A Robust and Adaptive Watermarking Technique for Relational Database

Yexing Zhang¹, Zihou Wang⁴, Zhaoguo Wang^{2,3(✉)}, and Chuanyi Liu^{2,3}

¹ School of Cyberspace Security, Beijing University of Posts
and Telecommunications, Beijing 100876, China

² School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen),
Shenzhen 518055, China
wangzhaoguo@hit.edu.cn

³ HIT-QIANXIN Data Security Research Institute, Shenzhen 518055, China

⁴ National Computer Network Emergency Response Technical Team/Coordination Center of
China, Beijing 100029, China

Abstract. A watermarking technology is a kind of marker covertly embedded to identify ownership of the copyright. The existing database watermarking techniques cannot automatically adapt to different types of data and have poor robustness. In this paper, we present a new robust database watermarking scheme. The scheme can automatically adapt the watermarking algorithm and parameters according to the data characteristics for numerical and text-based data. We have theoretically established and verified experimentally the performance of our method in terms of robustness and less data distortion. This makes it suitable for copyright protection, owner identification, or traitor tracing purposes.

Keywords: Relational data · Database watermarking · Copyright protection · Automatic adaptation · Safe and reliable

1 Introduction

In recent years, various industries have emerged with complex and diverse data, and these data have become important production factors [1]. As the value of data is increasing, people are more and more concerned about copyright protection, and the awareness of copyright protection is gradually increasing. Relational data is one of the most common types of data, therefore, the copyright protection of relational data has become one of the hot spots of research in related fields.

In the early 1990s, digital watermarking techniques were proposed by related researchers. As an important branch of information hiding technology, digital watermarking is an effective method to achieve copyright protection [2]. In 2002, Agrawal and Kiernan et al. proposed the first data watermarking scheme for relational data by combining the characteristics of relational data [3]. Since then, researchers in related fields have started their research work on data watermarking. Data watermarking refers to embedding the identity information of the owner who constitutes the data into the

structured data, and the embedded information should satisfy the data with less distortion. When the original data is illegally used or maliciously copied and disseminated, the database watermarking extraction technology can be used to extract the original watermark from the embedded data, thus confirming the copyright of the data and tracing the responsibility of the leakage.

The relational data can be commonly classified into numerical data and textual data. It has the characteristics of low data redundancy and frequent data updates, so the distortion reduction and attack resistance have become a difficult problem and research hotspot in the industry. The numerical data watermarking algorithms include Least Significant Bit (LSB) algorithm [3], difference expansion algorithm [4, 5], prediction-error expansion algorithm [6, 7], and histogram shifting algorithm [8–10], etc. The text-based data watermarking algorithms mainly contain natural language-based text watermarking algorithms [11–16], invisible coding algorithms [17–20], and punctuation coding algorithms [21], etc. The current research progress that there is the lack of algorithms for multiple data types and the automatic adaptation of parameters, as well as the lack of evaluation of the effectiveness of these algorithms in terms of distortion and robustness.

In this paper, we propose for the first time a robust and adaptive relational data watermarking scheme for numeric and text-based data, which can automatically adapt the variable row embedding ratio and selected embedding column of data watermark according to the data characteristics, in addition to the data watermark embedding/extraction algorithm and parameters through data type analysis, data volume evaluation, data sensitivity analysis, automatic parameter setting, and result visualization mechanism techniques. It has better robustness against common attacks such as data watermark addition, modification, and deletion attacks. It can effectively cope with watermark erasure and attack detection. It can process numerical and text-based data simultaneously. With the distortion effect and various robustness experiments, it has been proved that the scheme has less data distortion and higher robustness, which can be used as a data watermarking scheme.

The main innovation points of this paper are as follows.

1. A robust and adaptive watermarking technique for a relational database is proposed, which can effectively solve the problem of data copyright validation and leakage traceability.
2. For common numerical and text-based data, an automatic adaptation watermarking algorithm is proposed for the first time, which combines data type adaptation, data volume evaluation, data column sensitivity judgment, parameter tuning, and result visualization to achieve intelligent adaptation of database watermarking algorithm and model parameters.
3. With the data distortion and robustness experiments, it is shown that the watermarking algorithm with automatic adaptation and parameter tuning proposed in this paper has better robustness and usability.

2 Related Work

In contrast to multimedia data, the relational database mainly stores structured data, which has the characteristics of low data redundancy and frequent data updates [22].

Therefore, on the basis of multimedia data, a series of database watermarking schemes have been proposed by researchers in related fields, combining the characteristics of structured data.

In 2002, Agrawal and Kiernan used a hash function with a key to pick specific tuples and attribute values from the data and then embed special values in the least significant bits of the attribute values [3]. These special values constitute the database watermark and it doesn't affect the normal use of the data. In 2004, a database watermarking technique based on secret sorting and grouping was proposed by R. Sion et al. to perform watermark embedding by changing the distribution of data in each grouping [23]. The algorithm reduces the effect of watermark embedding on data distortion and improves the resistance to subset modification attacks and subset addition attacks. However, it has poor robustness to subset deletion attacks. In 2003, Niu Xiamu et al. proposed a watermarking scheme capable of embedding a small number of meaningful strings in a database. This algorithm is based on parity matching for embedding the watermark, which allows the primary key hash to be paired with the least significant bit. The presence of this matching relationship is verified when detecting the watermark [24]. However, it is only possible to verify whether such matching rules are embedded in the data, but it is not possible to detect the real watermark information.

