Comparative Analysis of Relational Database Watermarking Techniques: An Empirical Study

SAPANA RANI AND RAJU HALDER
Indian Institute of Technology Patna, 801106, India (e-mails: {sapana.pcs13, halder}@iitp.ac.in)
Corresponding author: Sapana Rani (e-mail: sapana.pcs13@iitp.ac.in).

ABSTRACT
Digital watermarking is considered one of the most promising techniques to verify the authenticity and integrity of digital data. It is used for a wide range of applications, e.g., copyright protection, tamper detection, traitor tracing, maintaining the integrity of data, etc. In the past two decades, a wide range of algorithms for relational database watermarking has been proposed. Even though a number of surveys exist in the literature, they are unable to provide insightful guidance to choose the right watermarking technique for a given application. In this paper, we provide an exhaustive empirical study and thorough comparative analysis of various relational database watermarking techniques in the literature. Our work is different from the existing survey papers as we consider both distortion-based and distortion-free techniques along with a rigorous experimental analysis demonstrating a detailed comparison on robustness, data usability, and computational cost with considerable empirical evidence.

INDEX TERMS
Digital Watermarking, Relational Database, Empirical Study, Robustness, Data Usability

I. INTRODUCTION
Digital watermarking is considered one of the most promising techniques to verify the authenticity and integrity of digital data. It is used for a wide range of applications, e.g., copyright protection, tamper detection, traitor tracing, maintaining the integrity of data, etc. For several decades, relational databases are at the heart of many information systems. As they contain crucial information, they must be protected before sharing them to the world of the internet. Although encryption is used to protect the data stored in a relational database from being accessed by individuals with malicious intent, but it is very restrictive in nature. Since the first proposal in 2000 in [1] that used digital watermark for protecting a database of map information, various relational database watermarking techniques have been proposed in the literature thereafter. Among them, the first and most significant one is proposed by Agrawal and Kiernan in [2]. The database watermarking techniques embed a piece of information (known as watermark) in an underlying data and extract it later from any suspicious content in order to verify the absence or presence of any possible attacks. The former phase is known as Embedding phase, whereas the later phase is known as Detection or Verification phase. In general, these database watermarking techniques are classified as (i) distortion-based techniques that embed the watermark into the underlying content of the data and (ii) distortion-free techniques that generate the watermark based on various characteristics of the data.

A number of survey papers [3]–[11] already exist in the literature, which provides a comprehensive summary of different techniques and their comparison. Authors in [3] elaborated the features of the relational databases, application of digital watermarking, attack analysis of the then existing distortion-based and distortion-free watermarking techniques. A survey of reversible watermarking approaches has been proposed in [4], [5]. A holistic study of distortion-based watermarking techniques has been proposed in [6]. A recent survey on multimedia and database watermarking is reported in [7] where, in addition to different multimedia artifacts, a comparative summary of only nine existing database watermarking techniques is presented. Other significant works related to the survey of relational database watermarking include [8]–[11].

Despite this, the existing survey papers do not carry the following insights that may provide an appropriate guidance to choose the right watermarking technique for a given appli-
cation: (i) what should be the criteria to compare different categories of watermarking techniques, (ii) how to show empirically that a particular watermarking technique is better than the other techniques, (iii) lack of emphasis towards distortion-free techniques.

To fill this knowledge gap and to provide a well-informed guidance to the users for a wise decision on choosing right watermarking technique, in this paper, we provide an exhaustive empirical study and thorough comparative analysis of various relational database watermarking techniques. Our work is different from the existing survey papers as we consider both distortion-based and distortion-free techniques along with a comprehensive experimental analysis of robustness, data usability, and computational cost, and their comparisons with considerable empirical evidence.

In order to achieve these objectives, our major contributions in this paper are as follows:

1) We classify the distortion-based and distortion-free techniques in various categories on the basis of the algorithmic steps adopted as well as the type of the watermark information used in the algorithm.

2) We perform an empirical study on a selected number of algorithms, each representing the class of algorithm it belongs to. In particular, we perform a rigorous experimental analysis demonstrating a detailed comparison on robustness, data usability, and computational cost.

3) Our empirical analysis provides a well-informed guidance to the users for a wise decision on choosing right watermarking technique.

The structure of the rest of the paper is as follows: Section II explains the research methodology we adopted. Section III and IV provide the detailed comparative performance analysis of distortion-based and distortion-free algorithms respectively. Section V discusses our evaluation-results w.r.t. the existing experimental observations. Section VI provides a guidance to the users for choosing the right watermarking technique for a given application. Finally, we conclude our work in Section VII.

II. RESEARCH METHODOLOGY

A. PRIMARY STUDY SELECTION

We perform the primary study by searching the major online scientific repositories (depicted in Table 1) using the following search queries: “relational database watermarking”, “watermarking of relational databases”, and “copyright protection of relational databases”. In all cases, we set as a filter the years from 2002 to 2022.

We carefully analyze each and every publication obtained in the search result by following the inclusion and exclusion criteria mentioned in the subsequent subsection.

B. INCLUSION/EXCLUSION CRITERIA

In this study, we consider research works published in journal, conference, symposium, or workshop and we exclude other kinds of works such as books, newsletters, magazines, technical reports, Ph.D. thesis, and undergraduate/master project documents. These criteria are depicted in Table 2.

| Criteria      | Explanation                                                                 |
|---------------|-----------------------------------------------------------------------------|
| Exclusion     | The research works related to other databases watermarking like XML, JSON, etc. |
|               | It is a patent.                                                              |
|               | It is not published in the English language.                                |
|               | It is a Ph.D. thesis or undergraduate/master project document.               |
|               | It is a book, newsletter, magazine, or technical report.                    |
| Inclusion     | It is a journal, conference, symposium, or workshop paper and the title, keywords, and abstract explicitly indicate that the paper is related to relational database watermarking. |

### TABLE 2: Inclusion and Exclusion Criteria

C. SELECTION RESULTS

Considering the above-mentioned search queries, we obtain the following results:

- Google Scholar: 497 results
- IEEE Xplore Digital Library: 24 results
- ACM Digital Library: 6 results
- Science Direct: 16 results
- MDPI: 2 results
- Springer: 68 results

As these search results overlap, we remove the duplicate entries and obtain 416 publications. Finally, after applying the inclusion and exclusion criteria, we obtain 94 publications that we consider in our paper. The summary of the articles by type of publication and the temporal trend of these research publications under consideration are depicted in Table 3 and Figure 1 respectively. We analyze these 94 papers on the basis of a brief overview of the watermarking technique, the data set used in the experiment, and the attacks performed.

### TABLE 3: Summary of articles by the type of the publication

| Sites      | #Articles | % Article |
|------------|-----------|-----------|
| Journal    | 46        | 48.93     |
| Conference | 34        | 36.17     |
| Symposium  | 7         | 7.44      |
| Workshop   | 7         | 7.44      |

D. QUALITY ASSESSMENT

We subsequently check the quality of the research works. The studies were classified on the basis of the types of algorithms adopted as follows: The distortion-based techniques are classified in the following six categories:
E. RESULTS

We examine the motivations, contributions, future works of the papers which passed the quality assessment. We select one algorithm in each category that follows any one of the following criteria for the experimental analysis:

1) Criteria 1: Select the pioneer work if the other recent works are minor variants of the pioneer work and there is no significant improvement.
2) Criteria 2: If there is a significant improvement in the recent work compared to the previous works then select the recent one.
3) Criteria 3: Select a work in a category if the work is published in a publication having a higher core ranking and h-5 index.

Let us discuss the research works under each category of distortion-based and distortion-free techniques in detail.

1) Distortion-based techniques:

The distortion-based watermarking techniques are classified on the basis of the algorithmic steps adopted as well as the type of the watermark information, described below:

a: Meaningless bit-pattern as the watermark

Authors in [2], [12]–[20] propose the watermarking algorithms that embed a meaningless bit pattern of the watermark into the data set. Authors in [2] have proposed the algorithm in which hash function is used to decide the marking of a particular tuple. Authors in [12], [14]–[16], [21] extend the proposal of [2]. For example, in [12] the pseudo-random number generator is used instead of a hash function. In [14] chaotic random number generator is used instead of the hash value. Gupta et al. in [15] extend the proposal of Agrawal et al. in [2] and propose a reversible watermarking algorithm. Authors in [16] use the similar approach of [2] but instead of flipping the least significant bits (LSB) they embed random digits (0 to 9) at LSB of the attribute values. Authors in [17] apply data flow analysis to identify the variant and non-variant parts of the relational database, and then apply the watermarking algorithm in [2] to embed the watermark.

b: Virtual Primary Key based

In most of the watermarking algorithms, it is assumed that the primary key exists and is not distorted by the attackers. However, it may not be always true. To deal with this situation, various techniques have been proposed to generate and use Virtual Primary Key (VPK) instead of a primary key. Authors in [2] propose an extended proposal named as S-Scheme in [22]. In S-scheme, one attribute is used to generate the VPK and the remaining attributes are used for watermark embedding. Authors in [22] propose E-scheme and M-scheme. The VPKs generated in E-scheme is similar to the S-scheme, but it considers all of the attributes. M-scheme considers more than one attribute per tuple to generate the VPK. In this approach, each time a different attribute is selected and hence is more resilient towards the delete problem. Other approaches based on virtual primary keys are proposed in [23]–[27]. Approaches in [23], [24] are similar to the M-Scheme. Two attributes having hash values near zero are considered. In [23], the textual attributes are considered. The VPK is generated based on two numeric attributes in [24]. Different attributes are selected each time in [25]. The HQR-Scheme [26] generates one VPK per tuple based on the cyclic model of the attribute.
Various watermarking approaches [28]–[36] embed images as the watermark into the relational databases. All these approaches first group the tuples and then embed the bit string of the image watermark. Authors in [28] insert a binary image watermark into a relation. In the case of text data, the carriage return character represents 1 and the linefeed character represents 0 of the watermark bits. In the case of numeric data, the watermark bits are embedded in the LSBs of the attribute value.

