Distortion Free Database Watermarking System Based on Intelligent Mechanism for Content Integrity and Ownership Control

Saad M. Darwish[1*], Hosam A. Selim[1], Mohamed M. Elsherbiny[2]

[1] Department of Information Technology, Institute of Graduate Studies and Research, University of Alexandria, 163 Horreya Avenue. El Shatby 21526. P.O. Box 832. Alexandria. Egypt.
[2] Department of Material Science, Institute of Graduate Studies and Research, University of Alexandria, Egypt.

* Corresponding author. Tel.: +201222632369; email: saad.darwish@alex-igsr.edu.eg
Manuscript submitted May 10, 2018; accepted July 10, 2018.
doi: 10.17706/jcp.13.9.1053-1066

Abstract: With the development of the Internet and wide applications of databases, the database providers are worried about their ownership of the databases. Database watermarking is an efficient method for database security, without affecting the usability of the data. Initially, most of the research effort in this field was depending on distortion based watermark, by introducing small changes in the underlying data of the database during embedding phase, while the few remaining studies concentrated on distortion-free. But there are many disadvantages in previous studies in distortion-free watermarking area; most notably some rely on adding watermark as an extra attributes or tuples which increase the size of the database. Hence, the ability to destroy the watermark by the attacker is enlarged because the watermark was added inside the database itself. Other techniques such as permutation and abstract interpretation framework require much effort to verify the watermark. The idea of this research is to apply the distortion free one hundred percent by adding the watermark as fake tuples in a separate file not within the database. The proposed system utilizes the GA, which boils down its role to create the values of the fake tuples which formed the watermark to be the closest to real values. So that it's very hard to any attacker to guess the watermark. The proposed watermark achieves more imperceptibility and security. Experimental results verify that the proposed algorithm is feasible, effective and robust against all kinds of attacks.

Key words: Databases copyright protection, genetic algorithm, database watermarking, content integrity.

1. Introduction

While demand for the use of databases is growing, the piracy issue of relational data has become a serious problem. A number of technologies have been developed to provide data protection, including cryptography and steganography [1]. Cryptography protects data by encrypting them. However, once the encrypted data are decrypted, the data are in the clear and are no longer under protection. On the other hand, steganography conceals the very existence of the data by hiding them in cover data. Digital watermarking, a new emerging technology, complements cryptography and steganography such that watermark should not affect the usefulness of data. Watermarking approaches do not prevent copying rather it deters illegal copying by providing a means of establishing the original owners a redistributed copy. There is a wide range of applications of digital watermarking including the verification of integrity, tamper detection, copyright protection.

Most watermarking research concentrated on watermarking multimedia. However, watermarking

databases have unique requirements that differ from those required for watermarking digital audio or video products [2]-[5]. These requirements include: (1) Few Redundant Data: in multimedia data, there is a large amount of space available to hide the watermark, while the database consist of the tuples, each tuple represents a separate object so, the watermark is spread over these separate objects, (2) Out-of-Order Relational Data: The relative spatial/temporal positions of different parts of multimedia objects do not change, whereas there is no ordering among the tuples in database relations as the collection of tuples is considered as a set, and (3) Frequent updating: Multimedia objects typically remain intact; any portion of such an object is not dropped or replaced arbitrarily without causing perceptual changes in the object, whereas tuples may be inserted, deleted, or updated during normal database operations. Due to these differences, we cannot directly use any of the technique as it is for the database, which developed for multimedia data.

The most significant properties to evaluate any watermark system are robustness, blindness, imperceptibility, detectability, and security [6]-[8]. The embedded watermark should be robust against various types of intentional (malicious) and unintentional (benign update) attacks which may damage or erase the watermark. Another important property is blindness, which means that no need for the original tables to detect the watermark. Imperceptibility confirms that the embedding process should not affect the usability of the database and the position of the watermark is not able to be detected by an attacker. The fourth important property is detectability which means that watermark should establish the owner's identity during the detection/extraction phase otherwise the watermark is useless. The fifth important property is security which means that the hacker can't remove the watermark without having full knowledge of embedding algorithm.