With the purpose of reducing the distortion caused by database watermarking techniques on data, a series of reversible techniques for structured data have been proposed by scholars. Currently, the common reversible database watermarking algorithms include difference expansion [4, 5], prediction-error expansion [6, 7], and histogram shifting [8–10]. (1) The difference expansion refers to picking an attribute value pair from a specific tuple and then implementing watermark embedding and data recovery by performing a specific numerical transformation on that attribute value pair [4]. In 2008, Gupta et al. proposed a scheme in which reversibility of watermarking was achieved using a difference expansion technique [4], which increased the capacity of watermark embedding and enabled more watermark information to be embedded in the original data, but the scheme was not specified for relational data. In 2013, Jawad et al. proposed a difference expansion database watermarking method based on genetic algorithm [5]. This method improves the robustness of database watermarking algorithm and reduces data distortion. (2) The prediction-error expansion technique achieves reversibility of the watermark by using a prediction algorithm which obtains the predicted value, and then selects a certain attribute value from the original data for which a numerical transformation similar to the difference expansion is performed. In contrast to difference expansion, this technique only requires modifying the value of an attribute in a tuple and it has less effect on data availability. In 2004, Thodi et al. proposed a watermark embedding process by introducing a prediction-error technique to implement image watermarking, in which the main idea is to embed the watermark into the difference between image pixel points [6]. In 2012, Farfoura et al. transformed recognizable images into bit streams embedded in the least significant bits of numerical attributes and achieved watermark reversibility by prediction-error expansion [7]. This method is mainly for floating-point type data and can be used for data tampering detection. However, it does not work for handling integer type data. (3) Histogram shifting is required to first calculate the differences of some attribute values in the data, and use the first non-zero number of these differences to

construct a histogram, which is then used to change the distribution characteristics of the non-zero number according to certain rules to achieve the embedding of the watermark [8]. When data is recovered, it is sufficient to restore the data distribution characteristics of the number. This method can track the degree of data distortion, but it is difficult to resist high-intensity attacks. In 2006, Zhang et al. proposed a reversible watermarking scheme by calculating the difference of attribute values to construct a histogram, and then using histogram expansion technique to achieve reversible watermarking, which increases the watermarking capacity [8]. In 2018, Hu et al. proposed a watermarking method combining genetic algorithm and histogram shifting prediction-error techniques, which used a genetic algorithm to generate optimal watermark information, and calculated a histogram of prediction-error for candidate attributes, which was then used to embed the watermark into the peak of the histogram using histogram expansion techniques [9]. The method reduces the distortion of the data and improves the robustness against various watermarking attacks. In 2019, Li et al. proposed a low-distortion digital watermarking method for Hu's watermarking scheme, which improved the method of selecting the embedding watermark position in Hu's watermarking scheme as well as changed the histogram shifting direction, which was experimentally shown to achieve lower data distortion [10]. In 2014, a reversible watermarking scheme proposed by Iftikhar et al. used the concept of mutual information in information theory to select the embedding position of the watermark and obtained the optimal watermark by genetic algorithm to reduce the data distortion [25]. This method is highly robust and can extract the embedded watermark information and recover the original data even in the face of large-scale watermarking attacks. However, the acquisition of the optimal watermark requires a large amount of computation, which is less efficient in the face of massive data. In 2015, Tong Deyu et al. applied the database watermarking technique to GIS by embedding the interval location value and watermark together in the database, which ensures the correspondence between the watermark and the watermark location and improves the correct rate of watermark detection [26]. The embedding and extraction of this watermarking algorithm were not dependent on the primary key, which could still extract the watermark information even if the primary key was attacked. In 2019, Wang Chundong et al. classified the existing techniques mainly using whether distortion is introduced to the underlying data, focusing on the watermark generation methods and embedding methods of several typical watermarking schemes and in which they compared the attacks and application scenarios that each type of scheme can cope with [27].

In recent years, the database watermarking technique based on non-numeric data has also been further developed. In 2004, a watermark embedding by attribute value replacement method was first proposed by Sion R et al. to determine whether a watermark is embedded based on the relationship between primary key values and non-numeric attribute values [28]. In 2010, Hanyurwimfura et al. proposed a non-numeric relational database watermarking method, which first dynamically selects tuples and attributes in the watermark embedding phase, and then uses an edit distance algorithm to move the horizontal position of a word according to the watermark bit thus achieving watermark embedding [29]. In 2015, by constructing two different watermark embedding mechanisms, Melkundi S et al. were able to embed watermarks in text-based and numerical

data, respectively. For text-based data, in case of embedding a binary number 1, an invisible Unicode character is added to the text data and not added if a 0 is embedded; for numerical data, the watermark is embedded through the lowest significant bit of the data [30].

Comprehensive the above watermarking schemes can be seen that only the literature [30] considered embedding watermarks into textual and numerical data separately, which did not consider the problem of automatic data type adaptation. Therefore, for this problem, a new robust and adaptive database watermarking scheme is proposed in this paper.

3 Scheme

The proposed scheme contains three parts according to data flow: (1) watermark pre-processing; (2) watermark embedding; and (3) watermark extraction. In which, the adaptive algorithm of watermark embedding contains (1) data type adaptation; (2) data volume evaluation; (3) data column sensitivity judgment; (4) automatic parameter setting; (5) result visualization mechanism in five parts. In the pre-processing stage, the processing is mainly done for copyright information. Before the embedding stage of the watermark, data type adaptation, data volume evaluation, data column sensitivity judgment, automatic setting of parameters and other operations are performed, and then by designing and using five database watermarking algorithms to embed/extract the watermark, and the final visualization results are fed back to the user, which solved the problem that the data cannot be automatically adapted in the current relational database watermarking scheme. The system architecture of the scheme is shown in Fig. 1.

