Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 697165, 10 pages
http://dx.doi.org/10.1155/2014/697165

Research Article

Distortion-Free Watermarking Approach for Relational Database Integrity Checking

Lancine Camara,[1,2] Junyi Li,[1,2] Renfa Li,[1,2] and Wenyong Xie[1]

[1] *College of Information Science and Engineering, Hunan University, Changsha, Hunan 410082, China*
[2] *Key Laboratory for Embedded and Network Computing of Hunan Province, Changsha, Hunan 410082, China*

Correspondence should be addressed to Junyi Li; junyilee@hnu.edu.cn

Received 21 March 2014; Revised 15 July 2014; Accepted 23 July 2014; Published 14 August 2014

Academic Editor: Kwok-Wo Wong

Nowadays, internet is becoming a suitable way of accessing the databases. Such data are exposed to various types of attack with the aim to confuse the ownership proofing or the content protection. In this paper, we propose a new approach based on fragile zero watermarking for the authentication of numeric relational data. Contrary to some previous databases watermarking techniques which cause some distortions in the original database and may not preserve the data usability constraints, our approach simply seeks to generate the watermark from the original database. First, the adopted method partitions the database relation into independent square matrix groups. Then, group-based watermarks are securely generated and registered in a trusted third party. The integrity verification is performed by computing the determinant and the diagonal's minor for each group. As a result, tampering can be localized up to attribute group level. Theoretical and experimental results demonstrate that the proposed technique is resilient against tuples insertion, tuples deletion, and attributes values modification attacks. Furthermore, comparison with recent related effort shows that our scheme performs better in detecting multifaceted attacks.

1. Introduction

Nowadays, with the wide distribution of digital data, it is necessary to protect them against illicit copying, intellectual property theft and falsification. Digital watermarking is part of the big family of information security that has been proposed to overcome the above issues. Traditionally, its purpose is to protect a digital content by embedding a secret mark (watermark) into the original data. The technique initially designed for image has been extended to other digital data such as video, audio, software, text document, and relational database. This paper focuses on numerical relational database watermarking.

In the literature, most of the research works in relational database watermarking focused on the following three main applications regardless of the database attributes type (numeric, nonnumeric, or mix format): database ownership protection, content authentication, and fingerprinting. Usually, the ownership protection and the fingerprinting techniques [1–4] can be considered as part of robust watermarking approach, whereas the authentication or integrity checking [5–7] is based on the fragile watermarking technique. Whether a watermarking scheme is fragile or robust, it may suffer from intentional or unintentional attacks which may destroy the watermark [8]. Benign update may cause the modification or deletion of marked tuples. Bit and rounding attacks intentionally try to demolish the watermark by changing the bits position from the marked value. In collusion attack, the attacker has access to many fingerprinted copies of the same database and inserts his watermark into the marked database to claim the ownership. The false claim of ownership attack attempts to give to a traitor the evidence that there is a doubt that the data belong to the owner. In the subset reverse order attack, the attacker hopes to destroy the watermark by simply exchanging the tuples or attributes positions in the relation. The brute force attack is made by trying to guess the private data used in the embedding process by traversing the possible spaces of the parameter. Unanimously, a good database watermarking technique should meet the following challenges: (i) imperceptibility: the embedded watermark should be invisible and the watermark insertion process should not

degrade the data usability, (ii) robustness: the watermark should be robust against attack with the aim to destroy the watermark, (iii) security: the watermarking embedding process must use a private key for the security purpose, (iv) blindness: the watermarking detection process should not require the knowledge of original data and the watermark information.

Since the first well known effort which showed the need of watermarking relational database [9], the area gains a lot of interest and many works have been done on it. Guo et al. identified the problem and the importance of fragile watermarking to verify the integrity of numeric streaming data [10]. Their technique can detect and locate any modification made to a data stream. In [11], the authors focused on the connection between relational database and optimization techniques. This method considered the watermarking issue as a constrained optimization problem for the embedding process; the watermark decoding process is based on a threshold based technique to minimize the probability of decoding errors. The authors in [12] proposed the reversible watermarking algorithm which enables recovering the original data back from tampered data. This technique is primary key dependent and is not flexible to linear transformation attack since the watermark embedded into some selected tuples cannot detect if marked tuples are deleted. The work in [13] which seeks to protect the database integrity proved that the database tables' indexes would improve the detection of unauthorized alterations. In this regard, the author used R-tree scheme data structure which does not change the value of the attribute. The proposed work in [14] argues about the rewatermarking attack. The approach integrated the watermark information from the date stamp to overcome the problem. Furthermore, the bi-folder security scheme is used to detect and resolve conflicting ownership issues in case of rewatermarking attacks.

Our approach is a distortion-free fragile watermarking technique; it does not change any data value from the database. In the literature there is a huge research work on relational database watermarking but only few of them are focused on fragile watermarking technique [7, 15].

