A Survey of Data Distortion Watermarking Relational Databases

Ming-Ru Xie[2], Chia-Chun Wu[3], Jau-Ji Shen[2], Min-Shiang Hwang[1,4]
(Corresponding author: Min-Shiang Hwang)

Department of Computer Science and Information Engineering, Asia University[1]
No. 500, Lioufeng Rd., Wufeng, Taichung 41354, Taiwan (R.O.C.)
(Email: mshwang@asia.edu.tw)
Department of Management Information System, National Chung Hsing University[2]
Department of Industrial Engineering and Management, National Quemoy University[3]
No. 1, University Rd., Jinning Township, Kinmen County 892, Taiwan (R.O.C.)
Department of Medical Research, China Medical University Hospital, China Medical University[4]
No. 91, Hsueh-Shih Road, Taichung 40402, Taiwan (R.O.C.)

(Received Oct. 18, 2015; revised and accepted Jan. 19, 2016)

Abstract

Watermarking relational database is a technique which can provide ownership protection and temper proofing for relational databases. Although it has been developed over ten years, it is still not popular. For attracting more people to study this technique, we introduce it in detail in this paper. The main contributions of this paper include: 1) To the best of our knowledge, this is the first paper which specially surveys data distortion watermarking relational databases; 2) We define a new requirement analysis table for data distortion watermarking relational databases and use it to analyze important and the newest research of data distortion watermarking relational databases; 3) We explain background knowledge of watermarking relational databases, such as types of attacks, requirements, and basic techniques.

Keywords: Database security, database watermarking, ownership protection, watermarking relational databases

1 Introduction

Since the first set of relational database product appeared in 1981, it has gradually become an important software system which is used to store data for a private company and government institutions. A private company uses the relational database to store customer data, ordering data, shipment data of a company, etc. The government uses it to store project data, tax data, etc.

In early stage, the relational database can only store data, and then Data Warehouse and Data Mining technology appear, which make the relational database can analyze and find out the hidden special relationships among data in the database by data mining; and these are available for a company to make decision [13]. It can be seen in the future, "data" will be an important asset. How do we protect the data stored in the relational database? Is it safe enough for relational databases nowadays? We'll discuss this issue in the following paragraphs.

The data stored in the relational database is the same as images, videos, etc., and they are all digital data; they all have a characteristic that can be duplicated, and moreover, the appearance of Internet makes digital data can be easily transferred to others through the Internet, so that these issues that result in theft problems are getting worse and worse. Although the relational database has an authority control security mechanism which can limit an illegal user to access database, in recent years, the news about a legal user stealing and selling data still sometimes happens. When a legal user steals the data in the relational database and sells it, the theft party claims that the data belongs to him; how can we prove whom do the data belongs to?

In addition, due to the development of data mining technology, data owners can provide the relational database to a data mining company for data mining [12]. In the process of transferring the relational database to the recipient, the data may be stolen and tampered by an attacker, and then the attacker transfers the tampered relational database to the recipient. In this case, how do we prove that the data in the relational database is not tampered? Based on the above, we can embed watermark information into the relational database in order to prove ownership and tamper proofing for relational databases. This kind of technique is known as watermarking relational databases [1, 10, 13].

The rest of our paper is arranged as follows: Section 2 briefly introduces the history of watermarking relational databases. We explain background knowledge of watermarking relational databases in Section 3. Section 4 surveys in detail important and the newest research of data distortion watermarking relational databases and elaborates requirement analysis tables for them. In Section 5, we compare techniques of Section 4 and conclude by some issues for watermarking relational databases, and then propose future work.

2 Related Work

In digital media, such as videos, images, the technique which embeds a digital watermark to prove copyright has been developed for many years. In 2000, Khanna et al. proposed a concept to use a digital watermark in a database in order to protect a database of map information [14], and then many scholars began to research in this area. Finally in 2002, Agrawal and Kiernan proposed the first implementation method [1]. They calculated one LSB of one numeric attribute of some tuples in the relational database, and this is where they intend to embed the watermark. Next, they embedded the watermark into the selected LSB.

The research for watermarking relational databases can be grouped into two kinds: Data distortion watermarking relational databases and data distortion-free watermarking relational databases [3, 6, 22]. The research proposed by Li et al. [18], Yang et al. [28], Mehta et al. [20], Ali et al. [2], Hanyurwimfura et al. [9], Prasannakumari et al. [24], and Melkundi et al. [21], all belong to data distortion watermarking relational databases.

Latest important research in data distortion watermarking relational databases is that Kamran et al. [12] proposed a robust, distortion minimizing technique. Their technique includes three main steps: The first step includes Data Partitioning, Selection of Data Set for Watermarking and Hash Value Computation. Its purpose is to pick the position used to embed the digital watermark. Data Partitioning uses Algorithm 1 (Get_Partitions) to partition the data. Selection of Data Set for Watermarking uses Algorithm 2 (Get_Data_Selection_Threshold) to establish threshold for singling out the data sets from data partitions in the first step, and then it uses Algorithm 3 (Get_Even_Hash_Value_Data Set) to decrease these data sets. After first step, we will get data sets which can be used to embed the watermark. The second step is Watermark Embedding, and it uses Algorithm 4 (Embed_Watermark) to embed the watermark. The third step is Watermark Decoding, and it uses Algorithm 5 (Detect_Watermark) to detect the embedded the watermark. This algorithm begins to detect the watermark after it uses Algorithms 1, 2 and 3 to find out data sets are embedded with the watermark.