Fig. 1. System structure.

To verify the effectiveness of this scheme, theoretical analysis, and practical experiments were used, and the relevant symbols used in the scheme and their meanings are shown in Table 1.

Table 1. Symbols and meanings in the scheme.

Symbols	Meaning	Symbols	Meaning
D	Original relational dataset	D_{AW}	D_W after the attack
D_W	Relational datasets with embedded watermarks	γ	Secret Parameters Collection
K_S	Key Collection	N_g	Number of data groupings
X_{max}	Hide function maximum list	X_{min}	Hide function minimum list
r	Tuple	$r.P$	Primary key value of tuple r
W	Watermark	ξ	Minimum number of members
T^*	Threshold of the optimization algorithm	G_i	Constraint set
S_i	Single data subgroup	$text$	Text to be embedded
bit_w	Watermark Bits	$text^*$	Complete the text of the marked-up sentences
l	Watermark length	W_d	Detected watermark
$Similarity$	Watermark similarity	$message$	Copyright information of the data owner

3.1 Pre-processing Stage

The specific tasks of the pre-processing phase include (1) the selection of the fields to be embedded and the key generation, (2) the watermark generation, and (3) the data grouping.

Selection of Fields to be Embedded and Key Generation. For the selection of fields to be embedded, the selection of attribute fields to be embedded is performed in a semi-automatic manner, i.e., the user can independently select the columns in which the attributes to be embedded are located. In terms of key generation, the specified length of key is generated based on the specified range of key characters and combined with the database and data table names.

Watermark Generation. Before the watermark is embedded, it is necessary to convert the watermark information to format for embedding into the data. In this paper, Chinese and English strings containing meaning are selected as watermark information and then embedded in the format of binary sequences. Therefore, first of all, the watermark

information message should be converted into a binary sequence, which is defined as shown in Eq. (1).

$$BinarySequence = Binary(message) = \{b_1, b_2, \dots, b_i, \dots, b_l | b_i \in \{0, 1\}\} \quad (1)$$

where *message* represents the watermark information, and *BinarySequence* represents the ASCII value corresponding to the *message*, i.e. the data stream in binary format, $\{b_1, b_2, \dots, b_i, \dots, b_l\}$ is a set of binary numbers, The length of the sequence *BinarySequence* is the sum of the binary bits occupied by all elements in the *BinarySequence*, i.e. $l_b = len(\{b_1, b_2, \dots, b_i, \dots, b_l\})$.

Data Grouping. Before embedding the watermark information, it is also necessary to group the data in the database. With data grouping, a number of non-intersecting subgroups can be created in the database. The database can be grouped using Eq. (2) to obtain the number of each data subgroup in the database.

$$partition_i = H(K_S || H(K_S || r_i.pk)) \bmod N_g \quad (2)$$

In which, $partition_i$ denotes the number of each data subgroup after grouping, $partition_i = 1, 2, 3, \dots, N_g$, N_g is the number of data subgroups; $H(\cdot)$ is the hash function. The hash function is used for the key (K_S) and the primary key ($r_i.pk$) of the database tuple, which can ensure the grouping is more secure. The “||” represents the string concatenation operation and *mod* represents the remainder operation.

The detailed process of the data grouping algorithm, as shown in Fig. 2.

Fig. 2. Data grouping algorithm.

3.2 Data Type Adaptation

Data type adaptation, which focuses on the automatic adaptation of fields in the data table before the watermark is embedded.

In this section, the proposed data type adaptation module only considers the adaptation for numeric and string (character) type data. The detailed adaptation process is shown in Fig. 3.

Fig. 3. Data type adaptation flow chart.

In Fig. 3, all the fields in the data table except the primary key and date/time are traversed one by one. If each row of data values of the field c in the data table satisfies the regular expression `"-?[0-9]+(\\.[0-9]+)?"`, then the field c is a numeric field; otherwise, the field c is a string(character) type field. The same determination operation is performed for the next field in the data table until all fields in the data table are traversed.

After all the fields in the data table except the primary key and date/time are traversed, this module can create a “field data type management registry”, according to which the user can quickly make a judgment on the data type of each field in the data table, and then choose the corresponding watermark embedding process.

3.3 Data Volume Evaluation

Data volume evaluation which focuses on evaluating the data volume of the original data table before watermark embedding work. For each original data table, by default the first field in the table is the primary key, and if there is no primary key in the data table, it will generate a new field for the data table as the primary key. Therefore, the last row of the tuple in the data table has the primary key value which is the total number of data volumes in the data table.

After the total amount of data in the data table is determined, the module can determine the number of data group divisions N_g for this data table based on the total number of tuples sum in the data table and the data grouping control parameter λ entered by the user, which is calculated as shown in Eq. (3).

$$N_g = \frac{sum}{\lambda} \quad (3)$$

With Eq. (3), the number of divisions that meet the user's expectations can be calculated.

After determining the number of data groupings N_g , the user can specify the proportion of rows embedded in the watermark μ , then the section will randomly select $fetch_count$ rows of data in the data table to embed the watermark. Where $fetch_count$ is the number of tuples that will be embedded in the watermark, which is calculated as shown in Eq. (4).

$$fetch_count = sum \times \mu \quad (4)$$

When the user does not specify the specific value of μ , the proportion of rows embedded in the watermark μ defaults to 100%.

3.4 Data Column Sensitivity Judgment

Data column sensitivity judgment, which is mainly based on the sensitivity to determine whether the data column in the data table is more important before the watermark embedding, and then determine whether the data column needs to be watermarked embedded, to achieve an intelligent selection of the column to be embedded.