In most of the techniques, the partitioning of tuples is based on hashing. However, in the case of Huang et al. in [29], the tuples are clustered into equivalent classes by using the k-means algorithm. The parity of the watermark bit is compared with the LSB of the candidate attribute for embedding the watermark bit. The location of the embedded watermark is assured by the clustering method. In [30], the original image of size $N \times N$ is converted into a binary string of length $L = N \times N$. The tuples are grouped into $L$ groups based on the hash function, and an $i^{th}$ bit of the binary string is embedded into the bit positions of a fixed attribute in the $i^{th}$ group. The authors in [31] follow the same algorithm as in [30] but they do not consider a fixed attribute and they do not consider the order of image during computing the bit position. After marking, the usability constraints are also checked. The approach in [32] is also similar to [31]. The difference is that they have divided the image into two parts: header and image data. The header is used for the grouping of the tuples and the image data is converted into a binary string and embedded into these groups.

d: Partitioning based
The partitioning based watermarking techniques [37]–[44] partition the data into various groups and embed the watermark into these groups independently. In [37], a marker tuple is used for partitioning and one watermark bit is embedded into one group maintaining the usability constraints. In [38], instead of marker tuple, the hash function is used for partitioning the tuples into groups, and in each group, the watermark bit is embedded by altering the group statistics satisfying the usability constraints. In [40] also, the hash function is used for partitioning. The changes are minimized by selecting a few tuples for watermarking and the watermark (generated from date-time) bit is embedded in each of the selected tuples. In [43], the tuples are partitioned and in each partition, two types of watermark, attribute watermark $W_1$ and tuple watermark $W_2$ are embedded.

e: Fake tuple/attribute insertion
The watermarking techniques in this category insert a new tuple or a new attribute into the database relation as a watermark. In [45], probability distributions are used to determine the properties of the new tuple inserted as a watermark. In [46], a new attribute is inserted into the existing relation. Parity checks are calculated from each attribute and appended to that attribute. The new attribute also has a value from the aggregate function of any of the attributes for all tuples.

f: Fingerprinting Techniques
A fingerprint is a piece of meaningful information, e.g., social security number that is used as a watermark. Authors in [48] extend the proposal of [2] but embed a fingerprint of length $L$ computed from a hash function taking input as secret key $K$ and user identifier $n$. Liu et al. in [47] propose a block-oriented fingerprinting technique. The hash function is based on a secret key and the buyer’s ID is used to generate the fingerprint. Authors in [49] propose a twice-embedding watermarking scheme. In the first process, the fingerprint value is used to select the position and the embedding value for every group. In the second process, a pattern is embedded using the fingerprint as the secret key. Authors in [50] also extend the proposal of [2] but they embed fingerprint instead of meaningless bit pattern and they propose schemes named as E-scheme and M-scheme for constructing the virtual primary keys. In [51], watermarking is based on integer linear programming constraint solving. In [52], a buyer’s “thumb impression” is used for embedding the fingerprints.

g: Other Meaningful Watermark Information based
In [41], the database tuples are partitioned based on the hash function, and meaningful information is embedded in a single attribute as the watermark. Authors in [54] use a pseudo-random sequence number to know the attribute and bit position where the watermark is to be embedded. Similarly [52], [53], [55], [56] also embed meaningful information as the watermark.

A brief overview of different distortion-based watermarking techniques within each category is depicted in Tables 4, 5, and 6.

2) Distortion-free techniques:
The distortion-free techniques can be classified into following categories:

a: Permutation of tuples
In these techniques, the order of the tuples is arranged based on secret parameters without causing any distortion in the data values. The significant proposals that perform tuple-reordering based watermarking are proposed in [57]–[61]. In [57], some secure parameters are used to partition the tuples into groups. The order of two tuples is changed based on the hash values of the tuple and the watermark bit. In [62], the value of some critical attribute(s) is used to re-order the tuples relative to a secret initial order, e.g., ascending. The proposed schemes in [58]–[61] are also similar to the approach as proposed in [57] as they also partition the tuples into groups and the tuples are re-ordered in a group that corresponds to the watermark.

b: Converting Database Relation into Binary Form
These techniques convert the database relation into a binary form. In [63], the watermark is generated from the most significant bits (MSBs) of the attribute values and can be verified publicly. In [64], the watermark can not be verified publicly as it uses a private key. The approaches in [65],
| Category                                      | Paper                        | Brief Overview                                                                 | Data Set used                          | Attacks analysed                               | Selected for experiment | Selection Criteria |
|----------------------------------------------|------------------------------|--------------------------------------------------------------------------------|----------------------------------------|-----------------------------------------------|------------------------|--------------------|
| Meaningless bit-pattern as the watermark     | Agrawal et al. [2]           | Hash function is used for selecting the attribute value and the bit position to be marked. | Forest CoverType                       | Bit flipping, mix and match, additive, invertibility, subset same as [2] | Yes                    | Criteria 1         |
|                                              | Agrawal et al. [12]          | Uses pseudo-random sequence generator instead of hash function.                   | same as [2]                            | same as [2]                                   | No                     |                    |
|                                              | Agrawal et al. [13]          | Uses pseudo-random sequence generator instead of hash function.                   | Random databases and keys were generated | mix and match                                 | No                     |                    |
|                                              | Lafayette [20]               | Describes the security analysis of [2]                                            | Generated 270 sets of random documents |                                                | No                     |                    |
|                                              | Gupta et al. [15]            | Watermark bits replace the bit of the integer part of the attribute values and it is inserted in the fraction part of the attribute value. | A table of retail data set              | subset extracting, subset addition, randomized data substituting, additive massive deletion, random deletion, incremental update, epsilon, collusion attack | No                     |                    |
|                                              | Qin et al. [14]              | Uses chaotic random series based on the Logistic chaos equation to avoid collision of hash function. | railway freight information            |                                                | No                     |                    |
|                                              | Xiao et al. [16]             | Digits (0,1,...,9) are embedded at the LSB bits of the candidate attribute values. | same as [2]                            | Update attack                                 | No                     |                    |
|                                              | Rani et al. [17]             | Applies data flow analysis to identify variant and invariant part and embeds the watermark into invariant part applying [2]. | same as [2]                            |                                                | No                     |                    |
|                                              | Zhang et al. [18]            | Based on LSB modification for numerical data and space embedding, symbol modification, text modification for textual data. | 1. German credit risk dataset, 2. Dow Jones Industrial Average (DJIA) stock dataset, 3. Reddit World News Channel historical news headlines dataset real time database relation related to ticket assignment. | subset deletion, subset modification, subset addition | No                     |                    |
|                                              | Melkundi et al. [19]         | Watermarked both textual and numerical data and for numerical data, LSB bit flipping is performed. | 1. German credit risk dataset, 2. Dow Jones Industrial Average (DJIA) stock dataset, 3. Reddit World News Channel historical news headlines dataset real time database relation related to ticket assignment. | subset insertion, subset deletion, subset alteration | No                     |                    |
| Virtual Primary Key Based                    | Agrawal et al. [2]           | The VPK is generated by using only one attribute, and the rest of the attributes are selected to perform the WM embedding. | No experiments performed               | -                                             | No                     | Criteria 1         |
|                                              | Li et al. [22]               | MSB of more than one attribute of each tuple is used to generate virtual primary key. | same as [2]                            | bit flipping, subset, superset, attributed addition, deletion, modification, collision, invertibility, additive | Yes                    |                    |
|                                              | Chang et al. [23]            | Textual attributes are used to generate virtual primary keys.                     | Artificially generated database        | Tuple deletion, tuple alteration, tuple insertion | No                     |                    |
|                                              | Khanduja et al. [24]         | Similar to M-scheme of [22] but only two specific attributes are used for VPK generation. | No experiments performed              | subset deletion, subset alteration, attribute attack, tuple sorting, additive, invertibility, linear transformation | No                     |                    |
|                                              | Gort et al. [25]             | Focused on selecting each time different attributes and different bit for VPK generation. | same as [2]                            | Attribute deletion                            | No                     |                    |
|                                              | Gort et al. [26]             | Based on the cyclic model of the attribute                                         | same as [2]                            |                                                | No                     |                    |
|                                              | Gort et al. [27]             | Proposes double fragmentation of the watermark by using the existing redundancy in the set of virtual primary keys. | same as [2]                            |                                                | No                     |                    |

TABLE 4: Distortion-based Watermarking Techniques ('-' indicates 'not mentioned').
| Category                  | Paper                          | Brief Overview                                                                 | Data Set used                          | Attacks analysed                                                                 | Selected for experiment | Selection Criteria |
|--------------------------|-------------------------------|---------------------------------------------------------------------------------|----------------------------------------|----------------------------------------------------------------------------------|-------------------------|-------------------|
| Image as Watermark       | Hu et al. [31]                | Similar to [30] but not considered a fixed attribute to embed the watermark bits| Generated synthetic data               | subset addition, subset selection, subset alteration                              | No                      | Criteria 2        |
|                          | Zhang et al. [28]             | Insert a binary image watermark W into a relation in form of 0,1 sequences       | Real life KDD Cup 98 data set          | subset addition, subset alteration, subset deletion, modification attack          | Yes                     |                   |
|                          | Sardoudi et al. [34]          | image watermark embedded into numerical attributes                              | Randomly generated tuples              | subset selective, subset additive, subset modification, subset reverse order      | No                      |                   |
|                          | Wang et al. [30]              | Bits of a watermark image is embedded into bit positions of a fixed attribute of selected tuples in each group | Meteorological database                | subset update, subset delete                                                      | No                      |                   |
|                          | Huang et al. [29]             | Uses k-means clustering algorithm to locate the embedded watermark               | -                                      | selection, addition, alteration                                                   | No                      |                   |
|                          | Zhou et al. [32]              | Similar to [31] but image is divided into two parts, one part is used in grouping of tuples and another is used in embedding | Constructed own database               |                                                                                 | No                      |                   |
|                          | Al-Haj et al. [35]            | Based on inserting binary image watermarks in non-numeric multi-word attributes of selected database tuples. | Same as [2]                           |                                                                                 | No                      |                   |
|                          | Yige et al. [36]              | Transformed frequency coefficients to embed image watermark                      | Same as [2]                           |                                                                                 | No                      |                   |
|                          | Gort et al. [33]              | Spatial image watermarking technique is used                                    | -                                      |                                                                                 | No                      |                   |
|                          | Liu et al. [47]               | Hash function is used to generate the fingerprint.                              | -                                      |                                                                                 | No                      |                   |
|                          | Li et al. [48]                | Extends the proposal in [2] but a fingerprint is embedded                        | Same as [2]                           |                                                                                 | Yes                     | Criteria 3        |
|                          | Guo et al. [49]               | Proposes twice-embedding scheme. The fingerprint bits are embedded in the first round. In second round a pattern is embedded using the fingerprint as secret key. | Same as [2]                           |                                                                                 | Yes                     |                   |
|                          | Lafaye et al. [50]            | Watermarking is based on integer linear programming constraint solving.         | 1. Synthetic data 2. Forest CoverType data set | subset attacks, data alteration                                                   | No                      |                   |
|                          | Solami et al. [51]            | A buyer’s “thumb impression” is used as the fingerprint to be embedded.         | Rail system ticket pricing related data set | collision, tuple deletion, tuple alteration, insertion | No                      |                   |
|                          | Li et al. [22]                | Extends the proposal in [2] but embed meaningful fingerprint.                   | Flight scheduling database             | bit flipping, subset, attribute (insert, delete, modify), collusion, additive     | No                      |                   |
|                          | Gourmashbhand [45]            | Inserts new fake tuples as watermarks and properties of these watermarks are determined by probability distributions. | Flying scheduling database             |                                                                                 | No                      |                   |
|                          | Prasannakumari [46]           | A virtual attribute is added as the watermark. The aggregate function and parity checks is used to calculate the values of that attribute. | Sample data set                       |                                                                                 | Yes                     | Criteria 2        |