A sub-domain called reversible watermarking relational databases was proposed in 2006. It comes from the image and belongs to data distortion watermarking relational databases. Generally speaking, after embedding digital watermarks, the data will distort, but this technique can recover the raw data. Zhang et al. proposed the first scheme in 2006 [29]. By expansion on a data error histogram, they accomplished reversible watermarking relational databases. However, it is not robust enough to resist violent attacks [10, 29]. Latest important research was proposed by Iftikhar and Kamran et al. [10]. They proposed an RRW technique. RRW includes four steps: The first step is Watermark preprocessing. It selects the features ready to embed the digital watermark, and then generates the watermark via Genetic Algorithm. The second step is Watermark encoding, and it uses Algorithm 1 (Watermark Encoding) to embed the watermark into selected features. The third step is Watermark decoding, and it uses Algorithm 2 (Watermark Decoding) to retrieve the watermark. The fourth step is Data recovery, and it uses Algorithm 3 (Data Recovery) to recover raw data.

Next, we discuss data distortion-free watermarking relational databases. The first scheme in this domain should be Li et al.'s scheme. Via parameters, the primary key and the secret key, they calculate the hash value of tuples and primary key, respectively. And then they determine the locations used to embed the digital watermark via the hash values. Their digital watermark is produced via the hash values and the secret key [17]. In 2006, Tsai proposed that digital watermark can be generated via using images and features of the relational database [8, 27]. Recent research in this domain is proposed by Camara et al.. Their technique first partitions the data into many square matrix groups, and then computes these groups in order to generate the watermark, and then encrypts the watermark in order to get the watermark certification. Eventually, a CA (Certification Authority) will enroll the watermark certification. At CA, we can get the original watermark from the watermark certification. After we retrieve a new watermark from the database, we can compare it with the original watermark in order to check the integrity of data in the relational database [5].

3 Background

3.1 Types of Watermarks

A digital watermark is a kind of digital signature of digital media, and it can represent the author. It is grouped into two kinds:

1) Invisible Watermark: It embeds digital watermarks which can represent the author into digital media, and tries not to affect the quality of digital media. Because the human senses cannot become aware of very tiny changes, the naked eye cannot distinguish whether the embedded digital media has digital watermarks or not.

2) Visible Watermark: Typically, it uses a logo or text as a watermark, and then these watermarks can be

identified with the naked eye [19]. Its advantage is without going through any operation, and the watermark is very clear and visible; its disadvantage is it would destroy the quality of the original digital media.

3.2 Types of Attacks

After embedding the watermark into the relational database, it might suffer from assorted purposeful and unwilled attacks. We explain these possible attacks in the following paragraphs [4, 8, 12]:

1) Insertion attack: The attacker inserts new tuples into the relational database in order to eliminate the digital watermark.

2) Alteration attack: The attacker eliminates the digital watermark by modifying the value of tuples in the relational database. As long as the attacks have changed the value of tuples, these all belong to this category, for example, Bit flipping attack.

3) Deletion attack: The attacker eliminates digital watermarks by deleting tuples in the relational database.

Above mentioned attacks are basic attacks. Advanced attacks are as follows:

1) Multifaceted attack: A sophisticated attacker would mix assorted attacks, such as insertion attack, deletion attack and alteration attack to eliminate the digital watermark in the relational database.

2) Additive attack: The attacker fakes his own ownership of the relational database by embedding his digital watermark into the relational database.

3) Subset attack: The attacker only modifies or deletes a subset of tuples or attributes in the relational database in order to eliminate the digital watermark.

4) Superset attack: The attacker adds new tuples or attributes into the watermarked relational database in order to influence retrieval of the digital watermark.

5) Subset reverse order attack: The attacker changes the locations or order of tuples or attributes in the watermarked relational database in order to eliminate the digital watermark.

6) Mix-and-Match attack: The attacker collects related information from a different relation to build his own relation.

7) Brute force attack: The attacker uses programs to guess at the possible private parameters, for example, a secret key. This attack will try all possible private parameters until it finds the correct answer. If the length of private parameters is long enough, then this attack can be prevented.

8) Benign update: A relational database embedded with the digital watermark may affect the embedded digital watermark under usual insertion, deletion and modification, so that the watermark cannot be retrieved.

9) Invertibility attack: The attacker finds the fake watermark in the watermarked database, but this fake watermark is created by a random sequence.

3.3 Requirements

According to many literatures we referred to, a technique for watermarking relational databases has the following requirements [8, 13, 15, 19]:

1) Robustness: A digital watermark must be able to resist malicious attacks. After the attack, it will not be destroyed easily, and the embedded digital watermark still be extracted.

2) Unambiguity: The digital watermark retrieved by this technique must clearly identify its owner.

3) Security: Selection of the position used to embed the digital watermark is determined by some secret parameters, for example, a secret key. These secret parameters must keep secret, and they only can be known by certain people, e.g. database owner.

4) Blindness: The digital watermark must be retrieved without the original relational database or digital watermark information.

5) Imperceptibility: The embedded digital watermark must be indistinguishable.

6) Usability: After embedding the digital watermark, the data in the relational database is still usable; the best situation is this technique does not lead to the distortion of raw data.