It is assumed that the data table contains multiple tuples, each of which has the same data schema and is all $R = (pKey, C_1, C_2, \dots, C_n, fKey)$, where $pKey$ represents the primary key, $fKey$ represents the foreign key, and $C_1, C_2, \dots, C_n$ represents the field columns. The primary key, the foreign key and the fields with unique constraints are grouped into the core field column set, denoted by *Core*; the non-sensitive numeric fields and text-based fields are grouped into the optional field column set, denoted by *Select*. In which, the non-sensitive numeric fields that is not sensitive to small changes in the value of the feeling (modifiable) fields, such as length, coordinates, weight and other fields.

In order to ensure the integrity of the data table, by default for the primary key $pKey$, the foreign key $fKey$ does not perform the operation of watermark embedding.

If the number of embeddable fields in the data table is Emb , the relationship between Emb and *Core* and *Select*, as shown in Eq. (5).

$$Emb = Size(Select + Core) - 2 \quad (5)$$

When the Emb value is determined, the watermark is not embedded under every embeddable field, but the watermark embedding field is selected jointly based on multiple information such as user information.

The sensitivity of the embeddable fields in the data table is set to S_i , and the corresponding sensitivity list of the data table is $\{S_1, S_2, \dots, S_{Emb}\}$. In which, the sensitivity is calculated as shown in Eq. (6).

$$S_i = \phi(\delta_1 \times \omega_1 + \delta_2 \times \omega_2 + \dots + \delta_n \times \omega_n) \quad (6)$$

In which, δ_1 denotes the proportion of null and abnormal values in the field, δ_2 denotes the number of valid bits of data values in the field, ..., δ_n denotes the watermark content. $\{\omega_1, \omega_2, \dots, \omega_n\}$ denotes the proportion of weights occupied by $\{\delta_1, \delta_2, \dots, \delta_n\}$ respectively, which are automatically assigned by this module. ϕ denotes the multi-information joint selection function, and S_i is obtained by mapping the result values of the δ_i and ω_i parameters after performing the relevant operations on them.

Finally, according to the sensitivity list $\{S_1, S_2, \dots, S_{Emb}\}$, the fields to be embedded are sorted from largest to smallest, which makes it possible to intelligently select the field columns to be embedded by this module, and then the corresponding watermark embedding process can be selected.

3.5 Automatic Parameter Setting

The automatic setting of parameters, it is mainly before the embedding of watermark, for numerical data, combined with the data type adaptation, data volume evaluation, data column sensitivity judgment, and other results information described in the previous section, and by pre-setting a number of limit parameters, so that the data value after embedding the watermark, limited to a controlled range. Thus, it can avoid excessive errors before and after embedding the watermark.

The details include (1) the control of precision, (2) the control of variation range, and (3) the control of embedded copyright information.

The control of precision, it is mainly in the watermark embedding before the user set the number of valid digits of the data value to be embedded in the watermark field. Thus, after the embedding of the watermark, it can control the degree of display of the data value. This list of specific precision control parameters is: *Precision_List* = {"10", "1", "0.1", "0.01", "0.001"}, as shown in *Precision_List*, which gradually refines the granularity of its precision control parameters.

The control of variation range, it is mainly before the embedding of the watermark, the user specified by the data value to be embedded in the watermark field of the upper and lower boundaries of the variation range (e.g. -100-100). Thus, after the watermark is embedded, it can control the approximate range where the data values are located.

The control of embedded copyright information is mainly to prove the original owner of the data by the user inputting the copyright information with actual meaning in Chinese and English, to enter the corresponding watermark embedding process (Table 2).

3.6 Watermark Embedding Stage

Table 2. The comparison of database watermarking algorithms in the watermark embedding stage

Algorithm name	Algorithm core ideas	Supported data types	Statistical distortion analysis	Robustness analysis
Optimal watermarking algorithm based on pattern search	For a given grouping S_i containing m data tuples, it is considered as an m -dimensional vector. To find an m -dimensional vector Δ_i , add it to the S_i vector to obtain the Sim vector, which can be considered as the grouped Sim after the watermark is added	Integer, floating-point	In comparison with the LSB modification algorithm, the percentage change in both mean and variance is smaller	General
LSB modification algorithm	A one-way hash function is used to make the selection of attributes and tuples to be embedded, and then their least significant bits are set to 0 or 1	Integer, floating-point	In comparison with the pattern search algorithm, the percentage change in both mean and variance is larger	Better
Space embedding algorithm	When the original watermark bit is 0, it is 2 spaces before and 2 spaces after the sentence; when the original watermark bit is 1, it is 4 spaces before and 1 space after the sentence	Text type, string type	–	General
Symbol modification algorithm	When the original text has punctuation marks, the end symbols are deleted when the specified conditions are met; when the original text does not have punctuation marks, the end symbols are appended when the specified conditions are met	Text type, string type	–	Better

(continued)

Table 2. (continued)

Algorithm name	Algorithm core ideas	Supported data types	Statistical distortion analysis	Robustness analysis
Lexical inverse ordinal number algorithm	If the inverse number is odd, the watermark bit after encoding is set to 1. If the inverse number is even, the watermark bit after encoding is set to 0	Text type, string type	–	Better

Numerical Database Watermarking Algorithm. (1) Optimal watermarking algorithm based on pattern search: The algorithm starts from the initial base point and implements two types of searches alternately: axial search and pattern search. The axial search is performed sequentially along the direction of the n coordinate axes, and is used to determine new iteration points and directions that favor the decrease of function values. The pattern search, on the other hand, is performed along the direction of the line connecting two adjacent iteration points in an attempt to make the function values fall faster [31]. (2) LSB modification algorithm: It is assumed that the attribute values of the tuples in the database are modified imperceptibly to the user and do not affect the availability of the data, then the least significant bit can be modified for watermark embedding. This algorithm is mainly for numerical relational databases which uses a one-way hash function to select the attributes and tuples to be embedded, and then sets the least significant bit to 0 or 1. In this way, the embedding process of the watermark is completed and does not affect the normal use of the data [3].