In this paper, the following contributions have been accomplished. (i) We identified that a numerical database can be securely partitioned into a set of square matrices (groups). As a result, some matrix-based properties like determinant could be applied to check the integrity of the database. (ii) We designed an algorithm that can detect and localize the tampered region in the database at group level. (iii) We conducted experiment to show the proposed technique feasibility and usefulness and also compared our technique with previous work.

This paper is organized as follows. The next section discusses the related work. In Section 3, the basic terminology and concept used in this paper are explained. Section 4 describes the details of our proposed scheme; the performance of our approach is also discussed. The results of our experiments are discussed in Section 5. Finally, Section 6 concludes our paper and provides some guidance for future work.

2. Related Work

Agrawal et al. [9] identified the need for watermarking techniques in relational database; they embedded the watermark in the least significant bits (LSB) of selected attributes of some selected tuples. A secure message authentication code (MAC) is computed using secret key and primary key for each tuple to select the candidate tuples, attributes, and LSB position for watermark embedding. Inserting watermark bits in LSB is efficient, but the watermark can be easily compromised by bits attacks. Guo et al. [15] proposed a fragile watermarking scheme to detect, localize, and characterize the malicious modifications of relational database. The technique used numeric data and divided all tuples into groups according to their primary key hash values. Furthermore, their technique modify two bits in the data LSB values and may not preserve the data usability constraints. In [5], another work used fragile watermark technique to detect and localize malicious alterations made to the relational database with categorical attribute. It does not introduce any distortion to the cover data; it is a distortion-free approach. Database tuples are securely parsed into groups; watermarks are then embedded and verified at group level independently. In [16], the authors presented a zero distortion watermarking technique to verify that the integrity of relational databases is introduced. The partitioning technique used is a virtual group operation which generates image of the partition as a watermark of that partition.

Recently, Khan and Husain [17] proposed a fragile scheme based on zero watermarking technique to protect the integrity of database relations. Their technique is algorithmically based on evaluating the local characteristics of database relation like data values frequency distribution of digit, length, and range. Moreover, their technique can characterize the malicious data in the data set using data parameters like the fraction of digit, length, and range to quantify the nature of tampering attack. Their technique is not resilient to attribute values substitution attack. In this paper, we present a distortion-free watermarking technique which partitions a database into groups of square matrices and then generates the watermark.

3. Basic Concept and Terminology

In this section, we present some notations, parameters, and mathematical approaches used in our proposed algorithm.

We consider D being a relational database having γ attributes, α tuples, and the following scheme $(P, A_0, \ldots, A_{\gamma-1})$, where P is the primary key and $A_0, \ldots, A_{\gamma-1}$ are the γ attributes of the database D. The primary key P may be one or a combination of some attributes of $(A_i)_{0\le i\le \gamma-1} \in R$. Notations Section summarizes some parameters used in our algorithms.

A matrix is a rectangular array of numbers. A matrix with m rows and n columns is called an $m \times n$ matrix [18]. The matrix may contain complex numbers and each number that composes the matrix is called an element of the matrix. Two matrices are equal if they have the same number of rows and the same number of columns and above that the corresponding entries in every position are equal. A matrix

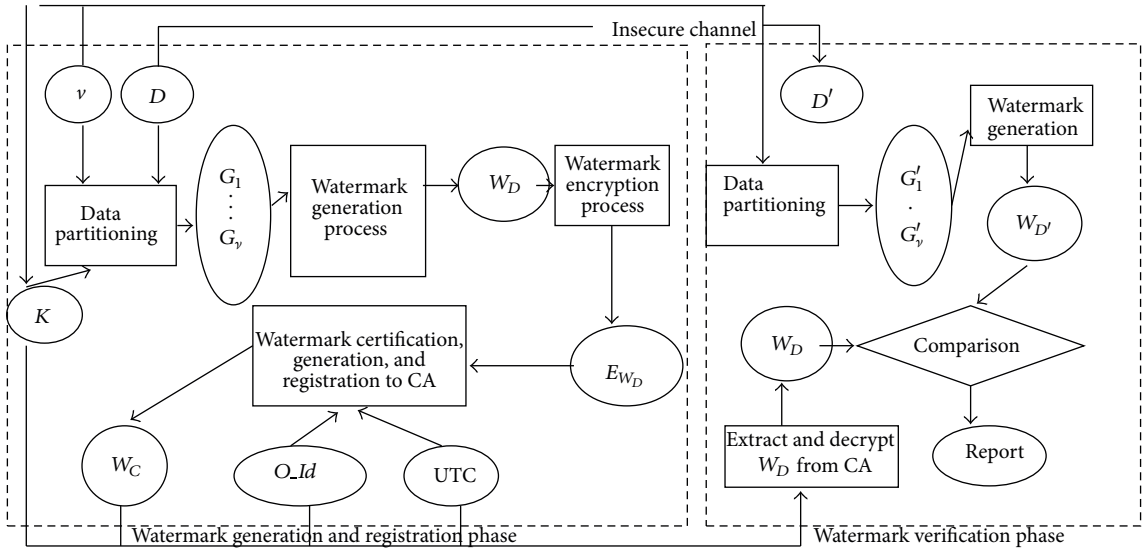

Figure 1: Architecture of watermark computation, certification, and verification process.