We think that a technique for watermarking relational databases needs to meet above mentioned six requirements, and then it will be an effective watermarking relational databases. After we survey above mentioned research, we find they only define requirements, but they don't analyze techniques for watermarking relational databases by these requirements. Therefore, it is hard to compare them. Next, we try to define a new requirement analysis table for data distortion watermarking relational databases. As far as we know, this is the first requirement analysis table which uses these requirements to analyze techniques for watermarking relational databases, so it can bring a lot of help for comparison. The explanation and the format of the requirement analysis table is as in Table 1.

About robustness, we list all attacks used in their experiments. About unambiguity, security, and blindness, we use Yes or No to show if this technique meets this requirement or not. About imperceptibility, we believe that

Table 1: The explanation and the format of the requirement analysis

Proposed Scheme	The name of proposed scheme.
Robustness	The attacks used in their experiments.
Unambiguity	Yes or No.
Security	"Yes" means it has secret parameters.
Blindness	Yes or No.
Imperceptibility (%)	The discontinuous degree of the watermark bits in the database.
Usability	The amount of data distortion.

the watermark bits are more scattered in the database, that is, the discontinuous degree of the watermark bits in the database is higher, and then this technology is better. About usability, we believe that the lower the amount of data distortion, the better this technology. The following techniques we survey will focus on the six points to explain.

3.4 Watermarking Relational Databases

watermarking relational databases is a technique which embeds an invisible digital watermark into the relational database. It includes two primary steps [16], watermark embedding stage and watermark retrieve stage. In Figure 1 it shows a watermark embedding stage for watermarking relational databases. During this stage we use a key to determine the locations used to embed digital watermarks or produce digital watermarks in data distortion-free watermarking relational databases. In Figure 2, it shows a watermark retrieve stage for watermarking relational databases. During this stage, we also use the same key to find out the locations of watermarks. If we can't retrieve our watermark from a suspicious database, it means that this database is not the original database.

3.5 Basic Techniques

1) LSB (Least Significant Bit): It's the rightmost position in a binary integer, and can decide if the number is odd or even. Because it represents the smallest unit in a binary integer, i.e. the change of LSB of the number will be very small, it is usually used to hidden watermark information.

2) Data partition (Data grouping): It's a technique which can partition database into logical non-overlapping data partitions. The basic concept is that use a secret key, hash function and number of partitions to assign tuples to partitions [12]. Because these data partitions are logically partitioned, it won't separate physical data.

3) Majority voting: It's a voting rule in real life. When it is used in watermarking relational databases, its purpose is to correct decoded watermark bits [12]. For example, during decoding stage, if a watermark bit 1 in a data partition is over half the decoded bits, the decoded watermark bit of this data partition is 1.

4 Data Distortion Watermarking Relational Databases

As mentioned above, the techniques for watermarking relational databases are mainly grouped into two kinds [3, 6, 22]:

1) Data distortion watermarking relational databases: It directly embeds the digital watermark into some data in the relational database. This will make the data produce change, and these changes represent watermark information. However, the data distortion must be tolerable, or it will make the data become worthless.

2) Data distortion-free watermarking relational databases: Its main concept is that it first partitions data into several partitions, and then uses these partitions to generate the digital watermark. Because during a watermark embedding stage, it will not embed the watermark into the database, so it doesn't result in data distortion. The purpose of most of these techniques is to keep the integrity of data in relational databases because their generated watermarks are fragile.

In the data distortion watermarking relational databases, it has many schemes, such as image-based, speech-based, content-based, and others [8]. The papers we surveyed are AHK algorithms and other schemes of data distortion watermarking relational databases which are not mentioned in [8]. Besides AHK algorithms, these schemes are not in [8], in our opinion, they belong to others of data distortion watermarking relational databases. The papers we surveyed are as follows.

4.1 Agrawal-Kiernan's Scheme

The technique proposed by Rakesh Agrawal and Jerry Kiernan [1]. Their technique has two main phases:

1) Watermark insertion:
Watermark Insertion Algorithm is used to embed the

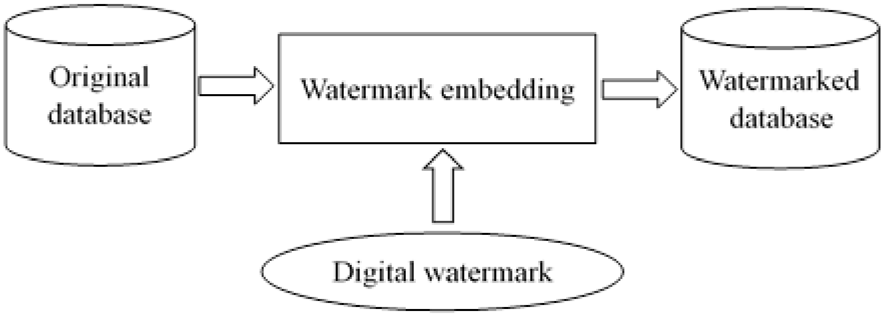

Figure 1: Watermark embedding stage for watermarking relational databases [16]

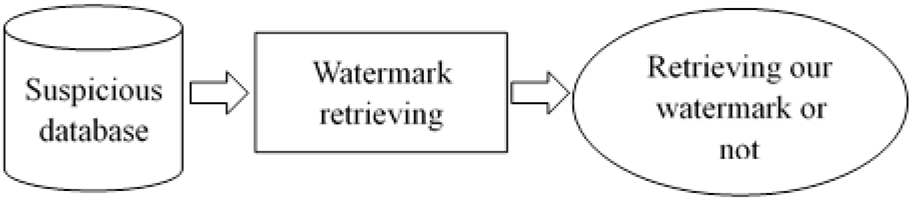

Figure 2: Watermark retrieve stage for watermarking relational databases [16]

watermark. This algorithm first uses hash function, primary key and private key e to mark one LSB of one numeric attribute of some tuples in the relational database, and then uses the value of the selected LSB to embed watermark bits: If the calculated value of the hash function (private key and primary key are passed as parameters) is even, then change the value of the selected LSB to 0; if it is odd, then change the value of the selected LSB to 1.