Text-Based Database Watermarking Algorithm. (1) Space embedding algorithm: the watermark information is hidden in the line break of text data, and the presence or absence of spaces and tabs at the end of the line or between words is visually difficult to detect, which can be decoded by the presence or absence, type and number of invisible codes during watermark extraction [20]. (2) Symbol modification algorithm: in contrast to text, people are not sensitive to punctuation marks, and therefore the possibility of embeddable watermarks exists. The substitution, deletion or addition of major punctuation marks in Chinese and English is generally unnoticeable [21]. (3) Lexical inverse ordinal number algorithm: After natural language sentences are processed by word division and lexical annotation, a lexical token sequence is obtained, which is composed of a finite number of lexical tokens. The inverse order number of the sequence is calculated by the biased order relationship defined in advance on the lexical token set, and the 1-bit information is hidden according to the parity of the inverse order number. If the current parity does not match with the hidden information bits, the lexical token sequence is modified by transformations such as swapping the positions of two symbols, adding or removing some symbols. Finally, the natural language sentence is modified according to the lexical token sequence [16].

3.7 Watermark Extraction Stage

Numeric database watermarking algorithm performs watermark extraction based on the corresponding Optimal watermarking algorithm based on pattern search and LSB modification algorithm. The text-based database watermarking algorithm performs watermark extraction on based on the corresponding space embedding algorithm, symbol modification algorithm, and lexical inverse ordinal number algorithm. The corresponding specific analysis processes, respectively, are shown in Table 3.

Table 3. The comparison of database watermarking algorithms in the watermark extraction stage

Algorithm name	Algorithm core ideas	Supported data types	Statistical distortion analysis	Robust-ness analysis
Optimal watermarking algorithm based on pattern search	According to the optimal threshold T calculated during the watermark embedding process, and using the majority voting mechanism, it is found the value of the watermarked bits embedded in this data subgroup. Then, the splicing is performed to calculate the original watermark information	Integer, floating-point	In comparison with the LSB modification algorithm, the percentage change in both mean and variance is smaller	General
LSB modification algorithm	It corresponds to the inverse process of the watermark embedding algorithm	Integer, floating-point	In comparison with the pattern search algorithm, the percentage change in both mean and variance is larger	Better
Space embedding algorithm	It corresponds to the inverse process of the watermark embedding algorithm	Text type, string type	–	General
Symbol modification algorithm	It corresponds to the inverse process of the watermark embedding algorithm	Text type, string type	–	Better
Lexical inverse ordinal number algorithm	It corresponds to the inverse process of the watermark embedding algorithm	Text type, string type	–	Better

3.8 Result Visualization Mechanism

The result visualization mechanism which is mainly for the data table after the watermark embedding/extraction, adding the result visualization mechanism. In this way, users can

quickly feel the difference in accuracy control and statistical distortion of the data table before and after watermark embedding/extraction.

When the watermark embedding/extraction is complete, the specific information displayed by the module is shown in Fig. 4.

Fig. 4. Result visualization mechanism.

Finally, it will make an exhaustive evaluation of the current watermark embedding/extraction results based on the results shown in Fig. 4 and a series of pre-set thresholds, which will eventually be fed back to the user.

4 Experimental Analysis

This section focuses on the database watermarking algorithms designed and used earlier, to further analyze the characteristics of various algorithms, and to conduct data distortion analysis experiments and robustness experiments. The experimental environment is: (1) Processor: 2.6 GHz hexa-core Intel Core i7; (2) Memory: 16 GB 2667 MHz DDR4; (3) Operating system: macOS Catalina version 10.15.7. All algorithms are implemented with Java language, and the experimental data sets are as follows (1) Integer dataset: German credit risk dataset [32]; (2) Floating-point dataset: Dow Jones Industrial Average (DJIA) stock dataset [33]; (3) Text-based dataset: Reddit WorldNews Channel historical news headlines dataset [33]. In which, (1) for the integer dataset, set the number of fields to 5 and the number of tuples to 1000; (2) select and transform the floating-point dataset, set the number of fields to 8 and the number of tuples to 1000; (3) for the text-based dataset, set the number of fields to 3 and the number of tuples to 1000.

4.1 Invisibility Analysis Experiments

To check the effect of statistical distortion on the data before and after watermark embedding, the invisibility analysis experiments are designed for numerical data in this section.

In this section, the effect of watermarking on the database is analyzed by applying the Optimal watermarking algorithm based on pattern search (OPS) and LSB modification algorithm (LSB) to integer data and floating-point data, respectively, and then three fields are selected to calculate the change values of the data before and after watermark

embedding, respectively. The two statistics, mean and variance, are selected in this section of the experiment to measure the effect of watermarking on the database. Finally, a comparison is made between them, and the final results are shown in Table 4. It can be seen from Table 4 that the change of the mean and variance of the data by the embedding of the watermark is relatively small and is basically invisible to the user. Therefore, it is clear from the experimental results that both algorithms have some invisibility.