A is called square matrix if its numbers of rows and columns are the same and it is represented by $A = \begin{pmatrix} a_{11} \cdots a_{1n} \\ \vdots \quad \vdots \\ a_{n1} \cdots a_{nn} \end{pmatrix}$. The determinant of a square matrix "A" denoted by "$\det A$" or simply $|A|$ is the real number that can be computed from the square matrix A; formula (1) shows its computation formula. The minor of any element a_{ij} of a square matrix "A" denoted by M_{ij} is the determinant obtained when the row and column of that element are deleted. Let "A" be a square matrix of order n with coefficients in $\mathbb{R}$; the determinant of A is computed as follows:

$$|A| = \sum_{i=1}^{n} (-1)^{i+1} a_{i1} \det\left(A_i\right), \tag{1}$$

where A_i is obtained by deleting the first column and ith row.

The ceiling function assigns to the real number x the smallest integer that is greater than or equal to x. The value of the ceiling function at x is denoted by $\lceil x \rceil$.

4. Proposed Approach

The different phases of the proposed approach are detailed in this section. Our technique is distortion-free watermarking based on using numeric data; it does not embed any watermark in the original database. Figure 1 shows the architecture of the proposed watermarking approach; there are two main phases: the watermark generation and certification phase and the watermark verification phase.

The watermark generation and certification phase focuses on the characteristics of the content of the subsets of numeric database values which are summarized as follows.

Step 1 (data set partitioning). The secret key K and the number of group ν are used to partition the data set D to obtain ν different square matrix groups or partitions $\{G_1, \ldots, G_\nu\}$.

Step 2 (watermark generation). Individual watermark is first computed for each group and the computed watermarks are then concatenated to obtain the data set watermark D_W.

Step 3 (watermark encryption). The data set watermark D_W is encrypted using a secure hash function to obtain the encrypted data set watermark E_{D_W}.

Step 4 (watermark certification and registration). The watermark certificate W_C is obtained by concatenating the encrypted data set watermark E_{D_W}, the data set owner ID (O_Id), and the coordinated universal time date time stamp (UTC). The watermark certificate is finally registered at a Certification Authority (CA) for certification purpose.

Our watermarking approach does not modify the data set but simply computes some information from the data set. The data set D is delivered to the intended recipient, and the data set D may be intercepted by an attacker through the insecure channel or attack channel and may be subject to intentional or unintentional attack which necessarily modifies the data set D.

The watermarking verification phase is the process of comparing the data set watermark W_D registered at the certification authority and the data set watermark $W_{D'}$ from suspicious data set.

The watermark verification is depicted as follows.

Step 1 (data set partitioning). The secret key K and the number of group ν are used to partition the data set D' from the insecure channel which allows third party to have access to the data set. The data set D' is partitioned into ν different square matrix groups or partitions $\{G'_1, \ldots, G'_\nu\}$.

Step 2 (watermark generation). Individual watermark is first computed for each group and the computed watermarks are concatenated to obtains the data set watermark $W_{D'}$.

Input: Data set relation D, Number of groups $\nu = \left\lceil \frac{\alpha}{\gamma} \right\rceil$, and Secret key K
Output: ν groups $(G_1, G_2, \ldots, G_{\nu-1}, G_\nu)$ of length γ each
(1) **Begin**
(2) for $i = 1$ to α **do**
(3) $h_i^r = Hash(k \,||r_i \cdot P|| \, k)$ // ith row primary key hash
(4) $j = h_i \mod \nu$ // group index
(5) Insert r_i into G_j
(6) **end for**
(7) return $(G_1, G_2, \ldots, G_{\nu-1}, G_\nu)$
(8) **end.**

Algorithm 1: Database partitioning.

Step 3 (watermark extraction from the CA). The data set watermark W_D is extracted from the watermark certificate registered at the certification authority.

Step 4 (watermark comparison). The data set watermark $W_{D'}$ and the data set watermark W_D are compared for the integrity verification of the target data set D.