**TABLE 5:** (Continuation of Table 4) Distortion-based Watermarking Techniques (’-’ indicates ‘not mentioned’).
| Category          | Paper                  | Brief Overview                                                                 | Data Set used                                      | Attacks analysed                                           | Selected for experiment | Selection Criteria |
|-------------------|------------------------|-------------------------------------------------------------------------------|---------------------------------------------------|-----------------------------------------------------------|-------------------------|-------------------|
| **Partitioning**  | Kamran et al. [39]     | Data is partitioned by using hash function satisfying the usability constraints and each bit of the watermark is embedded in each partition. | biomedical and biomedicine data sets               | deletion                                                  | No                      |                   |
| Based             | Kamran et al. [40]     | Data is partitioned by using hash function and watermarks bits are embedded in each selected tuple. | Real life data set that shows power consumption rates same as [40] | deletion, insertion, alteration, multifaceted, collusion, additive deletion, alteration, insertion | Yes                     | Criteria 3        |
|                   | Shehab et al. [38]     | Data is partitioned using hash function and watermark bit is embedded by altering the partition statistics. |                                                   | subset selection, subset alteration, subset addition, attribute cutting | No                      |                   |
|                   | Huang et al. [41]      | Data partitioned using hash function and single bit of owner’s watermark is embedded in one group. | Walmart Sales database same as [2]               | Data loss attack, data alteration attack | No                      |                   |
|                   | Sion et al. [37]       | Data partitioned based on marker tuple and one bit of the watermark is embedded into one partition. |                                                   | Attribute cutting, subset addition, alteration, Attribute value modify, tuple insertion (single and multiple), tuple deletion (single and multiple), tuple modification (single and multiple) | No                      |                   |
|                   | Rao et al. [42]        | Data partitioned based on marker tuple and one bit of the watermark is embedded into one partition. |                                                   | Attribute value modify, tuple insertion (single and multiple), tuple deletion (single and multiple), tuple modification (single and multiple) | No                      |                   |
|                   | Guo et al. [43]        | Data partitioned by using hash function and two kind of watermarks: attribute watermark and tuple watermark. | National geochemical survey database of US.      | attribute reordering, subset deletion, subset alteration, subset insertion, linear transformation | No                      |                   |
|                   | Khanduja et al. [44]   | Improved hash partitioning using bacterial foraging algorithm.                |                                                   |                                                           | No                      |                   |
|                   |                                                                                         |                                                                                     |                                                   |                                                           |                         |                   |
| **Other**         | Huang et al. [41]      | Partitions tuple based on hash function and embeds watermark bits in single attribute. | -                                                 | subset selection, subset alteration, subset addition, attribute cutting | Yes                     | Criteria 3        |
| meaningful        | Guo et al. [52]        | Meaningful watermark information embedded.                                    | Wisconsin Diagnostic Breast Cancer data set      | subset selection, subset addition                        | No                      |                   |
| watermark         | Cui et al. [53]        | Meaningful watermark information embedded.                                    | Synthetic data                                   | Same as [52]                                             | No                      |                   |
| information       | Hu et al. [54]         | Random pseudo sequence number is used to identify the candidate attribute and the bit to be embedded. | Synthetic data                                   | tuple deletion, tuple insertion, tuple alteration       | No                      |                   |
|                   | Cui et al. [55]        | Meaningful watermark information embedded.                                    | Synthetic data                                   | attribute subsetting, attribute flipping, mix and match | No                      |                   |
|                   | David Gross-Amblard [56]| Query preserving approach.                                                    | RDBMS and XML doc                                |                                                           | No                      |                   |

TABLE 6: (Continuation of Table 4) Distortion-based Watermarking Techniques ('-' indicates 'not mentioned').
| Category               | Paper                      | Brief Overview                                                                                                                                                                                                 | Data Set used                                                                 | Attacks analysed                                                                 | Selected for experiment | Selection Criteria |
|------------------------|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|-------------------------------------------------------------------------------|------------------------|--------------------|
| Tuple Reordering       | Li et al. [57]             | Tuples are divided into groups by using hash function and for each tuple pair in a group, the order of the tuples are changed according to their tuple hash values and the watermark bit. | -                                                                            | insert, delete, value modification                                              | Yes                    | Criteria 1         |
|                        | Bhattacharya et al. [58]   | Extends the proposal in [57]                                                                                                                                                                                 | Real life data sets representing roads, regions and streams in the US database | tuple update, tuple deletion, tuple insertion                                   | No                     |                   |
|                        | Kamel [59]                 | Extends the proposal in [57]                                                                                                                                                                                  | -                                                                            | Deletion, insertion, modification                                              | No                     |                   |
|                        | Li et al. [60]             | Extends the proposal in [57]                                                                                                                                                                                  | Synthetic data                                                                | -                                                                            | No                     |                   |
|                        | Arun et al. [61]           | The tuple ordering is done on the basis of value of critical attribute(s) and re-arrangement is done relative to a secret initial order.                                                                    | -                                                                            | random value updation                                                           | No                     |                   |
|                        | Kamel et al. [62]          |                                                                                           |                                                                              |                                                                               |                        |                   |
|                        | Li et al. [63]             | MSBs of attribute values are used to generate the watermark.                                                                                                                                             | -                                                                            | attribute insertion, tuple insertion, value modification, deletion              | Yes                    | Criteria 3         |
|                        | Bhattacharya et al. [64]   | Extends the proposal in [63] but it considers private key instead of public.                                                                      | -                                                                            | tuple deletion, insertion                                                      | Yes                    |                   |
|                        | Halder et al. [65]         | Extends the proposal in [63]                                                                                                                                                                                  | -                                                                            | No                                                                            | No                     |                   |
|                        | Halder et al. [66]         | Extends the proposal in [63]                                                                                                                                                                                  | -                                                                            | No                                                                            | No                     |                   |
|                        | Bhattacharya et al. [67]   | A fixed number of MSBs and LSBs of the selected filed are used for generating the watermark.                                                                                                             | -                                                                            | No                                                                            | No                     |                   |
| Attribute Reordering   | Hamadou et al. [68]        | Attributes are virtually sorted on hash values of attribute names to define a secret initial order of attributes.                                                                                      | same as [2]                                                                  | attribute value updation, tuple insertion, tuple deletion                       | Yes                    |                   |
| Content characteristics Based | Camara et al. [69]     | Partitions the data into square matrices and matrix operations are used to generate the watermark.                                                                                                           | same as [2]                                                                  | Tuple deletion, tuple insertion, attribute insertion, multifaceted              | Yes                    |                   |
|                        | Khan et al. [70]           | Based on local characteristics like frequency distribution of digits, lengths and ranges of data values.                                                                                                    | same as [2]                                                                  | Insertion, update, deletion                                                   | Yes                    |                   |
|                        | Darwish et al. [71]        | After grouping of tuples, three fake tuples are generated for each groups: first tuple is generated based on hash function and to other tuples are created using genetic algorithm.                        | real database                                                                | subset deletion, subset addition, subset modification                           | No                     | Criteria 3         |
|                        | Rani et al. [72]           | Adapted MapReduce paradigm for watermarking large relational databases.                                                                          | same as [2]                                                                  | insertion, deletion, alteration modification, superset, deletion                | No                     |                   |
|                        | Siledar et al. [73]        | Generates an image from the database content.                                                                                                                                                    | same as [2]                                                                  | Insertion, deletion, alteration modification, superset, deletion                | No                     |                   |
|                        | Kamel et al. [74]          | Each column is organized into groups and the data values are reordered corresponding to a watermark value.                                                                                                       | Synthetic data and Forest CoverType data set                                   | Yes                                                                            | No                     |                   |
|                        | Naz et al. [75]            | Data values are grouped and the group watermark is generated by extracting MSBs of hash of attribute names.                                                                                              | Patient’s medical record                                                      | No                                                                            | No                     |                   |
|                        | Shah et al. [76]           | Watermarking is done by adjusting the text case of selected data values.                                                                                                                                    | US Medicare Plan 2008 database                                                | No                                                                            | No                     |                   |

TABLE 7: Distortion-free Watermarking Techniques ("-") indicates 'not mentioned').
### Table 8: Reversible Watermarking Techniques (‘-‘ indicates ‘not mentioned’).  