2) Watermark detection:
Watermark Detection Algorithm is used to retrieve the watermark. It first uses the same hash function, primary key and private key e to find out the LSBs which are embedded in the watermark. If the value of hash function is even (first hash is even) and the value of the selected LSB equals 0, then the watermark bit matches successfully. Similarly, if the value of hash function is odd and the value of the selected LSB equals 1, then it matches, too. The number of successful matches can determine whether the database is private or not.

Comment:

1) This technique is only suitable for numeric attributes. And it is assumed to modify the value of the LSB of numeric attributes and will not affect the usability of these data.

2) After a hacker changed schemes of relations, this technique would not find the original position embedded with a watermark. For example, adding or deleting an attribute in a relation, or changing the order of a relation [13].

3) Even if we can extract the complete watermark, because the extracted watermark don't have any ownership information, it is hard to clearly find whom the database belong to [13].

4) Their data distortion can be controlled arbitrarily by data owner through parameters: ν, ζ, and γ.

5) The requirement analysis is listed in Table 2.

Table 2: The requirement analysis of Agrawal-Kiernan's scheme [1]

Proposed Scheme	Agrawal et al. Technique
Robustness	Bit flipping, Mix-and-Match, Additive, & Invertibility attacks
Unambiguity	No
Security	Yes
Blindness	Yes
Imperceptibility (%)	100%
Usability	Controlled by data owner

4.2 Sion-Atallah-Prabhakar's Scheme

The technique proposed by Radu Sion, Mikhail Atallah, and Sunil Prabhakar [26]. They proposed embedding the watermark into data statistics. Their technique has two main phases, encoding phase and decoding phase. During encoding phase, it first partitions original data into subsets, and then uses Single Bit Encoding Algorithm to embed watermark bits into these subsets. During decoding phase, it first uses the partition technique of encoding phase to recover the subsets, and then uses Watermark Detection Algorithm to retrieve the watermark bits from these subsets. Finally, because these watermark bits may suffer from attacker's damage, it uses an error correcting mechanism to recover the most possible original watermark bits.

Comment:

1) Their proposed data partition technique is difficult to resist tuple deletion attack and tuple insertion attack [25].
2) During decoding phase, they use a threshold technique with two thresholds. However, they don't use optimal thresholds, and they pick thresholds at random instead [25].
3) Their data distortion can be controlled arbitrarily by data owner through data quality (goodness) metrics.
4) The requirement analysis is listed in Table 3.

Table 3: The requirement analysis of Sion-Atallah-Prabhakar's scheme [26]

Proposed Scheme	Sion et al. Technique
Robustness	Insertion, Alteration, & Deletion attacks [25]
Unambiguity	Yes
Security	Yes
Blindness	Yes
Imperceptibility (%)	100%
Usability	Controlled by data owner

4.3 Shehab-Bertino-Ghafoor's Scheme

The technique proposed by Mohamed Shehab, Elisa Bertino, and Arif Ghafoor [25]. Their technique has two main phases:

1) Watermark encoding:

   a. Data set partitioning: Use a secret key Ks, number of partitions m and get_partitions algorithm to partition Data Set D into m non-overlapping data partitions $\{S_0, S_1, \cdots, S_{m-1}\}$.

   b. Watermark encoding: Use encode_single_bit algorithm to embed the watermark into partitions.

   c. Optimal threshold evaluation: Calculate the optimal threshold T^* to be used for decoding.

2) Watermark decoding:

   a. Data set partitioning: Use get_partitions algorithm of watermark encoding to find out partitions embedded with the watermark.

   b. Threshold-based decoding: Use optimal threshold T^* to decode watermark bits.It first computes the value of $\Theta(S_j, 0, c)$, and saves it into value. If value $\geq T^*$, it represents bit 1; else it represents bit 0.

   c. Majority voting: The watermark bit is determined through voting, and the majority of watermark bits are the final bit.

Comment:

1) This technique assumes that the tuples in every partition S_i all contain a numeric attribute, and therefore it is only suitable for numeric attributes.
2) Their data distortion can be controlled by data owner through usability constraints in G.
3) The requirement analysis is listed in Table 4.