4.1. Data Partitioning. The proposed data partitioning technique is described in Algorithm 1. The data set D that is composed of α tuples and γ attributes is partitioned into ν different groups as follows: $(G_1, G_2, \ldots, G_{\nu-1}, G_\nu)$, where $G_i \neq G_j$ for $i \neq j$. Our parsing process exploited the idea of the partitioning technique used in [11]. The difference between the two algorithms resides in the number of desired partitions. The number of partitions in [11] varies depending on the number of partitions decided by the data owner but for our approach the number of partitions is $\nu = \lceil \alpha/\gamma \rceil$ such as α the number of data set tuples, γ the number of data set attributes, and $\lceil \ \rceil$ the mathematical ceiling function. The secure message authentication code (MAC) is computed for each tuple α that belongs to D and few tuples α_i with the size that is equal to γ are inserted logically into each partition $G_{j(1 \leq j \leq \gamma)}$. The tuples partitioning assignment to the group is given as follows: $j = \text{Hash}(k \| \alpha_i \cdot P \| k) \bmod \nu$, where $\alpha_i \cdot P$ is the primary key of the tuple α_i, K is the secret key, $\|$ is the concatenation operator, and Hash() is the secure hash function. Furthermore, if the database does not have any primary key, the primary key can be a combination of few attributes from the relational database.

As a result, the data set D is partitioned into ν different groups $\{G_1, \ldots, G_\nu\}$.

If $\alpha \bmod \gamma \neq 0$, we simply securely insert records to complete the last group in order to form a square matrix and we assume that there is no identical tuple in the same group. Note that the added records will be deleted after the watermark computation.

To the best of our knowledge, no partitioning technique parses the database relation into square matrix-based groups. The group number is kept secret in order to reinforce the security of our scheme; however, even if an attacker knows the number of group used in watermark generation process, he cannot generate the groups because you need to know the secret key which is kept secret to do that.

The major advantage of such partitioning method is that many properties of square matrix can be applied to check the group's data integrity. By knowing the number of attributes and tuples of a data set, the number of group can be easily computed so as each group attribute cardinal equals its tuple cardinal.

4.2. Watermark Group Generation. In this step, the group watermark generation approach (Algorithm 2) is described. According to each group primary key hash value, the γ tuples are first sorted in a certain order (increasing order in our algorithm); this process will not physically affect the group data values. In lines 5-6, the determinant value of the group and the different values of the minor of diagonal values of the group are computed and finally the different computed values are concatenated to get the watermark value of the group. The watermark values of each group are used in the watermark certificate computation.

4.3. Watermark Computation and Registration. The watermark generation method is presented in Algorithm 3. In lines 1-2, the data set is partitioned into groups in such a way that each group has equal number of attributes and tuples. In line 3, the watermark is computed from each group as follows. First, the tuples are sorted in ascending order by considering their primary key hash values. Note that this operation is secure virtual operation; it is governed by the use of a secret key and it does affect the tuples physical position or the primary key value. Then, to compute the watermark of each group, we consider the group as square matrix and we compute and concatenate its determinant and the minors of each diagonal values as described in Section 4.2. It is important to note that for a matrix it is difficult to find two matrices having the same determinant and diagonal's minors. In lines 3-4, the database watermark group is computed by the concatenation of different group's watermarks. The secure hash function and the secret key are used in line 5 to encrypt the data set watermark. Finally, the encrypted watermark is concatenated with additional information (owner ID and coordinated universal time (UTC) time stamp) and then

Input: Group G_j
Output: Group Watermark W_j
(1) **Begin**
(2) **for** $j = 1$ **to** ν **do**
(3) Sort all tuples in G_j according to the increasing order of their primary key hash
(4) compute the determinant D_j of G_j // jth determinant group
(5) compute the minor of $M^j_{i(0\le i\le \gamma)}$ of $(A_{i,i})_{1\le i\le \gamma}$ // minor of jth group diagonal
(6) compute $W_j = D_j \,||\, M^j_{i(0\le i\le \gamma)}$ // jth watermark group
(7) **end for**
(8) return W_j
(9) end.

Algorithm 2: Group watermark generation.

Input: Database D, Secret key K, and number of groups $\nu = \left\lceil \frac{\alpha}{\gamma} \right\rceil$
Output: Watermark certificate
(1) **Begin**
(2) Database partitioning into groups // see Algorithm 1
(3) Group watermarks computation // see Algorithm 2
(4) $W_R = W_1 \,||\cdots||\, W_\nu$ //watermarks groups concatenation
(5) $E_{W_R} = Encrypt(W_R \,||\, K)$ // Encrypt watermark
(6) $W_C = E_{W_R} \,||\, owner\_id|date$ // watermark certificate
(7) Return W_C
(8) End

Algorithm 3: Watermark computation.

registered with the certification authority (CA) which is a trusted party. The purpose of CA is to identify a decision authority for the suspicious data set authentication. The owner ID and the UTC may give more reliability to the certification authority on the owner of the database.

4.4. Integrity Checking. The integrity of a suspicious database is verified in Algorithm 4. In lines 2-3, the watermark certificate registered at the CA is recuperated and the original relational watermark W_D is extracted using the secret key K used in watermark generation.