| Paper | Brief Overview | Data Set | Attacks analysed |
|-------|----------------|----------|------------------|
| Siledar et al. [77] | Based on quadratic difference expansion | Indian Liver Patient data set | Insertion, deletion, modification |
| Hou et al. [78] | Quality of watermarked data is used to claim copyright | Generated data | Tuple deletion, tuple addition, tuple alteration |
| Lin et al. [79] | Two different secret embedding keys are generated | Generated data | Alteration, deletion, mix-match, sorting, combination |
| Shen et al. [80] | Clustering-based and difference expansion technique is used | Generated data | Tuple delete, tuple modification |
| Li et al. [81] | Embeds the watermark bit by bit on the basis of grouping | Same as [2] | Insertion, deletion, modification |
| Lian et al. [82] | Differential expansion technology based on ant colony algorithm | Same as [2] | Subset deletion, modification |
| Li et al. [83] | Based on continuous columns in histogram | Same as [2] | Insert, delete, alter |
| Hamadou et al. [84] | Prediction-error expansion method | Same as [2] | Attribute Modification, tuple deletion, tuple insertion |
| Ge et al. [85] | Histogram shifting watermarking method | Wisconsin breast cancer diagnosis data set | Tuple addition, tuple deletion, attribute value modification |
| Tufail et al. [86] | Binary Bat Algorithm used for watermark creation | Heart disease medical data set | |
| Chai et al. [87] | Based on clustering grouping | Same as [2] | Attribute Modification or deletion, subset deletion, subset addition, subset alteration |
| Chai et al. [88] | Based on erasure code | Same as [2] | Attribute Modification or deletion, tuple deletion, tuple addition, tuple alteration |
| Wu et al. [89] | Difference-expansion reversible data hiding method is used | Protected numeric data | Insertion, deletion, modification |
| Li et al. [90] | Based on histogram gap low distortion | Same as [2] | Insertion, deletion and alteration |
| Hu et al. [91] | Genetic Algorithm with a new proposed Histogram Shifting of prediction error Watermarking (HSW) method to minimize distortion | Same as [2] | |
| Imamoglu et al. [92] | Difference expansion watermarking (DEW) with Firefly Algorithm is used to embed watermark | Same as [2] | Addition, deletion, bit-flipping, tuple-wise-multifaceted, attribute-wise-multifaceted, sorting |
| Chang et al. [93] | Watermark is embedded into the textual relational database | Textual relational database | Sorting, deletion, modification, addition |
| Chang et al. [94] | The content of textual attributes are used to generate the virtual primary attribute | Synthetic data | Tuple deletion, tuple alteration, tuple insertion |
| Itikhar et al. [95] | Genetic Algorithm is used for getting optimal watermark information | Heart disease medical data set | Insertion, deletion, alteration |
| Chang et al. [96] | Embeds the watermark into the fractional portion of the numerical attributes to minimize the distortion | Generated database | Alteration, deletion, mix-match, sorting |
| Jawad et al. [97] | Genetic algorithm is used to improve the capacity of difference expansion based watermarking in databases | Same as [2] | Addition, deletion, bit flipping, sorting, tuple-wise-multifaceted, attribute-wise-multifaceted, secondary watermarking |
| Farfoua et al. [98] | An identification image is converted into a stream of bits 0’s, and 1’s and embedded into numeric attributes | Synthetic data | Deletion, insertion, modification |
| Farfoua et al. [99] | Time-stamping protocol is used | Synthetic data | Tuple alteration, tuple deletion, mix and match, attribute deletion |
| Contreras et al. [100] | Based on a circular representation of a bijective transformation | Medical database | Modification of attribute values, elimination or insertion of tuples |
| Gupta et al. [101] | Difference expansion on integers is used to achieve reversibility | Generated database | Random bit wise flipping, subtractive, sorting, secondary watermarking |
| Gupta et al. [102] | Query-preserving watermarking scheme is proposed. | Generated database | - |
| Zhang et al. [103] | Based on expansion on data error histogram | Same as [101] | - |
| Gupta et al. [104] | Based on difference expansion | Synthetic data | Insertion, deletion and alteration |
| Unnikrishnan et al. [105] | Based on orthogonal learning particle swarm optimization | Same as [101] | |

[66] also, extend the approach of [63]. In [67], tuples are first grouped, then a fixed number of MSBs and LSBs of the selected attribute value are used to generate the watermark.

c: Attribute Reordering
Authors in [68] have proposed a fragile distortion-free watermarking technique based on the attribute reordering method. First, a secret initial order of attributes is defined by virtually sorting the attributes based on the hash of attribute names. Thereafter, the MSBs are extracted for generating the watermark.

d: Content Characteristics based watermarking
The watermarking approach in [70] generates the watermark based on the local characteristics like frequency distribution of various digits, lengths, and ranges of data values. In [69], the data set is grouped as the square matrices and the watermark is generated using the determinant and the minor of those square matrices.
We have classified the recent works in distortion-free watermarking in this category. The significant recent research works are proposed in [71]–[76]. In the proposed scheme in [71], some secure parameters are used to partition the tuples and three fake tuples are generated for each partition. A hash function is used to generate the first tuple. For the other two tuples, a genetic algorithm is used for numeric attributes, and for non-numeric attributes, the most frequent value is selected. These fake tuples are stored in a separate file, not inside the database, therefore making this approach distortion-free. Authors in [72] have adapted the MapReduce paradigm for watermarking of relational databases to decrease the computational cost and have implemented distortion-free algorithms in both sequential and MapReduce form. The proposal in [73] generates an image as a watermark from the database content. In [74], each column (attribute) is organized into new groups, each having $g$ data elements. The data elements in each group are re-ordered based on a watermark value. In [75], the data elements are grouped and the group watermark is generated by extracting $\mu$ MSBs of the hash of attribute names. They present the proposed watermarking as a service (WaaS) scheme.

A brief overview of different distortion-free watermarking techniques within each category is depicted in Table 7.

It is to observe that the reversible database watermarking techniques [4], [5], [77]–[105] as depicted in Table 8 have a wider scope of research and we would like to explore these techniques in the future separately.

Figure 2 provides a quick reference on the classification of different relational database watermarking algorithms.

## III. COMPARATIVE PERFORMANCE ANALYSIS OF DISTORTION-BASED WATERMARKING TECHNIQUES:

We select the distortion-based algorithms proposed in [2], [22], [28], [40], [41], [46], [48], [49] for the experimental analysis. We implement all the algorithms using Java. The experiments are performed on a server equipped with six core Intel Xeon Processor, 2.4 GHz Clock Speed, 128 GB RAM and Linux Operating System. We use benchmark data sets obtained by modifying the Forest CoverType data set into data sets of size 276MB, 532MB, 888MB, 1124MB, 1338MB, 1692MB, and 2237MB and perform the following analysis:

1) We analyze the usability of the watermarked databases in terms of differences between the mean and variance of attribute values, before embedding of watermark and after embedding.

2) We also analyze the watermark embedding and detection time by increasing the data set size.

3) We perform the robustness analysis of these techniques over various attacks, e.g., insertion, update, delete, zero out, and multifaceted attack.

The prime reason behind choosing this data set is its wide consideration by the majority of the proposals in the literature. In particular, 33 out of 94 research works considered this data set as their benchmark, whereas the rest of the proposals used either a different kind of real-world data or self-generated data which differ from one proposal to another. This makes it difficult to compare them empirically uniformly. In order to unify the comparative analysis, in this paper, we consider this most popular Forest CoverType data set as the benchmark for all the proposals under our consideration.

### A. COMPUTATIONAL TIME

In database watermarking, the time spent during watermark generation and detection is an important factor to consider. The watermark embedding and detection time for various approaches is shown in Table 9. The comparison of watermarking time for these techniques is depicted in Figure 3.

From Table 9 and Figure 3, we have the following observations:

1) For all algorithms, watermark embedding and detection time increase as the data size increases.

2) The watermark embedding and detection time are least in the case of [48].

3) The watermark embedding and detection time are highest in the case of [22].

4) The order of computational cost from lowest to highest is: [48] < [28] < [2] < [49] < [46] < [40] < [41] < [22].

There are many operations that may affect the computational time. We identify these operations as partitioning, hash calculation, random number generation, virtual primary key generation, updating the attribute value. Other parameters like the number of attributes and the number of tuples considered for watermark embedding also affect the computational time. We observe that the computational cost is highest in the case of [22] because it generates a virtual primary key for each of the tuples in the data set and therefore it takes longer time to execute.

### TABLE 9: Watermark Generation and Detection Time (in Milliseconds)

| Algorithm | N (MB) | $t_e$ | $t_d$ |
|-----------|--------|-------|-------|
| AMK 2002 | 276    | 44559 | 83353 |
| Li 2003  | 532    | 19159 | 26256 |
| Pramana 2004 | 888 | 124285 | 119988 |
| Zhang 2011 | 1124  | 178702 | 138994 |
| Karmen 2013 | 158994 | 885012 | 859342 |
| Peeten 2006 | 2237  | 223705 | 173126 |
| Li 2005  | 276    | 119660 | 103773 |
| Huang 2004 | 33010 | 164547 | 19883 |
| e: Others | 228411 | 142511 | 923584 |
|         | 158994 | 885012 | 859342 |
|         | 1124   | 178702 | 138994 |
|         | 532    | 19159 | 26256 |
|         | 276    | 44559 | 83353 |

1) https://kdd.ics.uci.edu/databases/covertype/covertype.html

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
more time. Whereas, the approach in [48] has least computational cost as it generates some random numbers instead of hash computation. The computational cost is less in the case of [2], [28]. In the case of [28], after partitioning only some of the tuples and attributes satisfying a criterion are considered for embedding the watermark. In the case of [2], the tuples are not partitioned, but only a fraction (γ) of tuples satisfying a particular condition are considered. The LSBs of one attribute in each selected tuple are flipped based on the watermark bits. The approach in [49] also considers only one fixed attribute in each partition to embed the watermark. Partitioning is a common operation in case of [40], [41], [46], [49] and all the approaches are having more computational cost after the approach in [22]. Therefore, the partitioning operation is affecting the computational cost.