Table 4: The requirement analysis of Shehab-Bertino-Ghafoor's scheme [25]

Proposed Scheme	Shehab et al. Technique
Robustness	Insertion, Alteration, & Deletion attacks
Unambiguity	Yes
Security	Yes
Blindness	Yes
Imperceptibility (%)	100%
Usability	Controlled by data owner

4.4 Kamran-Farooq's Scheme

The technique proposed by Kamran and Farooq [11]. Their technique has two main phases:

1) Watermark encoding:

   a. Data grouping: It first uses feature ranking to compute vector R_{nk} and C_{PT}, and then uses R_{nk}, C_{PT} and data grouping function to partition features into logical non-overlapping groups.

   b. Watermark embedding: Use Algorithm 1 to embed the watermark into non-numeric features. It first computes the hash value of each row, and then uses the order of these hash values to embed watermark bits. These hash values will be saved in temp for decoding. Use Algorithm 2 to embed the watermark into numeric features of selected data groups. It uses the row value and $\triangle_i$ to embed watermark bits. If the row value adds positive $\triangle_i$, it represents a watermark bit 1; and if the row value adds negative $\triangle_i$, it represents a watermark bit 0.

2) Watermark decoding:

   a. Data grouping: Use data grouping function of Watermark encoding to find out the data groups embedded with the watermark.

   b. Watermark extraction: Use Algorithm 3 to extract the watermark from selected non-numeric features. It first gets hash values from temp, and then analyzes the order of these hash values. The descending order of hash values represents bit 1, and the ascending order of hash values represents bit 0. Use Algorithm 4 to extract the watermark from numeric features of selected data group. It uses decoding threshold T^* and a parameter val to decode. If $val > T^*$, it represents bit 1; else it represents bit 0.

Comment:

1) In Algorithms 1 and 3, the data stored in temp is too large. For example, temp needs to store the hash value of each row, if there are 10,000 rows, it needs to store 10,000 hash values.

2) This technique is not only suitable for numeric attributes, but also suitable for non-numeric attributes.

3) Their data distortion only happens in numeric attributes, and can be controlled through $\triangle_i$.

4) The requirement analysis is listed in Table 5.

Table 5: The requirement analysis of Kamran-Farooq's scheme [11]

Proposed Scheme	Kamran and Farooq Technique
Robustness	Alteration & Deletion attacks
Unambiguity	Yes
Security	Yes
Blindness	Yes
Imperceptibility (%)	100%
Usability	Controlled by $\triangle_i$

4.5 Kamran-Suhail-Farooq's Scheme

The technique proposed by Kamran, Suhail, and Farooq [12]. Their technique has three main phases:

1) Pick the position used to embed the digital watermark:

   a. Data Partitioning: Use a secret key Ks, number of partitions m and Algorithm 1 (Get_Partitions) to partition Data Set D into m non-overlapping data partitions $\{S_,S_1, \cdots, S_{m-1}\}$.

   b. Selection of Data Set for Watermarking: Use Algorithm 2 (Get_Data_Selection_Threshold) to establish threshold for singling out the data sets from data partitions in (a).

   c. Hash Value Computation: Use Algorithm 3 (Get_Even_Hash_Value_Data Set) to decrease these data sets. And then we will get data sets which can be used to embed the watermark. By this way, the watermark is generated by Watermark Generating Function.

2) Watermark Embedding: Use Algorithm 4 (Embed_Watermark) to embed the watermark. It first computes the amount of data change. If a watermark bit is 1, the amount of data change is row value multiplied by positive ρ; and if a watermark bit is 0, the amount of data change is row value multiplied by negative ρ. And then it uses the row value plus the amount of data change to embed watermark bits.

3) Watermark Decoding: Use Algorithm 5 (Detect_Watermark) to detect the embedded watermark. This algorithm begins to detect watermarks after it uses Algorithms 1, 2 and 3 to find out data sets which are embedded with watermarks. Next step is to compute decoding threshold ν, and then use it to decode watermark bits. If $\nu \geq 0$, it represents bit 1; else it represents bit 0. Finally, the final watermark bits are determined through Majority voting.

Comment:

1) This technique is most suitable for unsigned numeric attributes.

2) Their data distortion can be controlled by data owner through ρ.

3) The requirement analysis is listed in Table 6.

4.6 Melkundi-Chandankhede's Scheme

The technique proposed by Swathi Melkundi and Chaitali Chandankhede [21]. Their technique has three main phases:

1) Watermark Insertion:

   a. Data Partitioning: Use a secret key Ks, number of partitions m and Algorithm 1 (Data Partition Algorithm) to partition Data Set D into m non-overlapping data partitions $\{P_0, P_1, \cdots, P_{m-1}\}$.

Table 6: The requirement analysis of Kamran-Suhail-Farooq's scheme [12]

Proposed Scheme	Kamran-Suhail-Farooq Technique
Robustness	Insertion, Alteration, Deletion, Multifaceted, Collusion, & Additive attacks
Unambiguity	Yes
Security	Yes
Blindness	Yes
Imperceptibility (%)	100%
Usability	Controlled by data owner

b. Insertion into a textual attribute: Use Unicode control characters ZWJ and ZWNJ to embed watermark bits. For ZWJ, its Unicode code point is U+200D and is the abbreviation of Zero width joiner. It is an invisible control character and used to represent a watermark bit 0. For ZWNJ, its Unicode code point is U+200C and is the abbreviation of Zero-width non-joiner. It is used to represent a watermark bit 1.

c. Insertion into numeric attribute: It first converts the value of the attribute into a binary value, and then flips the LSB of the binary value. That is, if you intend to embed a watermark bit = 0, then the LSB is changed to 0; if you intend to embed a watermark bit = 1, then the LSB is changed to 1.