The suspicious data set D' is partitioned using the technique used in the watermark generation; afterwards, the watermark W'_D is computed in line 6; note that the watermark generation technique is the same as the one used in Algorithm 2. In lines 7–10, the two generated watermarks are compared, and if they are different, the data set has been modified and if there are the same this means that the original data set is not tampered. To verify the integrity or the tamper proofing of a suspicious data set, the CA can solve the above problem by partitioning the suspicious data set D' into different groups, and each group watermark is computed and compared with its corresponding watermark extracted from the original watermark group registered. If the two watermarks differ from each other, this means that the original data set D has been modified at the group index level. In the same manner, the data set is modified. Furthermore, the CA can solve multiple watermark conflicts by comparing the UTC date and time issue from the data set owner to the attacker. It is important to mention the blindness of our technique because it does not require the original database or the original watermark information in the watermark detection.

4.5. Security Analysis. In this section, we provide the security analysis of our scheme using the theory of probability. Since the proposed technique is fragile in nature, the suspicious database can be subject to many attacks with the aim to maliciously modify the protected data while not touching the certified watermark. We analyze the success of the use of probability to change the database while keeping the watermark intact. We suppose that an attacker has access to the data partitions. Our watermark computation is based on the calculation of the determinant of group matrix. By substituting twice two columns of a matrix, the value of its determinant remains the same. Let D be a data set having γ attributes and α tuples; an attacker may generate the partitions of the database (recall that partitions are considered as square matrices) and succeed to substitute columns to get the value of determinant with the probability $P_{\text{succes}}(S)$ but cannot generate the expected values of minor for the same matrix. Since our technique is based on fragile database watermarking, any change made to the marked data

Input: Suspicious database D', Secret key K,
number of groups $\nu = \left\lceil \frac{\alpha}{\gamma} \right\rceil$ and W_C
Output: Verification report
(1) Begin
(2) extract E_{W_D} from W_C
(3) compute $W_D = Decrypt(E_{W_D} \,||\, K)$
(4) database partitioning into groups // see Algorithm 1
(5) for $j = 1$ to ν' do
(6) compute $W'_j = D'_j \,||\, M'^{j}_{i(0\le i\le \gamma)}$ // jth suspicious group watermark
(7) if $W_j = W'_j$ then // watermarks comparison
(8) return non-altered group
(9) else
(10) return altered group
(11) end if
(12) end for
(13) end

Algorithm 4: Integrity checking.

should affect highly the original watermark. In this study, we deal with the case of interchanging data values.

We treat the above problem as a distinct permutation of n objects taken r at a time and represented by the symbol $n\mathrm{Pr}$ and $n\mathrm{Pr} = P(n, r) = n!(n - r)!$. We analyze the case of large data set that can be partitioned to a number of square matrices. Suppose that each matrix has at least 3 attributes ($\gamma \geq 3$). The probability to successfully compute the determinant of a single group is

$$P_{\text{success}}(S) = \frac{2\gamma}{(\gamma^2)!} + \frac{2\gamma}{(\gamma^2)!} = \frac{4 * \gamma}{(\gamma^2)!}. \tag{2}$$

Recall that α is the number of database tuple and γ the number of database attribute. The probability of successfully modifying the data set group by preserving it determinant value is

$$P_{\text{success}}(S) = \left(\frac{4 * \gamma}{(\gamma^2)!}\right)^{\alpha/\gamma}. \tag{3}$$

It is clear for large data set that we have $4 * \gamma/(\gamma^2)! \prec 0$ and $\alpha/\gamma \succ 1$. So, for such databases the probability of successfully modifying the database by preserving each group determinant value is:

$$P_{\text{success}}(S) = \left(\frac{4 * \gamma}{(\gamma^2)!}\right)^{(\alpha/\gamma)^n}_{\gamma\geq 3}. \tag{4}$$

It is clear that this probability is very small and approaches zero. Consider

$$P_{\text{success}}(S) = \lim_{n\to\infty}\left(\frac{4 * \gamma}{(\gamma^2)!}\right)^{(\alpha/\gamma)^n} \longrightarrow 0. \tag{5}$$

5. Experimental Results

We performed experiments to show the effectiveness and the accuracy of the proposed technique. We made sure that the subject data set contains some numeric attributes. We evaluated our approach on a Forest Cover Type [19], a real life data set containing 581,012 tuples and each tuple contains 10 integer attributes, 1 categorical attribute, and 44 Boolean attributes. We added a primary key attribute to a Forest Cover Type for experiment purpose. We used Microsoft SQL Server 2008 running on 2.2 GHz Intel Dual CPU with 2 GB of RAM computer.