B. USABILITY OF DATA AFTER WATERMARK EMBEDDING

The usability of the database is based on the domain, e.g., a minor change in a voter database can create a problem, and hence the watermarking should not cause any changes to the voter database, whereas, minor changes in a forest survey database can be tolerated. Therefore, it is difficult to generalize the criteria for usability. However, Table 10 will help the users to understand the effect of watermark embedding on the mean and variance values of the attributes and give them an idea about whether watermarking causes more changes in the underlying data or not. Table 10 shows the change in variance of database values before embedding of watermark and after embedding. The watermark embedding algorithms in [2], [22], [28], [41], [49] embed the watermark bit in a particular bit of the attribute value. The number of bits available for watermark embedding is denoted by the variable ξ. We compute the variance of each attribute by varying the value of ξ to 3, 5, and 8. We observe that there is no change in mean after watermark embedding.

In distortion-based watermarking, it is assumed that a certain level of distortion is tolerable. In the case of [2], there are small changes in variance when the bits available for embedding are increased to 8 bits. In the case of [46] and [48], there is no change in the variance at all, whereas in the case of [22] and [28], the variance is highly affected after the watermark embedding.

From figure 10, we observe that in the case of approaches in [40], [41], [49] the variance of only one attribute is affected after watermark embedding because they only consider a single attribute to embed the watermark. In the case of [46], there is no change or negligible change in the variance as it inserts a new tuple into the database relation. Therefore, it does not cause any change in the attribute values and the variance of attributes is not affected. Similarly, in the case of [48], there is no change in the variance and in the case of [2], there is negligible change as both of the algorithms embed the watermark into a fraction of tuples. The usability is highly affected in the case of [28] because an attribute is selected for embedding if the length of the attribute value is greater than a particular value. This causes the watermark to be embedded in more than one attributes.

C. ROBUSTNESS ANALYSIS

We perform the robustness analysis of the watermarking techniques over various attacks, e.g. insertion, update, delete,
zero out, and multifaceted attack. We analyze the rate of detection by varying the intensity of the attacks as 10%, 30%, 50%, 70%, and 90%.

1) Delete Attack
In delete attack, the attacker deletes some of the tuples of the watermarked database in order to distort the embedded watermark. Though the attacker is supposed to delete the tuples keeping in mind the usability of the data, we analyze the detection rate by varying the attack percentage from 10% to 90%. The rate of detection for various distortion-based techniques after delete attack is shown in Figure 4(a).

From Figure 4(a), we can observe that the rate of watermark detection remains more than 90% even after 90% delete in case of [2], [22], [28], [41], [48].

2) Update Attack
In an update attack, the attacker randomly updates some of the values of the watermarked database with his own values and try to claim ownership of the database. We analyze the detection rate by varying the update percentage from 10% to 90% as depicted in Figure 4(b).

We can observe from Figure 4(b) that the rate of detection is more than 80% in case of [2], [22], [28], [40] even after 90% update.
90% update.

3) Insertion Attack

In an insertion attack, the attacker removes a particular number of tuples from the watermarked database and inserts the same number of tuples into the database to destroy the embedded watermark. The rate of watermark detection for various techniques after insertion attack is shown in Figure 4(c). We can observe from Figure 4(c) that the rate of detection is 100% in the case of [22], [28], [48] even after a 90% attack.

4) Zero-out Attack

In this attack, some tuple values are selected randomly and updated with zero by the attacker to destroy the embedded watermark. We analyze the rate of watermark detection by varying the attack percentage as shown in Figure 4(d).

We can observe from Figure 4(d) that the rate of detection is more than 80% in case of [2], [22], [28] even after 90% attack.

5) Multifaceted Attack

This is the combination of delete, update, and insertion attacks. The attacker randomly updates some of the tuple values, deletes some of the tuples, and inserts his own tuples to destroy the embedded watermark.

| Multifaceted attack | Delete Attack | Update attack | Insertion attack |
|--------------------|---------------|---------------|-----------------|
| 10%                | 3%            | 3%            | 4%              |
| 30%                | 10%           | 10%           | 10%             |
| 50%                | 16%           | 16%           | 18%             |
| 70%                | 20%           | 20%           | 30%             |
| 90%                | 30%           | 30%           | 30%             |

TABLE 11: Intensity of Attacks.

The robustness against various attacks is more in the case of [2] and [48] since the detection in [2] is based on the match counts that are computed on the remaining watermarked tuples after the attack. Similarly, in the case of [48], the detection is based on the majority voting for each fingerprint bit and form the remaining watermarked tuples after attacks, the fingerprint can be recovered.

IV. COMPARATIVE PERFORMANCE ANALYSIS OF VARIOUS DISTORTION-FREE WATERMARKING TECHNIQUES

The data values in the database are not changed in the case of distortion-free watermarking techniques. These techniques mainly generate the watermarks from the database contents. The primary phases in these techniques are (i) Partitioning of tuples into groups, and (ii) Watermark generation from each group. The watermarks for each group can be combined together to generate the watermark for the whole database.

A generic distortion-free watermarking algorithm GEN_WM is shown in Algorithm 1. The database relation R is taken as input. Steps 1 to 3 compute the group id \( g_i \) to which a tuple \( t \) belongs by applying function \( f \) (e.g. a hash function). In Steps 4 to 6, a group watermark \( W_i \) for the group \( g_i \) is generated by using the tuples belonging to group \( g_i \). Step 8 computes the overall watermark \( W \) by performing a suitable operation \( || \) (e.g. a concatenation operation) to the group watermarks.

Authors in [57] proposed the first work in this domain. We classify the distortion-free watermarking techniques in the following categories: (i) permutation of tuples, (ii) converting database relation into binary form, (iii) attribute reordering, (iv) content characteristics based, and (v) others. We analyze these techniques and select the algorithms for experimental analysis on the basis of the same criteria as discussed in Section II.

We consider the distortion-free watermarking algorithms proposed in [57], [63], [64], [68]–[70], [75] and perform the robustness analysis of these techniques over various attacks, e.g. insertion, update, delete, zero out, and multifaceted attack. We also analyze the computational cost. We implement all the algorithms using Java. The experiments are performed on a server equipped with six-core Intel Xeon Processor, 2.4 GHz Clock Speed, 128 GB RAM, and Linux Operating System. We use benchmark data sets obtained by modifying

VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
The Forest CoverType data set\(^2\) into data sets of size 276MB, 532MB, 888MB, 1124MB, 1338MB, 1692MB and 2237MB. The reason for choosing this data set is discussed in Section III.

A. COMPUTATIONAL TIME

In database watermarking, the time spent during watermark generation and detection is an important factor to consider. The watermark generation and detection time for various approaches is shown in Table 12. The comparison of watermarking time for these techniques is depicted in Figure 6.

![Watermark Generation Time](image1.png)

(a) Watermark Generation Time

![Watermark Detection Time](image2.png)

(b) Watermark Detection Time

FIGURE 6: Comparison of Watermark Generation and Detection Time for Distortion-free techniques.

Following are the observations from Table 12 and Figure 6:

1) For all the watermarking approaches, watermark embedding and detection time increases as the data size increases.

2) Watermark generation and detection time is least in case of [70] and highest in case of [64].

Authors in [72] adapted the MapReduce paradigm to watermark relational databases. They have implemented the algorithms proposed in [57], [64], [68], [69], [70] in sequential as well as MapReduce form and it was observed that as the data size increases, the percentage reduction in watermarking time increases from sequential to MapReduce.

In the case of distortion-free watermarking techniques, there are various operations that affect the computational cost, e.g. hash computation, partitioning, watermark generation, pseudo-number generation, matrix operations, etc. The number of attributes, tuples, and the bit positions available for watermark generation also affects the computational cost. From Figure 6, we can observe that the computational time is highest in the case of [64] since it partitions all attributes of all tuples for generating a binary form of the relational database. The computational cost is least in case of [70], since it does not partition the database relation. The watermark is generated by considering all attributes of all tuples and by generating digit, length, and frequency sub-watermarks. The basic step in the case of distortion-free technique is partitioning. For example, the approaches in [57], [64], [68], [69], [75] partition the data based on either hash function, pseudo-random number, etc. The group watermarks are then generated independently.

B. USABILITY OF DATA AFTER WATERMARK GENERATION

In the case of distortion-free watermarking approaches, the watermark is generated from the underlying content of the data and there is no distortion in the data itself, hence the data usability is not affected.

C. ROBUSTNESS ANALYSIS

We perform the robustness analysis of the watermarking techniques over various attacks, e.g. insertion, update, delete, zero out, and multifaceted attack. We analyze the rate of detection by varying the intensity of the attacks from 10% to 90%.

1) Delete Attack

In a delete attack, some of the tuples of the watermarked database are deleted by the attacker in order to distort the watermark. Though the attacker is supposed to delete the tuples keeping in mind the usability of the data, we analyze the detection rate by varying the attack percentage from 10% to 90%. The rate of detection for various distortion-based techniques after delete attack are shown in Figure 7(a). From Figure 7(a), we observe that the rate of detection remains 100% in case of [63] even after 90% attack.

2) Update Attack

In an update attack, the attacker randomly updates some of the values of the watermarked database with its own values and tries to claim ownership of the database. We analyze the detection rate by varying the update percentage from 10% to 90% as shown in Figure 7(b). We observe that the rate of detection remains 100% in the case of [63] even after a 90% attack.