Table 4. The comparison of numerical database watermarking algorithms - the statistical distortion results before and after database watermark embedding

Algorithm and data type names	Field name	The change of mean value	The change of variance	Percentage change in mean (%)	Percentage change in variance (%)
OPS (floating-point data)	C1	8.73	92.41	0.06%	0.00%
	C2	8.74	103.38	0.07%	0.00%
	C3	4.93	-205.32	0.04%	0.00%
LSB (floating-point data)	C1	-324.64	-100911.79	2.41%	1.02%
	C2	-54.53	537309.29	0.41%	5.41%
	C3	-59.57	484431.34	0.44%	4.90%
OPS (integer data)	C1	-0.14	2.42	0.39%	1.87%
	C2	4.80	-119.15	0.15%	0.00%
	C3	0.29	2.06	1.37%	1.42%
LSB (integer data)	C1	-0.89	3.21	2.50%	2.48%
	C2	-27.60	-146845.69	0.84%	1.84%
	C3	-0.15	0.50	0.74%	0.34%

4.2 Precision Control Analysis Experiment

To check the effect of watermark embedding before and after on the statistical distortion of the data, the precision control analysis experiment is designed in this section for numerical data. In which, precision is defined as the degree of accuracy and precision of data display.

In this section, it observes the changes of mean and variance of database watermark before and after embedding by using the optimal watermarking algorithm based on pattern search (OPS) and LSB modification algorithm (LSB) for integer and floating-point data, respectively, when the precision is {"10", "1", "0.1", "0.01", "0.001"}.

The results of the experiments are shown in Fig. 5. It can be seen from the left panel of Fig. 5 that for the optimal watermarking algorithm based on pattern search, the effect of mean change of integer data is slightly smaller than that of floating-point data, but the difference is not significant. For the LSB modification algorithm, the effect of the mean change of integer data is generally smaller than that of floating-point data

in terms of the overall trend. It can be seen from the right panel of Fig. 5 that for the optimal watermarking algorithm based on pattern search, whether it is for integer or floating-point data, it is almost negligible in terms of variance variation. For the LSB modification algorithm, on the other hand, there is less effect of variance change for integer data compared with floating-point data from the overall trend.

In conclusion, for precision control analysis, the effect of statistical distortion of the optimal pattern search-based watermarking algorithm is slightly less than that of the LSB modification algorithm.

Fig. 5. The effect of different precision on statistical distortion before and after watermark embedding in databases.

4.3 Watermark Robustness Ability Comparison Experiment

In this section, it is focused on the robustness analysis of the five database watermarking algorithms designed and used earlier, and the corresponding watermarking attack simulation experiments are designed.

In this section, the watermark similarity *Similarity* is used to measure the robustness of the watermarking method [34], and the watermark similarity is defined as shown in Eq. (7).

$$Similarity = 1 - \frac{sum(W \otimes W')}{|W|} \quad (7)$$

It is obvious that higher *Similarity* indicates better robustness; on the contrary, we can see that lower *Similarity* means worse robustness.

In this section, the experiments such as subset deletion attack, subset modification attack, and subset increase attack are performed respectively, and the specific cases can be classified as the attacker attacks (deletion, modification, and addition) 0%–90% (or 100%) of the tuples. In particular, the experimental results corresponding to the numerical watermarking algorithm are shown in Fig. 6, 7 and 8, and the experimental results corresponding to the text-based watermarking algorithm are shown in Fig. 9, 10 and 11, respectively. In which, the y-axis indicates the watermark similarity *Similarity*, and the x-axis indicates the percentage (%) of the number of attacked (deleted, modified, added) tuples to the total number of tuples. For each subset of attack experiments, the experimental results are the average of 10 experiments.

Robustness Analysis of Numerical Watermarking Algorithms. The numerical data supported in this section, which includes mainly the following types, are shown in Table 5.

Table 5. The numerical types supported by the numerical watermarking algorithm

Numerical types	Support or not
TINYINT	✓
SMALLINT	✓
MEDIUMINT	✓
INT or INTEGER	✓
BIGINT	✓
FLOAT	✓
DOUBLE	✓
DECIMAL	✓

In this section, the numerical watermarking algorithm designed and used in this paper is analyzed for robustness, and a comparative experimental analysis is performed with the literature [7], literature [25], literature [26], and literature [30] concerning the resistance to attacks of the watermark.

Subset Deletion Attack Experiments. The subset deletion attack refers to the deletion of a certain percentage of tuples from the relational data containing watermarks, to reduce the number of tuples embedded with the watermark and achieve the purpose of destroying the watermark [35].

The results of the experiments are shown in Fig. 6. From Fig. 6, it can be seen that for the LSB modification algorithm, the watermark similarity of this algorithm is still 1 when 90% of the tuples in the database are randomly deleted. When the percentage of randomly deleted tuples is less than or equal to 90%, the two algorithms in the literature [7] and the literature [25] are the same as this algorithm. The algorithms in literature [26] and literature [30] can no longer extract the original watermark completely when the proportion of randomly deleted tuples is 88% and 90%, respectively. Therefore, it is clear from the experimental results that the present algorithm is more robust against subset deletion attacks.

Subset Modification Attack Experiment. The subset modification attack refers to the modification of a certain percentage of tuples in the relational data containing the watermark, which leads to an error in the watermark detection and achieves the purpose of destroying the watermark [35].

The results of the experiments are shown in Fig. 7. It can be seen from Fig. 7 that for the LSB algorithm, the watermark similarity of this algorithm remains 1 when the percentage of tuples subject to subset modification attack reaches 100% for both

Fig. 6. Subset deletion attack experiment (numerical).

floating-point and integer data. When the percentage of modified tuples reaches 90%, the two algorithms in the literature [25] and the literature [30] are the same as this algorithm. The algorithms in literature [7] and literature [26] can no longer extract the original watermark completely when the proportion of modified tuples is 90% and 70%, respectively. Therefore, it is clear from the experimental results that the present algorithm is more robust against subset modification attacks.