We first generated the original data set watermark W_D; afterwards, we simulated different attacks against it with the aim to generate the same computed original watermark. In the detection process, the same secret parameters used in watermark generation are used. The resilience of our approach and Khan and Husain approach [17] are verified by performing different attacks on the cover Type database. The accuracy of our approach after an attack on the database is verified by computing and comparing the original watermark W_D and generated watermark $W_{D'}$ obtained after a modification of data set D.

We also measured the cost of our algorithm, the watermark computation, and registration program and the watermark verification program was run twice. The average times needed for the experiments are, respectively, 5281 seconds and 4952 seconds. The data set integrity verification is costly but can be performed once a day mostly depending on the need.

We randomly tested two kinds of attacks against the watermarked relation: (i) common attacks: tuples insertion, tuples deletion, and attributes values alteration and (ii) multifaceted (substitution) attacks in which the pirate interchanges attributes values and/or replaces some tuples by others while preserving the original size of the database relation [20]. A multifaceted attack is performed by an experienced

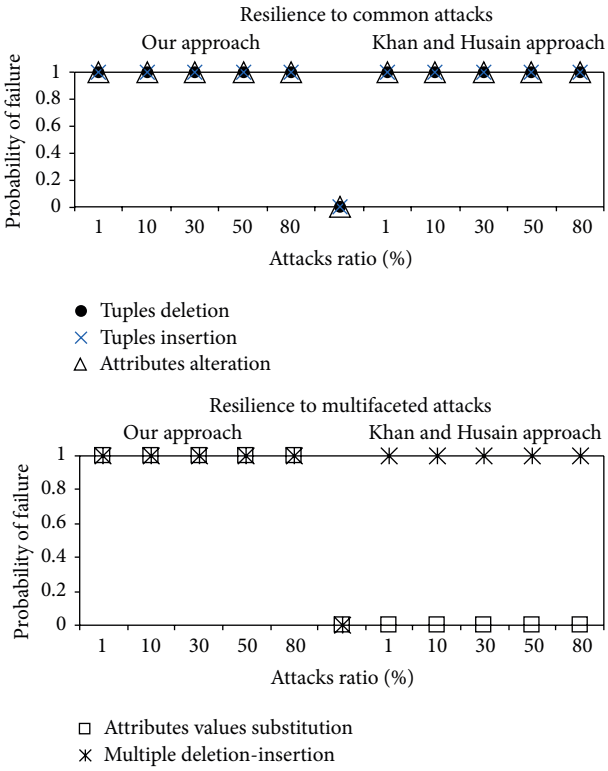

Figure 2: Resilience to malicious attacks.

attacker who may be aware of the whole process but does not know the secret key. Figure 2 shows the results of the different performed experiments. We have also compared our approach with Khan and Husain technique [17]. These results (Table 2) clearly demonstrate that Khan's technique is not resilient to attributes values interchanging attack, whereas with our scheme even a minor data change has significant impact on the extracted watermark. By knowing the number of groups used in the watermark computation process, after massive deletion or insertion attacks, we may have one or more incomplete groups than the expected one. For all types of attacks in fragile watermarking, the aim of an attacker is to alter the original database by keeping the watermark intact. The experiment results of all attacks are showed in Figure 2.

5.1. Common Attacks. Such attacks are traditionally performed to verify the integrity of a relational database watermarking technique.

(i) *Insertion Attacks*. Both approaches have been tested against insertion attacks. We inserted randomly and progressively new tuples in the data set D. Both techniques are resilient to insertion attack, and the attack is easily detected. In our approach if the insertion reaches a certain threshold, by comparing the number of groups of the suspicious database and the group ν, the tampering can be easily detected. Using the number of group should not dispute (doubt) our technique blindness because the number of group is not the original database, it is used in watermark integrity checking process as watermark function input.

(ii) *Deletion Attacks*. Tuples are deleted progressively from original database and the watermark verification algorithms of both approaches compute the different watermarks from the deleted database. Experiments show that the two techniques are resilient to deleting attack even if the rate of deleted tuple is small. From our algorithm partitioning process we can easily realize that the suspicious database group is smaller than the expected one ν; then, the tampering is proved.

(iii) *Alteration Attacks*. We assume that he attacker does not have access to the original data set D or the secret key K. We randomly modify progressively the original database data values. We can see that any change in the data set is detected after comparing the original watermark and the one from the modified data set. Experimental results reported in Figure 2 showed that the two approaches are resilient against the alteration attack; the tampering is proved even if one data value of the database is modified.

5.2. Multifaceted Attacks. In such attacks, we alter the database as a sophisticated attacker.

(i) *Attribute Values Substitution Attacks*. In this kind of attacks, we change the position of some attribute values of the database. Our approach is resilient to data value interchanging attack and can prove the nonintegrity of the database even if two data values have been exchanged. In contrary, khan's approach is not resilient to such attack, after computing the suspicious database watermark; we may erroneously conclude the integrity of the database.