\(2\)https://kdd.ics.uci.edu/databases/covertype/covertype.html

### TABLE 12: Watermark Generation and Detection Time (in Milliseconds).

| Data Set Size (MB) | Li 2004 [57] | KhanHusain [70] | Li Deng 2006 [63] | Hamadou2011 [68] | Camara2014 [69] | Bhat2009 [64] | Naz 2019 [75] |
|-------------------|--------------|-----------------|-------------------|-----------------|----------------|--------------|--------------|
| 276               | 35650        | 35602           | 3999              | 39028           | 39179          | 49708        | 52609        |
| 532               | 19383        | 38445           | 14592             | 15710           | 37035          | 37098        | 42248        |
| 888               | 14687        | 97114           | 23915             | 26082           | 91723          | 82103        | 241412       |
| 1124              | 152515       | 98501           | 27054             | 12862           | 128935         | 248656       | 245674       |
| 1538              | 215858       | 125221          | 38276             | 35903           | 129615         | 284090       | 256484       |
| 1892              | 223353       | 141930          | 43006             | 40990           | 204996         | 192296       | 340214       |
| 2437              | 398133       | 379725          | 60735             | 67350           | 219622         | 208221       | 320238       |

**FIGURE 6:** Comparison of Watermark Generation and Detection Time for Distortion-free techniques.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3157866, IEEE Access
3) Insertion Attack

In an insertion attack, the attacker removes a particular number of tuples from the watermarked database and inserts the same number of tuples into the database to destroy the watermark. The rate of watermark detection for various techniques after insertion attack is depicted in Figure 7(c). We observe that the rate of detection remains 100% in the case of [63] even after 90% attack.

4) Zero out Attack

Some of the tuple values of the watermarked database are randomly selected by the attacker and updated with zero to destroy the watermark. We analyze the rate of watermark detection by varying the attack percentage as shown in Figure 7(d). The rate of detection even after a 90% attack is highest in the case of [63].

5) Multifaceted Attack

This is the combination of delete, update, and insertion attacks. The attacker randomly updates some of the tuple values, deletes some of the tuples, and inserts his own tuples to distort the watermark.

FIGURE 7: The rate of detection after various attacks in case of distortion-free techniques.

FIGURE 8: Rate of detection after Multifaceted Attack in case of distortion-free algorithms.
The intensity of the update, delete, and insertion attacks are taken as shown in Table 11. We analyze the rate of detection after the multifaceted attack in Figure 8. The rate of detection remains near 100% in the case of [63] even after a 90% attack.

From Figure 7 and Figure 8, we observe that the approach in [63] has the highest robustness against four types of attacks since it considers the number of attributes as that of binary attributes ($\gamma$) present in the database relation. It generates the watermark bits from the MSB positions of the attribute values. If the value of $\gamma$ is increased, though it will increase the robustness, the computational cost will be increased.

V. DISCUSSION W.R.T. THE EXISTING EXPERIMENTAL OBSERVATIONS

While comparing our evaluation-results with the results reported in the existing papers, we draw the following observations:

1) Although watermark-embedding and detection times have significance when to apply in case of large-scale data set, none of the existing proposals (except [2]) under distortion-based approaches performs this evaluation. To the best of our knowledge, this paper first reports a detailed comparative study on the computational costs incurred by different algorithms under consideration. Under distortion-free approaches, only one proposal [75] evaluates its performance on patient’s medical data achieving the watermark embedding and detection time of 13 and 21.1 seconds respectively. Even though [75] considers data set different from CoverType, we observe a linear growth of computational time in both embedding and detection phases similar to [2], [75].

2) Experimental evaluation on data usability using CoverType data set is being conducted by the authors in [2], [48], [49], whereas [40] considers a data set comprising consumers’ power consumption rates. Like the results reported in [2], [48], [49], our evaluation results also reveals the similar fact that there is no notable change in the mean value of the data after watermark embedding, while very little change in the range 1-99 is observed in case of variance when more number of LSBs (e.g., 8 bits) for watermark embedding is considered. On the other hand, when we conduct experiments for the algorithms in [40] on CoverType data set, we observe a significant increase in the variance and little decrease in the mean values than that reported in [40] on power consumption data. This is due to the difference in the semantic domains of the attributes used for watermark-embedding in case of two different data sets. Note that distortion-free approaches do not suffer from this issue.

3) Attack analyses to manifest the robustness of the algorithms are being conducted over CoverType data set in [2], [22], [48], [49], [68]–[70]. Interestingly, we gain a similar experience in our results also. To be more precise, in both the cases, the results show that the watermark can be detected even after a 90% attack. On the other hand, the attack analysis of the algorithms in [28], [40], [75] are performed on data sets different from CoverType. This is worthwhile to mention that the result reported in [28], [40] is similar to the result obtained using CoverType in our case, which shows that the watermark detection rate is above 70% even after a 90% attack. Similarly, the attack result reported in [75] exhibits similar trend as we observe in our case (on CoverType data), which show that the watermark-detection rate drops below 20% after a 90% attack.

VI. OVERALL RESEARCH GUIDANCE

In this section, we discuss in detail the guidance to the users for a wise decision on choosing the right watermarking technique. We observe that various operations and parameters (such as the number of attributes, tuples, and bit positions for embedding or generation) in the watermarking algorithms impact the computational cost, data usability, and robustness. Few observations are listed below:

- **The number of attributes involved in watermark embedding**: Some algorithms embed the watermark in all attributes of the database relation. Even though this increases the robustness, this may cause more distortion and may affect the usability with increased computational time.

- **The number of tuples considered for watermark embedding**: If all of the tuples are considered for embedding the watermark, then it will increase the computational cost. It will also affect the usability more, though the robustness may be increased. Whereas, some watermarking algorithms consider a fraction of tuples for embedding the watermark. This will decrease the computational cost and the data usability will be less affected.

- **The number of bits available for watermarking**: If the number of bits considered for embedding watermarks is increased, it will increase the distortion. The computational time and robustness will not be affected much by this.

- **Parameters particularly affecting the computational cost**: There are many operations that may affect the computational time. We identify these operations as: partitioning, hash calculation, random number generation, virtual primary key generation, matrix operations, updating the attribute value.

Although we can not generalize, we categorize the usability, computational time, and robustness towards attacks for the relative comparison of various watermarking techniques in the following groups:

$$\text{UsabilityAffected} = \begin{cases} \text{Very High, if } \Delta \text{Variance} > \pm 10 \text{ in } 2 \text{ attribute} \\ \text{High, if } \Delta \text{Variance} \in [\pm 5 \text{ to } \pm 10] \text{ in } 1 \text{ or } 2 \text{ attribute} \\ \text{Less, if } \Delta \text{Variance} = 0 \text{ or } \leq \pm 5 \text{ in } 1 \text{ or } 2 \text{ attribute} \end{cases}$$
The \( \Delta \)Variance represents the change in variance of the attribute values after the embedding of the watermark. A comparative summary of the distortion based algorithms that we have considered for the experimental analysis is shown in Table 13.

![Table 13: A Comparative Summary of Distortion-based Techniques.](image)

The best algorithm should affect the usability “Less” after watermark embedding, have “Less” computational cost, and have “Very High” robustness against various attacks. In case of the distortion-based algorithms, if the usability is the main concern then the approaches in [2], [46], [48] are the better options since the attributes are having no change or negligible change in the variance after watermark embedding. If we consider the robustness and computational cost, then [2], [48] are better, but the approach in [46] has less robustness against all types of attacks. If only computational cost is considered, then the approach in [48] is having the least computational cost. If only robustness is considered, then the approach in [22] is the most robust, but the usability is highly affected after embedding. The computational cost is also highest in the case of [22] since it computes a virtual primary key for each of the tuples.

From Table 13, we can observe the following in the case of both [2] and [48]:

- The data usability is least affected after the watermark embedding.
- The computational cost is “Less”.
- “Very High” robustness against three kinds of attacks.

Therefore, considering the usability constraints as defined, the computational cost and the robustness towards various attacks, we can say that the watermarking algorithms in [2] and [48] perform better than the other distortion-based watermarking algorithms we have considered for experimental analysis.

A comparative summary of the distortion-free algorithms that we have considered for the experimental analysis is shown in Table 14.

![Table 14: A Comparative Summary of Distortion-free Techniques.](image)

Overall, considering the above-mentioned facts, the watermarking algorithm in [63] performs better than the other distortion-free watermarking algorithms in terms of computational-overhead and robustness.

### VII. CONCLUSION

In this paper, we perform a detailed comparative analysis of various relational database watermarking techniques empirically. We classify the existing distortion-based watermarking techniques into six categories, namely (i) meaningless bit-pattern as the watermark, (ii) virtual primary key based, (iii) image as watermark, (iv) partitioning based, (v) fake tuple/attribute insertion, (vi) fingerprinting techniques, and (vii) other meaningful watermark information. Similarly, the existing distortion-free techniques are classified as (i) permutation of tuples, (ii) conversion of the database into binary form, (iii) attribute reordering, (iv) content characteristics based, and (v) others. We perform an exhaustive empirical study and comprehensive analysis of a number of algorithms selected based on our quality-criteria. In particular, our evaluation focuses the following three crucial factors: computational cost, data usability, and robustness, as a way to provide an insightful guidance to choose the right watermarking technique for a given application.

### REFERENCES

1. Sanjeev Khanna and Francis Zane. Watermarking maps: hiding information in structured data. In Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms, pages 596–605, 2000.

2. Rakesh Agrawal and Jerry Kiernan. Watermarking relational databases. In VLDB ’99: Proceedings of the 25th International Conference on Very Large Databases, pages 155–166. Elsevier, 2002.

3. Raju Halder, Shantanu Pal, and Agostino Cortesi. Watermarking techniques for relational databases: Survey, classification and comparison. J. Univers. Comput. Sci., 16(21):3164–3190, 2010.

4. Asifullah Khan, Ayesha Siddiqa, Summuyya Munib, and Sana Ambreen Malik. A recent survey of reversible watermarking approaches. Sciences, 279:251–272, 2014.

5. Saman Iftekhar, M Kamran, and Zahid Anwar. A survey on reversible watermarking techniques for relational databases. Security and communication networks, 8(15):2580–2603, 2015.
18

[6] Ming-Ru Xie, Chia-Chun Wu, Jau-Ji Shen, and Min-Shiang Hwang. A survey on data distortion watermarking relational databases. Int. J. Netw. Secur., 18(6):222–229, 2016.

[7] Satyendra Kumar, Binod Kumar Singh, and Mohit Yadav. A recent survey on multimedia and database watermarking. Multimedia Tools and Applications, 79(27):20149–20197, 2020.

[8] Muhammad Kamran and Muddassar Farooq. A comprehensive survey of watermarking relational databases. arXiv preprint arXiv:1801.08271, 2018.