From the other viewpoint, most of the previous studies rely on adding the watermark with some distortion to the database and little of them assign the watermark with distortion free. It was observed that most previous studies that used the distortion- free' characteristics to add a watermark are based on (1) Abstract interpretation framework as a theory of the approximation of the semantics in such a way that semantic meaning of data is preserved [9]-[11]; (2) Permutation –based framework that performs the exchange of tuples' positions based on linear permutation un-ranking algorithm to increase the embedding capacity [1], [12], [13]. (3) Fake (virtual) attribute adding framework that inserts a new column that is not real to the relation as a watermark; or by adding only one hidden tuple with a secret function where its values are known only by the data owner [14]-[19]. This type of watermarking has the ability to know the latest updates made by users but it needs a technique to handle primary key collision. (4) Watermark technique through using trusted third party or certificate authority (CA) in which the group-based watermarks are securely generated and registered in a trusted third party. Finally, (5) Zero-watermark technique in which the watermarks are generated by using the local characteristics of database relation itself, like frequency distribution of various digits, lengths, and ranges of data values [20]-[23].

In general, these techniques are more powerful against little types of attacks or modifications such as deleting or updating cell values. Still, most of these database distortion free watermarking schemes are usually developed to protect specific types of data such as numeric or categorical attributes. The majority of these watermarking techniques are fragile and used for maintaining database integrity while the little dealt with copyright protection. The current permutation approaches suffer from the reordering attack and need strong mapping, whereas the recent fake tuple (either hidden or appended) approaches need a secret function based on table attributes. Different from the above-mentioned approaches, the suggested model is truly distortion free watermark approach without using the third party that requires additional cost. In this approach, the watermark is extracted and preserved in a separate file so that it can handle both of attacks against watermark itself and attacks over data itself. Putting the watermark in a separate file prevents the

attackers from destroying it and proves the authentication where the attacker tries to add its own watermark.

The paper is organized as follows: Section 2 discusses the related work that includes the available distortion free watermarking techniques and highlights the shortcomings of these techniques. An overview of our watermarking technique is described in Section 3, where the watermark embedding and extracting phases on relational data based on genetic algorithm are discussed in details, and also presents the theoretical analysis. The experimental results are presented in Section 4. Finally, conclusions are given in Section 5.

2. Literature Review

In recent years, distortion-free database watermarking has been received much attention from researchers. Some pioneer works have been done by incorporating abstract interpretation framework to characterize the approximation of the semantics inside the table [9]-[11], [20]. In this case, the partitioning can be seen as a virtual grouping. Instead of inserting the watermark directly to the database partition, they treat it as an abstract representation of that concrete partition, such that any change in the concrete domain reflects in its abstract counterpart. Basically, such types of schemes are designed for categorical data that cannot tolerate distortion.

Another idea at the same field from the same researchers depending on building the watermark after partitioning tuples with actual attribute values, then building hash functions on top of this grouping and get a watermark as a permutation of tuples in the original table [9], [12], [13], [21]. Still, to defeat the reordering attack, a data structure is required to preserve the original location of the tuples. With this same objective, in 2009, I. Kamel [21] introduced a new distortion free watermarking scheme based on R-tree data structures as a means of data integrity. In their scheme, entries inside R-tree nodes are rearranged, relative to a ‘‘secret’’ initial order (a secret key) in a way that corresponds to the value of the watermark. To achieve that, they introduced a one-to-one mapping between all possible permutations of entries in the R-tree node and all possible values of the watermark. Without loss of generality, watermarks are assumed to be numeric values.

Other researchers have tried to address database watermarking problem from a different perspective. They have investigated ways of improving distortion free by introspectively analyzing fake (virtual) tuple or column insertion [15], [17]-[19]. For instance, the authors in [15] introduced a new technique to protect the ownership of relational database by adding only one hidden column with a secret formula where it has the ability to know the latest updates made by users. The calculation of this formula is based on the values of other numeric and textual columns. Many techniques follow the same idea with sophisticated secret functions in order to reserve against types of attacks.