(ii) *Tuples Insertion-Deletion Attacks*. To deal with this kind of attack, we progressively delete and add the same number of tuples (if β tuples are deleted, β tuples are inserted). The reason is that our technique can easily detect any diminution or augmentation of tuples in the data set; it cannot generate the expected number of groups. Experimental results in Figure 2 showed that the two approaches are resilient against the combined insertion and deletion attack.

6. Discussions

The experiments showed that our technique is feasible and resilient to malicious attacks. The massive insertion and deletion attacks may be easily detected up to the group partitioning process. The reason is that the number of group is known by the owner and the partitioning process may fail by giving an unexpected group after a high number of tuples insertion or deletion.

By changing around the group data values, it may be possible to get the group determinant value, however, getting the diagonal minor of the same group is more complex, may be impossible. Accordingly, it gives more reliability to

Input: Group Watermark W_j, and W_j'
Output: tamper area
(1) **for** each G_γ^j
(2) **if** $W_j \neq W_j'$
(3) **for** $i = 1$ to **i** $= \gamma$
(4) **if** $M^j_{i(0\leq i\leq \gamma)} = M^{j\prime}_{i(1\leq i\leq \gamma)}$
(5) **return** "the tampered data in ith column or ith attribute"
(6) **end if**
(7) **continue**
(8) **end if**
(9) **end for**
(10) **end for**
(11) **end**

Algorithm 5: Localization of the altered data.

our technique since it is based on determinant and minors computing at group level.

The success probability that an attacker can modify a group and retrieve the determinant and minor values computed from watermark generation phase approaches zero. After a wide experimentation and researches, we have not found one possible case. Suppose that an attacker can generate the data set groups, he (she) can modify the data set group and obtain the same determinant values computed from watermark generation phase by substituting an even number of group attribute. After such attack it is clear from Figure 2 that the probability that the attacks fail approaches 1. All the experimental results are reported in Figure 2. We use minor because the data set can be modified, while the value of the determinant remains the same; adding minor gives more strength to our technique, and it will be difficult or even impossible to compute the same minors values. Moreover, it can help to localize the tampered area. Algorithm 5 explains how a tampered area can be detected from a specific group after slight alteration.

The experimental results indicate that our algorithm performs well enough to be used in real world database applications. We are planning to reduce the cost of our approach by computing and selecting some individual square matrix for watermark certificate computation and verification.

In our approach partitioning process, supposing the data set had α tuples, γ attributes, and $\lceil \alpha/\gamma \rceil \neq 0$, we securely complete the last group to square matrix. For example, if $\alpha = 14$ and $\gamma = 3$ we can generate 5 groups: 4 groups with 3 attributes and 3 columns considered as 4 square matrices, but the 5th group is incomplete and one more tuple should be added to it to complete it to 3 tuples. To deal with such case, we complete the 5th group to 3 rows by adding the first tuple of the first group. The added tuple must be deleted after watermark generation.

The observation from Figure 2 shows our approach resilience against various attacks and when an attack occurs, the probability that it fails is 1. Furthermore, a sophisticated attacker may not be able to succeed any alteration to our database without affecting the watermark.

From the comparison between our approach and Khan's approach, one important difference is the resilience of our approach against database attribute values substitution attacks. For example, let us consider a simple example given by the following student records as shown in Table 1.

Table 1: Students records.

	A			A'	
ID	Age	Score	ID	Age	Score
ML001	22	90	ML001	22	85
ML008	19	85	ML008	19	90

The database has been altered; the two score values of the record A have been substituted and accordingly new altered record A' is created. By computing the determinant and the diagonal minor values of the two database values, we can detect the nonintegrity of the database. But, by computing the length, the range, and the digit of the database attributes values, we may erroneously conclude that the integrity of the database remains intact. Our technique is resilient to such kind of attack, but khan et al.'s approach is not resilient to data values substituting attacks. Another difference is that our approach is best suited for a database having at least two numeric data attributes and khan's approach can be used even with a single numeric attribute database.

7. Conclusion and Future Work

In this paper, a novel fragile watermarking technique for database integrity verification is presented. The proposed technique is based on distortion-free watermarking concepts and does not modify the original data. Our technique partitions the data set into different sets of square matrices and generates the relational watermark using the determinant and the minor of the generated square matrix. We have compared our technique with previous technique and experiments to show the simplicity and usefulness of the proposed technique to verify the integrity of database after data values interchanging attack and other malicious attacks. Our approach overcomes some limitations in existing fragile watermarking techniques like preserving the data usability constraints and the integrity verification when various attacks occur, and it may retrieve the tamper attribute up to group

Table 2: Comparison with Khan's approach.