2) Watermark Extraction:

a. Data Partitioning: Use Data Partition Algorithm of Watermark Insertion to find out the partitions embedded with the watermark.

b. Extraction from a textual attribute: If the value of the selected textual attribute in a data partition is ZWJ, it represents bit 0; and if the value is ZWNJ, it represents bit 1.

c. Extraction from numeric attribute: If the value of the LSB of the selected numeric attribute in a data partition is 0, it represents bit 0; and if the value is 1, it represents bit 1.

d. Majority voting: Through voting is used to determine the watermark bit, and the majority of watermark bits is the final bit.

3) Watermark Verification: Use Algorithm 2 (Watermark Verification Algorithm) to compare the extracted watermark with the raw watermark. Its concept is based on Levenshtein distance, and therefore it computes Levenshtein Distance between the extracted watermark and the raw watermark. If their difference is too large, it shows this database is not the original one.

Comment:

1) This technique is only suitable for a relation with numeric attributes and textual attributes at the same time.

2) Their data distortion only happens in numeric attributes, and the amount of data change is only the value of the LSB.

3) According to their description, their subset addition attack, subset deletion attack and subset alteration attack are our insertion attack, deletion attack and alteration attack, respectively.

4) The requirement analysis is listed in Table 7.

Table 7: The requirement analysis of Melkundi-Chandankhede's scheme [21]

Proposed Scheme	Melkundi et al. Technique
Robustness	Insertion, Alteration, & Deletion attacks
Unambiguity	Yes
Security	Yes
Blindness	Yes
Imperceptibility (%)	100%
Usability	LSB

4.7 Mehta-Pratap Rao's Scheme

The technique proposed by Brijesh B. Mehta and Udai Pratap Rao [20]. Their technique has three main phases:

1) Watermark insertion: It first uses hash function, primary key and private key k1 to select tuples in the database.

a. Insertion into a numeric attribute: Choose a LSB of a numeric attribute of selected tuples, and a watermark bit is substituted for the selected LSB.

b. Insertion into a date attribute: Choose seconds field (SS) of a date attribute of selected tuples, and embed watermark bits into the SS.

2) Watermark extraction: It first uses the same hash function, primary key and private key k1 to find out tuples which are embedded with the watermark.

    a. Extraction from a numeric attribute: Find out the LSB of the selected numeric attribute of these selected tuples, and the value of the LSB represents a watermark bit.

    b. Extraction from a date attribute: Find out the SS of the selected date attribute of these selected tuples, and extract watermark bits from the SS.

3) Watermark verification: Compare the extracted watermark with the raw watermark. It only needs the extracted watermark bits from one place instead of two places to match the original watermark bits successfully.

Comment:

1) This technique is only suitable for a relation with numeric attributes and date attributes at the same time. Because it actually embeds a watermark bit into two attributes (numeric, date) selected by k1 at the same time. Therefore, if this relation don't have the two attributes (numeric, date), it won't work.

2) Their data distortion happens in numeric attributes and date attributes. The amount of data change for numeric attributes is the value of the LSB, and the amount of data change for date attributes is SS.

3) According to their description, their subset addition attack, subset deletion attack, subset alteration attack and subset selection attack are our insertion attack, deletion attack, alteration attack and Mix-and-Match attack, respectively.

4) The requirement analysis is listed in Table 8.

Table 8: The requirement analysis of Mehta-Pratap Rao's scheme [20]

Proposed Scheme	Mehta et al. Technique
Robustness	Insertion, Alteration, Deletion, & Mix-and-Match attacks
Unambiguity	Yes
Security	Yes
Blindness	Yes
Imperceptibility (%)	100%
Usability	LSB and SS

1Because this technique uses data partition or data grouping technique, we think the watermark bits will distribute at random in the relational database.

5 Conclusion and Future Work

In this paper, we first introduce the history and background of watermarking relational databases, and then focus on surveying data distortion watermarking relational databases. Furthermore, we analyze these techniques by six requirements we mentioned in BACKGROURD. Next, we compare these techniques through requirement analysis table.

5.1 Comparison

Our comparison method is to rate them by scores. The best score is five points, and the worst score is one point. The result is as in Table 9.

About robustness, because Kamran and Farooq Technique only uses two basic attacks in their experiment, it scores 2 points. Kamran, Suhail and Farooq Technique can resist three basic attacks (Insertion attack, Alteration attack, Deletion attack) and three advanced attacks, so it scores the best grades.

About unambiguity, because Agrawal et al. Technique is hard to find any ownership information of the embedded digital watermark, it only scores 1 point. About security and blindness, every technique meets their conditions, and therefore these techniques all score 5 points. About imperceptibility, Agrawal et al. Technique, Sion et al. Technique, etc. scores 3 points because they only use basic data partition technique. Kamran, Suhail and Farooq Technique scores the highest grades because it uses advanced technique to further decrease the number of tuples which are ready to be watermarked, hence it has the best discontinuous degree of the watermark bits. About usability, because Melkundi et al. Technique's the amount of data distortion is only LSB, it scores 5 points.

According to total score, Kamran, Suhail and Farooq Technique is a better technique than others because it has a balanced performance in six requirements. It not only is the most robust technique, but also is the imperceptiblest. Therefore, we think a good technique for watermarking relational databases should consider all six requirements, it can't only focus on a few requirements.

