14	56	0	0
4	269	45177	+1	0
5	46	3398	0	0
6	2350	2431272	0	-12
7	212	717	0	0
8	223	391	0	0
9	143	1465	0	0
10	1980	78391	0	+1

FIGURE 2: Detection probability for attribute alteration attack.

FIGURE 3: Detection probability for tuple deletion attack.

because the cryptographic hashing function used (SHA-1) uniformly distributed the new tuples into the groups according to the value of g , thus randomizing each relevant watermark.

The results for the alteration of multiple values are shown in Figure 2. We can observe that all modified groups are correctly detected, except when $g=90$ where a few groups are missed. The reason is that the detection probability is low when the number of groups is large. This observation confirms our theoretical results.

Figure 3 shows the detection probability in case of multiple tuple deletion. We can easily see that for values of g ranging from 10 to 50 and a deletion rate of up to 90%, the detection probability is equal or greater than 0.88. However, with smaller groups, as the deletion rate increases, the detection probability decreases considerably. For instance, when $g=90$ and the deletion rate is equal to 90%, the detection probability is only 0.1. The reason is that, after such an attack, it is highly possible that most of the groups were completely deleted.

TABLE 3: Comparison with previous works.

	Farfoura et al. [29]	Chang et al. [30]	Proposed scheme
Data type	Numeric	Numeric	Numeric
Watermark information	Image	Data characteristics	Data characteristics
Distortion control	Fractional part of numeric attributes is used for watermark embedding	2 last digits from numeric attributes are used for watermark embedding	LSD of numeric attributes is used for watermark embedding
Data recovery	Yes	Yes	Yes
Tamper detection and localization	No	No	Yes at group level

Regarding data recovery, due to the use of prediction-error expansion, we noticed that no matter the attack, the scheme was able to restore 100% of the original data from authentic groups.

Table 3 shows a comparison between our technique and some recent related works, both based on PE. From the best of our knowledge, so far, a reversible fragile scheme based on PE has not been proposed in the context of relational databases. We consider different aspects, including the watermark information, the watermark encoding, the data recovery, and the granularity level of tamper detection and localization. We can easily see that, unlike other methods, our scheme can detect and localize malicious modification. In addition, our technique minimizes the distortion caused by the embedding process, while Farfoura et al.'s method [29] is not suitable for numeric data without fractional part, and Chang et al. [30] considered two last digits for marking.

7. Conclusion

In this study, a new reversible fragile watermarking approach is developed for tamper detection of relational databases. To preserve data usability, the watermark is encoded by expanding the prediction-error of the LSD of numeric attributes. Through theoretical analysis and experiments, we demonstrated the effectiveness of the proposed technique to detect and localize malicious modifications from various attacks. In addition, the original data can be fully recovered from nontampered groups. In the future, we will extend our solution to nonnumeric, object-oriented, and Extensible Markup Language (XML) data.

Data Availability

The data related to the manuscript can be freely made publicly available without any constraints.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] V. Khanduja, "Database watermarking, a technological protective measure: perspective, security analysis and future directions," *Journal of Information Security and Applications*, vol. 37, pp. 38–49, 2017.
- [2] V. Khanduja, S. Chakraverty, and O. P. Verma, "Ownership and tamper detection of relational data: framework, techniques and security analysis," in *Gate to Computer Science and Research*, pp. 21–36, Science Gate Publishing, Thrace, Greece, 2016.
- [3] M. Kamran and M. Farooq, "A comprehensive survey of watermarking relational databases research," 2018, <https://arxiv.org/abs/1801.08271>.
- [4] J. Franco Contreras and G. Coatrieux, "Protection of relational databases by means of watermarking: recent advances and challenges," *Advances in Security in Computing and Communications*, pp. 101–123, BoD-Books on Demand, Norderstedt, Germany, 2017.
- [5] S. Iftikhar, M. Kamran, and Z. Anwar, "A survey on reversible watermarking techniques for relational databases," *Security and Communication Networks*, vol. 8, no. 15, pp. 2580–2603, 2015.
- [6] A. Hamadou, X. Sun, L. Gao, and S. A. Shah, "A fragile zero-watermarking technique for authentication of relational databases," *International Journal of Digital Content Technology and Its Applications*, vol. 5, no. 5, pp. 189–200, 2011.
- [7] L. Camara, J. Li, R. Li, and W. Xie, "Distortion-free watermarking approach for relational database integrity checking," *Mathematical Problems in Engineering*, vol. 2014, Article ID 697165, 10 pages, 2014.
- [8] L. Camara, D. Coulibaly, H. Ali, and J. Li, "An effective approach for non-numeric relational database verification," *International Journal of Database Theory and Application*, vol. 10, no. 6, pp. 35–46, 2017.
- [9] W. Yaqub, I. Kamel, and Z. Aung, "Distortion-free Watermarking Scheme for Compressed Data in Columnar Database," in *Proceedings of the International Conference on Security and Cryptography*, pp. 343–353, Amalfi, Italy, January 2018.
- [10] R. Hou, H. Xian, X. Wang, and J. Li, "A robust reversible watermarking scheme for relational data," in *Security and Privacy in New Computing Environments SPNCE 2019-Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, J. Li, Z. Liu, and H. Peng, Eds., Vol. 284, Springer, Berlin, Germany, 2019.
- [11] D. Hu, D. Zhao, and S. Zheng, "A new robust approach for reversible database watermarking with distortion control," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 6, pp. 1024–1037, 2019.
- [12] Y. Li, J. Wang, S. Ge, X. Luo, and Bo Wang, "A reversible database watermarking method with low distortion," *Mathematical Biosciences and Engineering*, vol. 16, no. 5, pp. 4053–4068, 2019.
- [13] V. Khanduja, O. P. S. Chakraverty, and O. Verma, "Enabling information recovery with ownership using robust multiple watermarks," *Journal of Information Security and Applications*, vol. 29, pp. 80–92, 2016.