[9] Abd S Alfagi, A Abd Manaf, BA Hamida, S Khan, and Ali A Elrowayati. Survey on relational database watermarking techniques. ARPN-JEAS, 11:422–423, 2016.

[10] Asmaa Alqassab and Mafaz Alanezi. Relational database watermarking techniques: A survey. In Journal of Physics: Conference Series, volume 1818, page 012185. IOP Publishing, 2021.

[11] Vidhi Khanduja. Database watermarking, a technological protective measure: Perspective, security analysis and future directions. Journal of information security and applications, 37:38–49, 2017.

[12] Shipin Wu, Dong-Ping Li, Jian-Pin Hu, and Jerry Kiernan. A system for watermarking relational databases. In Proceedings of the 2003 ACM SIGMOD international conference on Management of data, pages 674–674, 2003.

[13] Rakesh Agrawal, Peter J Haas, and Jerry Kiernan. Watermarking relational data: framework, algorithms and analysis. The VLDB journal, 12(2):157–169, 2003.

[14] Zhu Qin, Yang Ying, LE Jia-jin, and LUO Yi-shu. Watermark based approach of database watermarking relational databases using optimization-based techniques. IEEE transactions on knowledge and data engineering, 16(12):1509–1525, 2004.

[15] Song Yige, Liu Weidong, Song Jiaxing, and Wong Ming Sze Angela. Minimum distortion: High capacity watermarking technique for relational data. In Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security, pages 111–121, 2017.

[16] Huiping Guo, Yingjiu Li, Anyi Liu, and Sushil Jajodia. A fragile watermark mechanism for relational data. In The Fourth International Conference on Multimedia and Database watermarking, pages 111–121, 2008.

[17] Radu Sion, Mikhail Atallah, and Sunil Prabhakar. Rights protection for relational data using once-for-all usability constraints. IEEE Transactions on Knowledge and Data Engineering, 25(12):2694–2707, 2013.

[18] Xiangzhou Hou, Zaihui Cao, and Jianhua Sun. An image based algorithm for watermarking relational databases. In Proceedings of the 2007 ACM symposium on Applied computing, pages 254–258, 2007.

[19] Maikel Lázaro Pérez, Claudia Feregrino-Uribe, Agostino Cortesi, and Felix Fernández-Peña. A double fragmentation approach for improving virtual primary key-based watermark synchronization. IEEE Access, 6:81504–61516, 2020.

[20] Lisheng Zhang, Wei Gao, Nan Jiang, Liqiu Zhang, and Yan Zhang. Relational databases watermarking for textual and numerical data. In 2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC), pages 1633–1636. IEEE, 2011.

[21] Kaiyin Huang, Min Yue, Pengfei Chen, Yanhsan He, and Xiaoyn Chen. A cluster-based watermarking technique for relational database. In 2009 First International Workshop on Database Technology and Applications, pages 107–110. IEEE, 2009.

[22] Chaokun Wang, Jianmin Wang, Ming Zhou, Guisheng Chen, and Deyi Li. Atbam: An arnold transform based method on watermarking relational data. In 2008 International Conference on Multimedia and Ubiquitous Engineering (MUE 2008), pages 263–270. IEEE, 2008.

[23] Zhongyan Hu, Zaihui Cao, and Jianhua Sun. An image based algorithm for watermarking relational databases. In 2009 International Conference on Measuring Technology and Mechatronics Automation, volume 1, pages 425–428. IEEE, 2009.

[24] Xiaogang Lai, Xianlai Wang, and Jianxin Zou. An add-only-attack proof watermarking mechanism for databases’ copyrights protection using image. In Proceedings of the 2007 ACM symposium on Applied computing, pages 254–258, 2007.

[25] Maikel L Pérez-Gort, Claudia Feregrino-Uribe, and Jyrki Nummenmaa. A minimum distortion: High capacity watermarking technique for relational data. In Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security, pages 111–121, 2017.

[26] Hossein Moradian Sardroudi and Subarib Ibrahim. A new approach for relational database watermarking using image. In 5th International Conference on Computer Sciences and Convergence Technology, pages 606–610. IEEE, 2010.

[27] Ali Al-Haj and Ashraf Odeh. Robust and blind watermarking of relational database systems. 2008.

[28] Guoguang Wang, Liu Weidong, Song Jinxiang, and Wong Ming Sze Angola. Det transform based relational database robust watermarking algorithm. In 2010 Second International Symposium on Data, Privacy, and E-Commerce, pages 61–65. IEEE, 2010.

[29] Radu Sion, Mikhail Atallah, and Sunil Prabhakar. Rights protection for relational data. IEEE transactions on knowledge and data engineering, 16(12):1509–1525, 2004.

[30] Mohammad Shehab, Elisa Bertino, and Arif Ghafoor. Watermarking relational databases using optimization-based techniques. IEEE transactions on Knowledge and Data Engineering, 20(1):116–129, 2007.

[31] M Kamran and Muddassar Farooq. A formal usability constraints model for watermarking of outsourced datasets. IEEE transactions on information forensics and security, 8(6):1061–1072, 2013.

[32] Mohammad Kamran, Sabah Suhal, and Muddassar Farooq. A robust, distortion minimizing technique for watermarking relational databases using once-for-all usability constraints. IEEE Transactions on Knowledge and Data Engineering, 25(12):2694–2707, 2013.

[33] Min Huang, Jiaheng Cao, Zhiyang Peng, and Ying Fang. A new watermark mechanism for relational data. In The Fourth International Conference onComputer and Information Technology, 2004. CIT’04., pages 946–950. IEEE, 2004.

[34] Burepalli VS Rao and Munaga VNK Prasad. Subset selection approach for watermarking relational databases. In International Conference on Data Engineering and Management, pages 181–188. Springer, 2010.

[35] Huiping Guo, Yingjiu Li, Anyi Liu, and Sushil Jajodia. A fragile watermarking scheme for detecting malicious modifications of database relations. Information Sciences, 176(10):1350–1378, 2006.

[36] Vidhi Khanduja, Onkar Pashask Verma, and Shampa Chakravorty. Watermarking relational databases using bacterial foraging algorithm. Multimedia Tools and Applications, 74(3):813–839, 2015.

[37] Vahab Pourmoghaddam. A new watermarking approach for relational data. In Proceedings of the 46th annual southeast regional conference on XX, pages 127–131, 2008.

[38] Vahab Prasanna Kumari. A robust tamperproof watermarking for data integrity in relational databases. Research Journal of Information Technology, 1(3):115–129, 2009.

[39] Siyuan Liu, Shuhong Wang, Robert H Deng, and Weizhong Shao. A virtual primary key generation approach for watermarking relational data. Expert Systems with Applications, 138:112770, 2019.
tional conference on information security and cryptology, pages 455–466. Springer, 2004.

[48] Yingjiu Li, Vipin Sarupar, and Sushil Jajodia. Fingerprinting relational databases: Schemes and specialties. IEEE Transactions on Dependable and Secure Computing, 2(1):34–45, 2005.

[49] Fei Guo, Jianmin Wang, and Deyi Li. Fingerprinting relational databases. In Proceedings of the 2006 ACM symposium on Applied computing, pages 487–492, 2006.

[50] Julien Lafaye, David Gross-Amblard, Camelia Constantin, and Meryem Guerrouani. Watermarking: An optimized fingerprinting system for databases under constraints. IEEE Transactions on Knowledge and Data Engineering, 20(4):532–546, 2008.

[51] Eesa Al Solami, Muhammad Kamran, Mohammed Saeed Alkhaterei, Fouzia Rafiq, and Ahmed S Alghamdi. Fingerprinting of relational databases for stopping the data theft. Electronics, 9(7):1093, 2020.

[52] Fei Guo, Jianmin Wang, Zhihao Zhang, Xiaojun Ye, and Deyi Li. An improved algorithm to watermark numeric relational data. In International Workshop on Information Security Applications, pages 138–149. Springer, 2005.

[53] XinChun Cui, Xiaolin Qin, Gang Sheng, and Jiping Zheng. A robust algorithm for watermark numeric relational databases. In Intelligent Control and Automation, pages 810–815. Springer, 2006.

[54] Tian-Lei Hu, Gang Chen, Ke Chen, and Jin-Xiang Dong. Garwm: towards a generalized and adaptive watermark scheme for relational data. In International Conference on Web-Age Information Management, pages 380–391. Springer, 2005.

[55] XinChun Cui, Xiaolin Qin, and Gang Sheng. A weighted algorithm for watermarking relational databases. Wuhan university journal of natural sciences, 12(1):79–82, 2007.

[56] David Gross-Amblard. Query-preserving watermarking of relational databases and xml documents. ACM Transactions on Database Systems (TODS), 36(1):1–24, 2011.

[57] Yingjiu Li, Huiping Guo, and Sushil Jajodia. Tamper detection and localization for categorical data using fragile watermarks. In Proceedings of the 4th ACM workshop on Digital rights management, pages 73–82, 2004.

[58] Sukriti Bhattacharya and Agostino Cortesi. A distortion free watermark framework for relational databases. In ICSOFT (2), pages 229–234. Citeseer, 2009.

[59] Ibrahim Kamel. A scheme for protecting the integrity of databases, computers & security, 28(7):698–709, 2009.

[60] Min Li, Wenyue Zhao, and Jie Guo. An asymmetric watermarking scheme for relational database. In 2011 IEEE 3rd International Conference on Communication Software and Networks, pages 180–184. IEEE, 2011.

[61] R Arun, K Praveen, Divya Chandra Bose, and Hiran V Nath. A distortion free relational database watermarking using patch work method. In Proceedings of the International Conference on Information Systems Design and Intelligent Applications 2012 (INDIA 2012) held in Visakhapatnam, India, January 2012, pages 531–538. Springer, 2012.

[62] Ibrahim Kamel, Maha Alaeddin, Waheeb Yaqub, and Karrem Kamel. Distortion-free fragile watermark for relational databases. International Journal of Big Data Intelligence, 3(3):190–201, 2016.