5.2 Issues

Although watermarking relational databases has been developed over ten years, it still has some issues, and these issues are as follows:

1) Experiments: Unlike image processing domain, some scholars of watermarking relational databases usually perform their experiments with their own databases, and without comparing their technique with others in robustness, distortion, etc.. such as [28, 20, 9, 21]. Some scholars didn't perform experiment very well, for example, Javier et al. proposed a paper in 2014 [7]. Although they compared their technique with others in experiments, they still used their own database to perform experiments. Shehab et al.

Table 9: The comparison

Proposed Scheme	Robustness	Unambiguity	Security	Blindness	Imperceptibility	Usability	Total
Agrawal et al. Technique	4	1	5	5	3	3	21
Sion et al. Technique	3	5	5	5	3	3	24
Shehab et al. Technique	3	5	5	5	3	3	24
Kamran-Farooq Technique	2	5	5	5	4	3	24
Kamran-Suhail-Farooq Technique	5	5	5	5	5	3	28
Melkundi et al. Technique	3	5	5	5	3	5	26
Mehta et al. Technique	4	5	5	5	3	4	26

provide a good example in experiments in this domain [25].

They not only compare their technique with others, but also use an online database for their experiment. Therefore, after we read their research, we think that watermarking relational databases needs some open databases for everyone to do experiment. Therefore, we will provide a website that provides open data sets for everybody. Its internet address is: `http://archive.ics.uci.edu/ml/`. In watermarking relational databases, we strongly recommend that everyone should perform complete and fair experiments.

2) Data distortion: Although data distortion watermarking relational databases in academic research can tolerate data distortion, but for commercial purposes, data distortion in the relational database is not allowed. Even a bit of data distortion, it may cause a significant impact. Therefore, the commercial value of the research in the data distortion watermarking relational databases is not high, we believe that the goal of watermarking relational databases should develop towards data distortion-free watermarking relational databases or reversible watermarking relational databases, and data distortion watermarking relational databases should be eliminated.

5.3 Future Work

1) As mentioned above in B.2), for the purpose of distortion-free data, we can consider not to embed the watermark into the content of a database, but other places of a database, such as the comment of a table, database relationship [23], or using the number of tables in a large database to embed the watermark, etc. Because we don't embed digital watermarks into the database, we don't damage the raw data, it can achieve the purpose of distortion-free data.

2) Computation time: As far as we know, most of watermarking relational databases don't consider computation time in experiments except [7]. We think that computation time is another important issue except robustness. Because in the age of big data, the amount of data will become bigger and bigger, and then the amount of data will affect the computation time. A technique which takes a lot of computation time is worthless. Therefore, in addition to robustness, computation time should be considered in experiments. In the future, we must strike a balance between robustness and computation time.

Acknowledgments

This study was supported by the National Science Council of Taiwan under grant MOST 104-2221-E-468-004. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] R. Agrawal and J. Kiernan, "Watermarking relational databases", in *Proceedings of the 28th International Conference on Very Large Data Bases*, pp. 155–166, Hong Kong, China, 2002.

[2] A. Al-Haj and A. Odeh, "Robust and blind watermarking of relational database systems", *Journal of Computer Science*, vol. 4, no. 12, pp. 1024–1029, 2008.

[3] M. H. Bhesaniya, J. Rathod, and K. Thanki, "Various approaches for watermarking of relational database", *International Journal of Engineering Science and Innovative Technology*, vol. 3, no. 1, pp. 215–220, 2014.

[4] M. H. Bhesaniya, K. Thanki, "Watermarking of relational databases", *International Journal for Research in Technological Studies*, vol. 1, no. 1, pp. 11–16, 2013.

[5] L. Camara, J. Li, R. Li, and W. Xie, "Distortion-free watermarking approach for relational database integrity checking", *Mathematical Problems in Engineering*, Article ID 697165, 2014.

[6] A. K. Dwivedi, B. K. Sharma, and A. K. Vyas, "Watermarking techniques for ownership protection of re-

lational databases", *International Journal of Emerging Technology and Advanced Engineering*, vol. 4, no. 1, pp. 368–375, 2014.

[7] J. Franco-Contreras, G. Coatrieux, F. Cuppens, N. Cuppens-Boulahia, 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.

[8] R. Halder, S. Pal, and A. Cortesi, "Watermarking techniques for relational databases: Survey, classification and comparison", *Journal of Universal Computer Science*, vol. 16, no. 21, pp. 3164–3190, 2010.

[9] D. Hanyurwimfura, L. Yuling, and L. Zhijie, "Text format based relational database watermarking for non-numeric data", in *International Conference On Computer Design And Appliations (ICCDA'10)*, vol. 4, pp. 312–316, 2010.

[10] S. Iftikhar, M. Kamran, and Z. Anwar, "RRW-A robust and reversible watermarking technique for relational data", *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 4, pp. 1132–1145, 2015.

[11] M. Kamran and M. Farooq, "A formal usability constraints model for watermarking of outsourced datasets", *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 6, pp. 1061–1072, 2013.

[12] M. Kamran, S. Suhail, and M. Farooq, "A robust, distortion minimizing technique for watermarking relational databases using once-for-all usability constraints", *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 12, pp. 2694–2707, 2013.

[13] C. H. Ke, M. S. Wang, *A Study of Watermarking in Relational Database*, M.S. Thesis, Department of Engineering Science, National Cheng Kung University, Tainan, Taiwan, 2006.