- [14] A. Khan, A. Siddiqua, S. Munib, and S. A. Malik, "A recent survey of reversible watermarking techniques," *Information Sciences*, vol. 279, pp. 251–272, 2014.
- [15] Y. Zhang, X. Niu, and B. Yang, "Reversible watermarking for relational database authentication," *Journal of Computers*, vol. 17, no. 2, pp. 59–66, 2006.
- [16] G. Coatrieux, E. Chazard, R. Beuscart, and C. Roux, "Lossless watermarking of categorical attributes for verifying medical data base integrity," in *Proceedings of the Annual International Conference of the IEEE EMBS*, pp. 8195–8198, IEEE, Boston, MA, USA, September 2011.
- [17] H. Khataeimaragheh and H. Rashidi, "A novel watermarking scheme for detecting and recovering distortions in database tables," *International Journal of Database Management Systems*, vol. 2, no. 3, pp. 1–11, 2010.
- [18] H. Guo, Y. Li, A. Liu, and S. Jajodia, "A fragile watermarking scheme for detecting malicious modifications of database relations," *Information Sciences*, vol. 176, no. 10, pp. 1350–1378, 2006.
- [19] J. Franco-Contreras, G. Coatrieux, N. Cuppens-Boulahia, F. Cuppens, and C. Roux, "Robust lossless watermarking of relational databases based on circular histogram modulation," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 3, pp. 397–410, 2014.
- [20] J. Franco Contreras, G. Coatrieux, E. Chazard, F. Cuppens, N. Cuppens Boulahia, and C. Roux, "Robust lossless watermarking based on circular interpretation of bijective transformations for the protection of medical databases," in *Proceedings of the 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 5875–5878, IEEE, San Diego, CA, USA, August 2012.
- [21] V. Khanduja and S. Chakraverty, "Fragile watermarking of decision system using rough set theory," *Arabian Journal for Science and Engineering*, vol. 43, no. 12, pp. 7621–7633, 2018.
- [22] U. K and P. K v, "Robust optimal position detection scheme for relational database watermarking through HOLPSOFA algorithm," *Journal of Information Security and Applications*, vol. 35, pp. 1–12, 2017.
- [23] J. -N. Chang and H.-C. Wu, "Reversible fragile database watermarking technology using difference expansion based on SVR prediction," in *Proceedings of the International Symposium on Computer, Consumer and Control*, pp. 690–693, Taichung, Taiwan, June 2012.
- [24] L. Gao, D. Wang, and A. Hamadou, "New fragile database watermarking scheme with restoration using reed-solomon codes," *Journal of Computational and Theoretical Nanoscience*, vol. 10, no. 1, pp. 147–153, 2013.
- [25] G. Gupta and J. Pieprzyk, "Reversible and blind database watermarking using difference expansion," *International Journal of Digital Crime and Forensics*, vol. 1, no. 2, pp. 42–54, 2009.
- [26] R. Agrawal and J. Kiernan, "Watermarking relational databases," in *Proceedings of the 28th International Conference on Very Large Data Bases (VLDB '02)*, pp. 155–166, Hong Kong, China, August 2002.
- [27] M. Farfoura and S. Horng, "A novel blind reversible method for watermarking relational databases," in *Proceedings of the IEEE International Symposium on Parallel and Distributed Processing with Applications*, pp. 563–569, IEEE, Taipei, Taiwan, September 2010.
- [28] D. M. Thodi and J. J. Rodriguez, "Prediction-error based reversible watermarking," in *Proceedings of the 2004 International Conference on Image Processing*, pp. 1549–1552, IEEE, Singapore, October 2004.
- [29] M. E. Farfoura, S.-J. Horng, J.-L. Lai, R.-S. Run, R.-J. Chen, and M. K. Khan, "A blind reversible method for watermarking relational databases based on a time-stamping protocol," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3185–3196, 2012.
- [30] C.-C. Chang, T.-S. Nguyen, and C.-C. Lin, "A blind reversible robust watermarking scheme for relational databases," *The Scientific World Journal*, vol. 2013, Article ID 717165, 12 pages, 2013.
- [31] <http://kdd.ics.uci.edu/databases/coverttype/coverttype.html>.