Fig. 7. Subset modification attack experiment (numerical).

Subset Addition Attack Experiment. The subset addition attack refers to adding a certain percentage of tuples to the relational data containing the watermark, which does not affect

the original data but disrupts the watermark detection, leading to incorrect watermark detection [35].

The results of the experiments are shown in Fig. 8. It can be seen from Fig. 8 that for the LSB algorithm, the watermark similarity of this algorithm remains 1 when the percentage of tuples added by the attacker reaches 100%. The three algorithms in the literature [25], literature [26], and literature [30] are the same as this algorithm when the percentage of tuples added reaches 100%. The algorithm in literature [7] can extract only 80% of the watermark information when the percentage of tuples added is 60%. Therefore, it is clear from the experimental results that the present algorithm is more robust against the subset addition attack.

Fig. 8. Subset addition attack experiment (numerical).

Robustness Analysis of Text-Based Watermarking Algorithms. The string type data supported in this section, which includes mainly the following types, are shown in Table 6.

In this section, the text-based watermarking algorithm designed and used in this paper is analyzed for robustness, and the space embedding algorithm, the symbol modification algorithm, and the lexical inverse ordinal algorithm are compared and experimentally analyzed with respect concerning of the watermark.

Table 6. The string types supported by the text-based watermarking algorithm

String types	Support or not
CHAR	✓
VARCHAR	✓
TINYBLOB	✗
TINYTEXT	✓
BLOB	✗
TEXT	✓
MEDIUMBLOB	✗
MEDIUMTEXT	✓
LOB	✗
LONGTEXT	✓

Subset Deletion Attack Experiments. The results of the experiments are shown in Fig. 9. It can be seen from Fig. 9 that the watermark similarity of the space embedding algorithm and the symbolic modification algorithm have exactly the same change trend with the addition of the percentage of the deleted tuples. The changing trend of the lexical inverse ordinal number algorithm is more similar to the above two algorithms, and its corresponding watermark similarity is also slightly lower, but the difference is less than 5%–6%. Therefore, it is clear from the experimental results that the above three algorithms are generally robust against subset deletion attacks.

Fig. 9. Subset deletion attack experiment (text-based).

Subset Modification Attack Experiment. The results of the experiments are shown in Fig. 10. It can be seen from Fig. 10 that when the percentage of modified tuples is smaller, the difference of the three algorithms against attacks is not much, and the difference is less than 5%–6%; when the percentage of modified tuples is larger, the robustness of the lexical inverse ordinal algorithm is better than the symbolic modification algorithm in terms of the overall trend, and both of them are much better than the space embedding algorithm. Therefore, it is clear from the experimental results that the symbolic modification algorithm and the lexical inverse ordinal number algorithm are more robust against subset modification attacks, while the space embedding algorithm is generally robust.

Fig. 10. Subset modification attack experiment (text-based).

Subset Addition Attack Experiment. The results of the experiments are shown in Fig. 11. It can be seen from Fig. 11 that the symbolic modification algorithm and the lexical inverse ordinal number algorithm can still maintain an extremely high watermark similarity even after adding twice the number of tuples to the database. Specifically, the symbolic modification algorithm slightly outperforms the lexical inverse ordinal number algorithm, but the difference is not significant, and the difference is less than 5% to 6%. When the percentage of added tuples exceeds 60%, the robustness of the space embedding algorithm is much worse than the above two algorithms. Therefore, it is clear from the experimental results that the space-embedding algorithm, with respect to the subset addition attack, is generally robust, while the symbolic modification algorithm and the lexical inverse ordinal number algorithm, with respect to the subset addition attack, are more robust.

Fig. 11. Subset addition attack experiment (text-based).

5 Summary

In this paper, we have proposed a new robust database watermarking scheme the originality of which stands in a novel adaptive method. As we have shown, five database watermarking algorithms are designed and used. For numerical data, LSB modification algorithm and Optimal watermarking algorithm based on pattern search are used. For text-based data, space embedding algorithm, symbol modification algorithm and lexical inverse ordinal number-based text modification algorithm are used. The adaptive algorithm is implemented by data type analysis, data volume evaluation, data sensitivity analysis, automatic parameter setting and visualization techniques. It is robust against most common database attacks: tuple deletion and insertion as well as attributes' values modification. Our scheme is appropriate for copyright protection or traitor tracing. In addition, we have theoretically established and verified experimentally the performance of our method in terms of robustness. The proposed results allow the user to correctly select our scheme's parameters under constraints of robustness and capacity.