[63] Yingjiu Li and Robert Huijie Deng. Publicly verifiable ownership protection for relational databases. In Proceedings of the 2006 ACM Symposium on information, computer and communications security, pages 78–89, 2006.

[64] Sukriti Bhattacharya and Agostino Cortesi. A generic distortion free watermarking technique for relational databases. In International Conference on Information Systems Security, pages 252–264. Springer, 2009.

[65] Raju Halder and Agostino Cortesi. A persistent public watermarking of relational databases. In International Conference on Information Systems Security, pages 216–230. Springer, 2010.

[66] Raju Halder and Agostino Cortesi. Persistent watermarking of relational databases. In Proceedings of the IEEE International Conference on Advances in Communication, Network, and Computing (CNC’10), pages 46–52, 2010.

[67] Sukriti Bhattacharya and Agostino Cortesi. Distortion-free authentication watermarking. In International Conference on Software and Data Technologies, pages 205–219. Springer, 2010.

[68] Ali Hamadou, Xingming Sun, Lingyun Gao, and Saeed A Shah. A fragile zero-watermarking technique for authentication of relational databases. International Journal of Digital Content Technology and its Applications, 5(5), 2011.

[69] Lamine Camara, Junyi Li, Renfa Li, and Wenyong Xie. Distortion-free watermarking approach for relational database integrity checking. Mathematical problems in engineering, 2014, 2014.

[70] Aibah Khan and Syed Afaiz Hasan. An efficient zero watermark scheme to detect and characterize malicious modifications in database relations. The Scientific World Journal, 2013, 2013.

[71] Saad M Darwish, Hosam A Selim, and Mohamed M Elsherbiny. Distortion-free database watermarking system based on intelligent mechanism for content integrity and. Journal of Computers, 2018.

[72] Sapana Rani, Dileep Kumar Kosshy, and Raju Halder. Adapting mapreduce for efficient watermarking of large relational dataset. In 2017 IEEE Trustcom/BigDataSE/ICCSS, pages 729–736. IEEE, 2017.

[73] Seema Sileedar and Sharvari Tamane. A distortion-free watermarking approach for verifying integrity of relational databases. In 2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC), pages 192–195. IEEE, 2020.

[74] Ibrahim Kamel, Waheeb Yaqub, and Zeyar Aung. Distortion-free watermarking scheme for compressed data in columnar database. Proceedings of the 15th ICETE. Porto, Portugal, pages 343–353, 2018.

[75] Farah Naz, Abid Khan, Mansoor Ahmed, Majid Iqbal Khan, Sadia Din, Awais Ahmad, and Gwaggon Je. Watermarking as a service (waas) with anonymity. Multimedia Tools and Applications, 79(23):16051–16075, 2020.

[76] Saeed Arif Shah, Imran Ali Khan, Syed Zaki Hassan Kazmi, and Fariza Hanum Binti Md Nasseruddin. Semi-fragile watermarking scheme for relational database tamper detection. Malaysian Journal of Computer Science, 34(1):1–12, 2021.

[77] Seema Babusigam Sileedar and Sharvari Tamane. Quadratic difference expansion based reversible watermarking for relational database. Journal of Integrated Science and Technology, 9(2):107–112, 2021.

[78] Ruitao Hou and Hequn Xian. A graded reversible watermarking scheme for relational data. Mobile Networks and Applications, 26(4):1552–1563, 2021.

[79] Chia-Chen Lin, Thai-Son Nguyen, and Chin-Chen Chang. Lwr-crb: Lossless robust watermarking scheme for categorical relational databases. Symmetry, 13(11):2191, 2021.

[80] Xi Shen, Yingzhou Zhang, Tianqi Wang, and Yuxin Sun. Relational database watermarking for data tracing. In 2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), pages 224–231. IEEE, 2020.

[81] Yan Li, Junwei Wang, and Xiuyong Luo. A reversible database watermarking method non-redundancy shifting-based histogram gaps. International Journal of Distributed Sensor Networks, 16(5):15501742921769, 2020.

[82] Junyao Liu. A new reversible database watermarking approach with ant colony optimization algorithm. In Journal of Physics: Conference Series, volume 1616, pages 042004. IOP Publishing, 2020.

[83] Yan Li, Junwei Wang, and Hongyong Jia. A robust and reversible watermarking algorithm for a relational database based on continuous columns in histogram. Mathematics, 8(11):1994, 2020.

[84] Ali Hamadou, Lacine Camara, Abdoul Aziz Issaka Hassane, and Harouna Naroua. Reversible fragile watermarking scheme for relational database based on prediction-error expansion. Mathematical Problems in Engineering, 2020, 2020.

[85] Chonghui Ge, Jian Sun, Yuxin Sun, Yunlong Di, Yongjun Zhu, Linfeng Xie, and Yingzhou Zhang. Reversible database watermarking based on random forest and genetic algorithm. In 2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), pages 239–247. IEEE, 2020.

[86] Hina Tufail, Kashif Zafar, and Abdul Rauf Baig. Relational database security using digital watermarking and evolutionary techniques. Computational Intelligence and Industrial Applications 2020 (CyberC), pages 716–719, 2020.

[87] Heyan Chai, Shuangyang Wang, Lihao Gao, Renfa Li, and Xuan Wang. A robust and reversible watermarking technique for relational dataset based on clustering. In 2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), pages 411–418. IEEE, 2019.

[88] Heyan Chai, Shuangyang Wang, Lihao Gao, Renfa Li, and Xuan Wang, Yiqun Chen, and Hengyu Luo. A new robust and reversible watermarking technique based on erasure code. In International Conference on Algorithms and Architectures for Parallel Processing, pages 153–168. Springer, 2019.
[89] Guan-Yu Wu and Che-Wei Lee. Framework for cloud database protection by using reversible data hiding methods. In 2019 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), pages 1–2. IEEE, 2019.

[90] Yan Li, Junwei Wang, Shuangkui Ge, Xiangyang Luo, and Bo Wang. A reversible database watermarking method with low distortion. Mathematical Biosciences and Engineering, 16(5):4053–4068, 2019.

[91] Donghui Hu, Dan Zhao, and Shuli Zheng. A new robust approach for reversible database watermarking with distortion control. IEEE Transactions on Knowledge and Data Engineering, 31(6):1024–1037, 2018.

[92] Mustafa Bilgehan Imamoglu, Mustafa Ulutas, and Guzin Ulutas. A new reversible database watermarking approach with firefly optimization algorithm. Mathematical Problems in Engineering, 2017, 2017.

[93] Chin-Chen Chang, Thai-Son Nguyen, and Chia-Chen Lin. A virtual primary key for reversible watermarking textual relational databases. In Intelligent Systems and Applications, pages 756–769. IOS Press, 2015.

[94] Chin-Chen Chang, Thai-Son Nguyen, and Chia-Chen Lin. A blind robust reversible watermark scheme for textual relational databases with virtual primary key. In International Workshop on Digital Watermarking, pages 75–89. Springer, 2014.

[95] Saman Iftikhar, M Kamran, and Zahid Anwar. Rrw—a robust and reversible watermarking technique for relational data. IEEE Transactions on knowledge and data engineering, 27(4):1132–1145, 2014.

[96] Chin-Chen Chang, Thai-Son Nguyen, and Chia-Chen Lin. A blind reversible robust watermarking scheme for relational databases. The Scientific World Journal, 2013, 2013.

[97] Khurram Jawad and Asifullah Khan. Genetic algorithm and difference expansion based reversible watermarking for relational databases. Journal of Systems and Software, 86(11):2742–2753, 2013.

[98] Mahmoud E Farfoura, Shi-Jinn Horng, and Xian Wang. A novel blind reversible method for watermarking relational databases. Journal of the Chinese Institute of Engineers, 36(1):87–97, 2013.

[99] Mahmoud E Farfoura, Shi-Jinn Horng, Jui-Lin Lai, Ray-Shine Run, Rong-Jian Chen, and Muhammad Khurram Khan. A blind reversible method for watermarking relational databases based on a time-stamping protocol. Expert Systems with Applications, 39(3):3185–3196, 2012.

[100] J Franco Contreras, Gouenou Coatrieux, Emmanuel Chazard, Frédéric Cuppens, Nora Cuppens-Boulahia, and Christian Roux. Robust lossless watermarking based on circular interpretation of bijective transformations for the protection of medical databases. In 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pages 5875–5878. IEEE, 2012.

[101] Gaurav Gupta and Josef Pieprzyk. Reversible and blind database watermarking using difference expansion. International Journal of Digital Crime and Forensics (IJDCF), 1(2):42–54, 2009.

[102] Gaurav Gupta, Josef Pieprzyk, et al. Reversible and semi-blind relational database watermarking. In SIGMAP, pages 283–290, 2007.

[103] Yong Zhang, Bian Yang, and Xia-Mu Niu. Reversible watermarking for relational database authentication. Journal of Computers, 17(2):59–66, 2006.

[104] Gaurav Gupta and Josef Pieprzyk. Reversible and blind database watermarking using difference expansion. In Proceedings of the 1st international conference on Forensic applications and techniques in telecommunications, information, and multimedia and workshop, pages 1–6, 2008.

[105] K Unnikrishnan and KV Pramod. Robust optimal position detection scheme for relational database watermarking through holpsofa algorithm. Journal of Information Security and Applications, 35:1–12, 2017.

SAPANA RANI received the B.Tech. degree in Information Technology from MIT, Muzaffarpur, India, in 2012. Currently, she is working toward the Ph.D. degree in the Department of Computer Science and Engineering, IIT Patna. Her research areas include Database Watermarking, Information security, and Big Data.

RAJU HALDER is an associate professor in the Department of Computer Science and Engineering, IIT Patna. He received the doctoral degree from Università Ca’ Foscari, Italy, in 2012. Before joining IIT Patna, he served as a post doctoral researcher at Macquarie University, Australia. He worked with the Robotics team at HASLab (University of Minho), Portugal, in 2016. Prior to his Ph.D., he had also worked as an associate system engineer at IBM India Pvt. Ltd. during 2007-2008. His area of research interests include Formal Methods, Blockchain Technology, Program Analysis and Verification, Data Privacy and Security.