[14] S. Khanna and F. Zane, "Watermarking maps: Hiding information in structured data", in *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 596–605, San Francisco, California, USA, 2000.

[15] S. S. Kshatriya and S. S. Sane, "A Study of Watermarking Relational Databases", *International Journal of Application or Innovation in Engineering & Management*, vol. 3, no. 10, pp. 154–158, 2014.

[16] Y. Li, "Database watermarking: A systematic view", in *Handbook of Database Security*, pp. 329–355, Springer US, 2008.

[17] Y. Li, H. Guo, and S. Jajodia, "Tamper detection and localization for categorical data using fragile watermarks", in *Proceedings of the 4th ACM Workshop on Digital Rights Management*, pp. 73–82, Washington DC, USA, 2004.

[18] Y. Li, V. Swarup, and S. Jajodia, "Constructing a virtual primary key for fingerprinting relational data", in *Proceedings of the 3rd ACM Workshop on Digital Rights Management*, pp. 133–141, Washington, DC, USA, 2003.

[19] B. B. Mehta and H. D. Aswar, "Watermarking for security in database: A review", in *IEEE Conference on IT in Business, Industry and Government (CSIBIG'14)*, pp. 1–6, 2014.

[20] B. B. Mehta and U. P. Rao, "A novel approach as multi-place watermarking for security in databas", in *International Conference on Security and Management*, pp. 703–707, 2011.

[21] S. Melkundi and C. Chandankhede, "A robust technique for relational database watermarking and verification", in *IEEE International Conference on Communication, Information & Computing Technology (ICCICT'15)*, pp. 1–7, 2015.

[22] A. A. Mohanpurkar and M. S. Joshi, "Applying watermarking for copyright protection, traitor identification and joint ownership: A review", in *IEEE World Congress on Information and Communication Technologies (WICT'11)*, pp. 1014–1019, 2011.

[23] H. Pieterse and M. Olivier, "Data hiding techniques for database environments", in *IFIP Advances in Information and Communication Technology*, Advances in Digital Forensics VIII, pp. 289–301, Springer Berlin Heidelberg, 2012.

[24] V. Prasannakumari, "A robust tamperproof watermarking for data integrity in relational databases", *Research Journal of Information Technology*, vol. 1, no. 3, pp. 115–121, 2009.

[25] M. Shehab, E. Bertino, and A. Ghafoor, "Watermarking relational databases using optimization-based techniques", *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 1, pp. 116–129, 2008.

[26] R. Sion, M. Atallah, and S. Prabhakar, "Rights protection for relational data", *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 6, pp. 1509–1525, June 2004.

[27] M. H. Tsai, H. Y. Tseng, and C. Y. Lai, "A Database Watermarking Technique for Temper Detection", in *Proceedings of the 2006 Joint Conference on Information Sciences (JCIS'06)*, Kaohsiung, Taiwan, Atlantis Press, 2006.

[28] Y. Y. Yang, D. C. Wu, W. H. Tsai, "Watermarking of numerical databases using spread spectrum techniques", in *Proceedings of 4th Workshop on Digital Archives Technologies*, pp. 79-84, 2005.

[29] Y. Zhang, B. Yang, and X. M. Niu, "Reversible watermarking for relational database authentication", *Journal of Computers*, vol. 17, no. 2, pp. 59–66, 2006.

Ming-Ru Xie received the B.S. in Computer and Information Science from Aletheia University, New Taipei City, Taiwan, Republic of China, in 2002. He had worked in IT industry in Taiwan for ten years. He is currently a master's degree student in the Department of Management Information System, National Chung Hsing University, Taichung, Taiwan. His current research interests include database security, information security, and digital image techniques.

Chia-Chun Wu received a Ph.D. degree in Department of Computer Science and Engineering from National Chung-Hsing University, Taichung, Taiwan, in 2011. He is currently an assistant professor at the Department of Industrial Engineering and Management, National Quemoy University, Kinmen County, Taiwan. His current research interests include database security, secret image sharing, mobile applications development, and digital image techniques.

Jau-Ji Shen received his Ph.D. degree in Computer Science and Information Engineering in 1988 from National Taiwan University, Taipei, Taiwan. Currently, he is a professor at Management Information Systems, National Chung Hsing University, Taichung, Taiwan. His research interests include software quality assurance, data and knowledge techniques, and digital image techniques.

Min-Shiang Hwang received the B.S. in Electronic Engineering from National Taipei Institute of Technology, Taipei, Taiwan, Republic of China, in 1980; the M.S. in Industrial Engineering from National Tsing Hua University, Taiwan, in 1988; and the Ph.D. in Computer and Information Science from National Chiao Tung University, Taiwan, in 1995. He also studied Applied Mathematics at National Cheng Kung University, Taiwan, from 1984-1986. Dr. Hwang passed the National Higher Examination in field "Electronic Engineer" in 1988. He also passed the National Telecommunication Special Examination in field "Information Engineering", qualified as advanced technician the first class in 1990. From 1988 to 1991, he was the leader of the Computer Center at Telecommunication Laboratories (TL), Ministry of Transportation and Communications, ROC. He was also a project leader for research in computer security at TL in July 1990. He obtained the 1997, 1998, and 1999 Distinguished Research Awards of the National Science Council of the Republic of China. He is a member of IEEE, ACM, and Chinese Information Security Association. His current research interests include database and data security, cryptography, image compression, and mobile communications.