		Our approach	Khan and Husain approach
Insertion, deletion, alteration attacks		Detect and localize the tampering at group level and may detect the tampered tuples	Detect and characterize malicious modification made to the database and characterize the tamper attack
Database partitioning		Partition based technique	No partitioning
Multifaceted attack	Tuples insertion-deletion attacks	Resilient to such kind of attack	Resilient to such kind of attack
	Attributes substitution attacks	Resilient to such kind of attack	Not resilient to such kind of attack
Security		Secret key based technique	Secret key based technique

level. The proposed approach does not tolerate any alteration in the numeric database; it therefore detects malicious attacks with high probability. In the future work, we are planning to insert our own watermark into the original database by controlling the usability constraints and extend our approach to nonnumeric data.

Notations

α: Number of tuples in the database
γ: Number of attributes in the relation
$M^j_{i(0\le i\le\gamma)}$: Minor of jth group diagonal data values
G^j_γ: jth group
r_iP: ith primary key row
D_j: jth group determinant
W_j: jth watermarked group
W_C: Watermark certificate
D_W: Data set watermark
E_{D_W}: Encrypted data set watermark
K: Secret key.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This project is sponsored by "the Scientific Research Foundation for the Overseas Chinese Scholars."

References

[1] R. Sion, M. Atallah, and S. Prabhakar, "Rights protection for categorical data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 7, pp. 912–926, 2005.

[2] U. P. Rao, D. R. Patel, and P. M. Vikani, "Relational database watermarking for ownership protection," in *Proceedings of the 2nd International Conference on Communication, Computing & Security (ICCCS '12)*, vol. 6, 2012, Procedia Technology, vol. 6, pp. 988–995, 2012.

[3] Y. Li, V. Swarup, and S. Jajodia, "Fingerprinting relational databases: schemes and specialties," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 1, pp. 34–35, 2005.

[4] X. Xiangrong, S. Xingming, and C. Minggang, "Second-LSB-dependent robust watermarking for relational database," in *Proceedings of the 3rd Internationl Symposium on Information Assurance and Security (IAS '07)*, pp. 292–297, Manchester, UK, August 2007.

[5] Y. Li, H. Guo, and S. Jajodia, "Tamper detection and localization for categorical data using fragile watermarks," in *Proceedings of the 4th ACM Workshop on Digital Rights Management (DRM '04)*, pp. 73–82, October 2004.

[6] H. Xiang, X. Sun, and C. Tang, "New fragile watermarking scheme for text documents authentication," *Wuhan University Journal of Natural Sciences*, vol. 11, no. 6, pp. 1661–1666, 2006.

[7] S. Bhattacharya and A. Cortesi, "Distortion-free authentication watermarking," in *Software and Data Technologies*, vol. 170 of *Communications in Computer and Information Science*, pp. 205–219, Springer, Berlin, Germany, 2013.

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

[9] R. Agrawal, P. J. Haas, and J. Kiernan, "Watermarking relational data: framework, algorithms and analysis," *The VLDB Journal*, vol. 12, no. 2, pp. 157–169, 2003.

[10] H. Guo, Y. Li, and S. Jajodia, "Chaining watermarks for detecting malicious modifications to streaming data," *Information Sciences*, vol. 177, no. 1, pp. 281–298, 2007.

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

[12] M. E. Farfoura, S. J. Horng, J. L. Lai, R. S. Run, R. J. Chen, and M. K. Khan, "A blind reversible method for watermarking relational databases based on a time-stamping protocol," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3185–3196, 2012.

[13] I. Kamel, "A schema for protecting the integrity of databases," *Computers and Security*, vol. 28, no. 7, pp. 698–709, 2009.

[14] S. Suhail, M. Kamran, and F. Arif, "Watermarking of relational databases with emphasis on re-watermarking attack," *International Journal of Computer Science*, vol. 9, no. 1, 2012.

[15] H. Guo, Y. Li, A. Liu, and S. Jajodia, "A fragile watermarking scheme for detecting malicious modifications of database relations," *Information Sciences*, vol. 176, no. 10, pp. 1350–1378, 2006.

[16] S. Bhattacharya and A. Cortesi, "A distortion free watermark framework for relational databases," in *Proceedings of the 4th International Conference on Software and Data Technologies*

(ICSOFT '09), pp. 229–234, INSTICC Press, Sofia, Bulgaria, July 2009.

[17] A. Khan and S. A. Husain, "A fragile zero watermarking scheme to detect and characterize malicious modifications in database relations," *The Scientific World Journal*, vol. 2013, Article ID 796726, 16 pages, 2013.

[18] K. H. Rosen, *Discrete Mathematics and Its Applications*, McGraw-Hill Education, Singapore, 6th edition, 2007.

[19] http://archive.ics.uci.edu/ml/datasets/covertype.

[20] M. Kamran, S. Suhail, and M. Farooq, "A robust, distortion minimizing technique for watermarking relational databases using once-for-all usability constraints," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 12, pp. 2694–2706, 2013.