References

1. Naimi, A.I., Westreich, D.J.: Big data: a revolution that will transform how we live, work, and think (2014)
2. Katzenbeisser, S., Petitcolas, F.A.P.: Digital Watermarking, p. 2. Artech House, London (2000)
3. Agrawal, R., Kiernan, J.: Watermarking relational databases. In: VLDB 2002: Proceedings of the 28th International Conference on Very Large Databases, pp. 155–166. Morgan Kaufmann (2002)
4. Gupta, G., Pieprzyk, J.: Reversible and blind database watermarking using difference expansion. *Int. J. Digit. Crime Forensics (IJDCF)* **1**(2), 42–54 (2009)
5. Jawad, K., Khan, A.: Genetic algorithm and difference expansion based reversible watermarking for relational databases. *J. Syst. Softw.* **86**(11), 2742–2753 (2013)

6. Thodi, D.M., Rodriguez, J.J.: Prediction-error based reversible watermarking. In: 2004 International Conference on Image Processing, ICIP 2004, vol. 3, pp. 1549–1552. IEEE (2004)
7. Farfoura, M.E., Horng, S.J., Lai, J.L., et al.: A blind reversible method for watermarking relational databases based on a time-stamping protocol. *Expert Syst. Appl.* **39**(3), 3185–3196 (2012)
8. Zhang, Y., Yang, B., Niu, X.M.: Reversible watermarking for relational database authentication. *J. Comput.* **17**(2), 59–66 (2006)
9. Hu, D., Zhao, D., Zheng, S.: A new robust approach for reversible database watermarking with distortion control. *IEEE Trans. Knowl. Data Eng.* **31**(6), 1024–1037 (2018)
10. Li, Y., Wang, J., Ge, S., et al.: A reversible database watermarking method with low distortion. *Math. Biosci. Eng.* **16**(5), 4053–4068 (2019)
11. Kaur, M., Mahajan, K.: Performance evaluation of natural language text watermarking using encryption techniques. *Int. J. Comput. Appl.* **129**(3), 22–28 (2015)
12. Li, G., Chen, J., Ma, H., et al.: Method for text watermarking based on subject-verb encoding. *Comput. Sci.* **S2** (2015)
13. Lin, X., Tang, X., Wang, J.: A reversible text watermarking algorithm based on coding and synonymy substitution. *J. Chin. Inf. Process.* **29**(4), 151–158 (2015). (in Chinese)
14. Atallah, M.J., Raskin, V., Crogan, M., et al.: Natural language watermarking: design, analysis, and a proof-of-concept implementation. In: Moskowitz, I.S. (ed.) *Information Hiding. IH 2001. Lecture Notes in Computer Science*, vol. 2137, pp. 185–200. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45496-9_14
15. Kamaruddin, N.S., Kamsin, A., Por, L.Y., et al.: A review of text watermarking: theory, methods, and applications. *IEEE Access* **6**, 8011–8028 (2018)
16. Dai, Z.X., Hong, F.: Text information hiding based on inverse order of part of speech symbol sequence. *Jisuanji Gongcheng yu Yingyong (Comput. Eng. Appl.)* **43**(14), 160–161 (2007). (in Chinese)
17. Mir, N.: Copyright for web content using invisible text watermarking. *Comput. Hum. Behav.* **30**, 648–653 (2014)
18. Taleby Ahvanooe, M., Dana Mazraeh, H., Tabasi, S.H.: An innovative technique for web text watermarking (AITW). *Inf. Secur. J.: Glob. Perspect.* **25**(4–6), 191–196 (2016)
19. Zhenyu, Z., Qianmu, L., Yong, Q.: Text watermarking design based on invisible characters. *J. Nanjing Univ. Sci. Technol.* (2017). (in Chinese)
20. Zhaocan, L., Liming, W., Sijiang, G., et al.: A plain text watermarking method for big data based on orthogonal coding. *Comput. Sci.* **46**(12), 148–154 (2019). (in Chinese)
21. Zhang, X.: Several algorithms of digital watermark based on text document. *Comput. Mod.* **3** (2009). (in Chinese)
22. Wang, S., Sa, S.X.: *Introduction to Database System*, pp. 36–69. Higher Education Press, Beijing (2011)
23. Sion, R., Atallah, M., Prabhakar, S.: Rights protection for categorical data. *IEEE Trans. Knowl. Data Eng.* **17**(7), 912–926 (2005)
24. Niu, X.M., Zhao, L., Huang, W., et al.: Watermarking relational databases for ownership protection. *Acta Electron. Sinica A* **12** (2003). (in Chinese)
25. Iftikhar, S., Kamran, M., Anwar, Z.: RRW—a robust and reversible watermarking technique for relational data. *IEEE Trans. Knowl. Data Eng.* **27**(4), 1132–1145 (2014)
26. Tong, D., Zhu, C., Ren, N.: A watermarking algorithm for geodatabases without relying on primary keys. *Geogr. Geogr. Inf. Sci.* **5** (2015). (in Chinese)
27. Wang, C., Yang, L., Wan, F., et al.: Survey on database watermarking models and algorithms. *Acta Electron. Sinica* **47**(4), 946 (2019). (in Chinese)
28. Sion, R.: Proving ownership over categorical data. In: *Proceedings of 20th International Conference on Data Engineering*, pp. 584–595. IEEE (2004)

29. Hanyurwimfura, D., Liu, Y., Liu, Z.: Text format based relational database watermarking for non-numeric data. In: 2010 International Conference on Computer Design and Applications, vol. 4, pp. V4-312–V4-316. IEEE (2010)
30. Melkundi, S., Chandankhede, C.: A robust technique for relational database watermarking and verification. In: 2015 International Conference on Communication, Information & Computing Technology (ICCICT), pp. 1–7. IEEE (2015)
31. Ma, J.: Generalized pattern search algorithm in optimization problems. Dalian University of Technology (2009). (in Chinese)
32. Blake, C.L., Merz, C.J.: UCI repository of machine learning databases (1998)
33. Sun, J.: Daily news for stock market prediction, version 1, August 2016. <https://www.kaggle.com/aaron7sun/stocknews>
34. 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)
35. Hou, R.T., Xian, H.Q., Li, J., Di, G.D.: Graded reversible watermarking scheme for relational data. *Ruan Jian Xue Bao/J. Softw.* **31**(11), 3571–3587 (2020). (in Chinese)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.