Another work involving the support vector regression (SVR) for guaranteeing the database integrity underlying distortion-free of database content may be seen in [22]. In this paper, SVR is used to train highly correlation attributes and generate a predictive difference. Then it uses the Huffman coding method to build attribute characteristic table for additional payload information by particular numeric attribute. A different approach is adopted in the use of fake tuples by using probability distributions to determine the properties of the mark. Hence, there is no private key used to generate the marks into the relation [14]. Generally, it is a big challenge to figure out what and how many fake tuples should be inserted into the relation. This is because marks should not by any means degrade the quality of the data. Besides, the size of the database is increased automatically by many tuples.

In order to solve fake tuple approaches problem, numerous researchers have proposed zero-watermarking using trusted third party methods [23]-[25]. These approaches rely on partitioning the

database relation into independent square matrix groups. Then, group-based watermarks are securely generated and registered in a trusted third party. An insight into the potential benefits of using trusted timestamp valued models for database watermarking is provided in [25], where the owner's private key along with a timestamp issued by a trusted timestamp authority is embedded to protect the database from the threats of secondary watermark addition attack. From the survey conducted, it has been inferred that the existing distortion free methods have the following limitations like preserving the data usability constraints and the integrity verification when various attacks occur, and it may retrieve the tamper attribute up to group level [24].

In summary, all of these approaches to distortion-free have led to greatly improved watermarking algorithms for database copyright protection. They attempt to improve the ability to resist the majority of attacks by adopting watermarking with the deeper knowledge of database characteristics that may go some way to predicting the true adaptability of the distortion free. However, none use actual adaptation knowledge, and this ultimately limits the robustness of their methods. This paper is an attempt to step in this direction by providing an adapting tuples values selection algorithm to build fake tuples instead of using conventional secret function by using genetic algorithm together with the idea of embedding the watermark in a separate file for robust distortion free database watermarking that can deal with attacks against watermark itself and over data itself.

3. The Solution

Despite the intensive research in the area of database security, building a robust distortion free database watermarking remains a holy grail. This paper proposes a new fake tuples-based distortion free watermarking using the genetic algorithm, which tackles the problems mentioned above. Being different from other approaches of the same category, this new method creates a so-called Separable Fake Tuples Watermark (SFTW) based on hash function to select tuple for each partition and adopting genetic algorithm to create the fake tuples values for numerical attributes such that these values are near optimal solution (near to real values and taking into account the difference between attribute's values). For other non-numerical attributes, the most frequent value within the attributed is selected to be embedded in the fake tuples.

This technique has many advantages over existing techniques. (1) It is available for any relational database. (2) It does not require any additional time because the calculations required for the new records are done offline. (3) It is not possible to delete the watermark because it has been tapped in a separate file known only by the data owner (may be encrypted for more security). (4) Allowable for any update such as adding rows and changing the values of the columns. (5) It does not depend on any particular type of attributes (categorical, numerical, and text). (6) There is no need of original data for watermark detection so it is a fully blind scheme. (7) Nevertheless, it does not have sophisticated requirements or infrastructure on either database design or administration. (8) It is partition based, we are able to detect and locate modifications as we can trace the group which is possibly affected when a tuple tampers. (9) Neither watermark generation nor detection depends on any correlation or costly sorting among data items. Each tuple in a table is independently processed; therefore, the scheme is particularly efficient for tuple oriented database operations. (10) Here, only three fake tuples for each partition are sufficient to protect data inside each table.

The proposed system has two procedures: watermark embedding procedure and watermark extraction procedure as illustrated in Fig. 1. The two procedures are described in the following sub-sections. The idea behind this algorithm is based on adapting GA as a heuristic method for building fake tuples for each partition that are stored in a separate file.

3.1. Watermark Embedding

The input to the watermark encoding algorithm is dataset D, secret key Ks and the number of partitions m which are known only to the owner. The dataset D is a database relation with scheme D ($P, A_0, \ldots, A_{N-1}$), where P is the primary key attribute, $A_0, \ldots, A_{N-1}$ are N attributes, and $|D|$ is the number of tuples in relation D. The encoding algorithm results in a separate file that contains 3 fake tuples per partition (total of $3m$ fake tuples) without any modification for the data or the schema.

3.1.1 Data partitioning

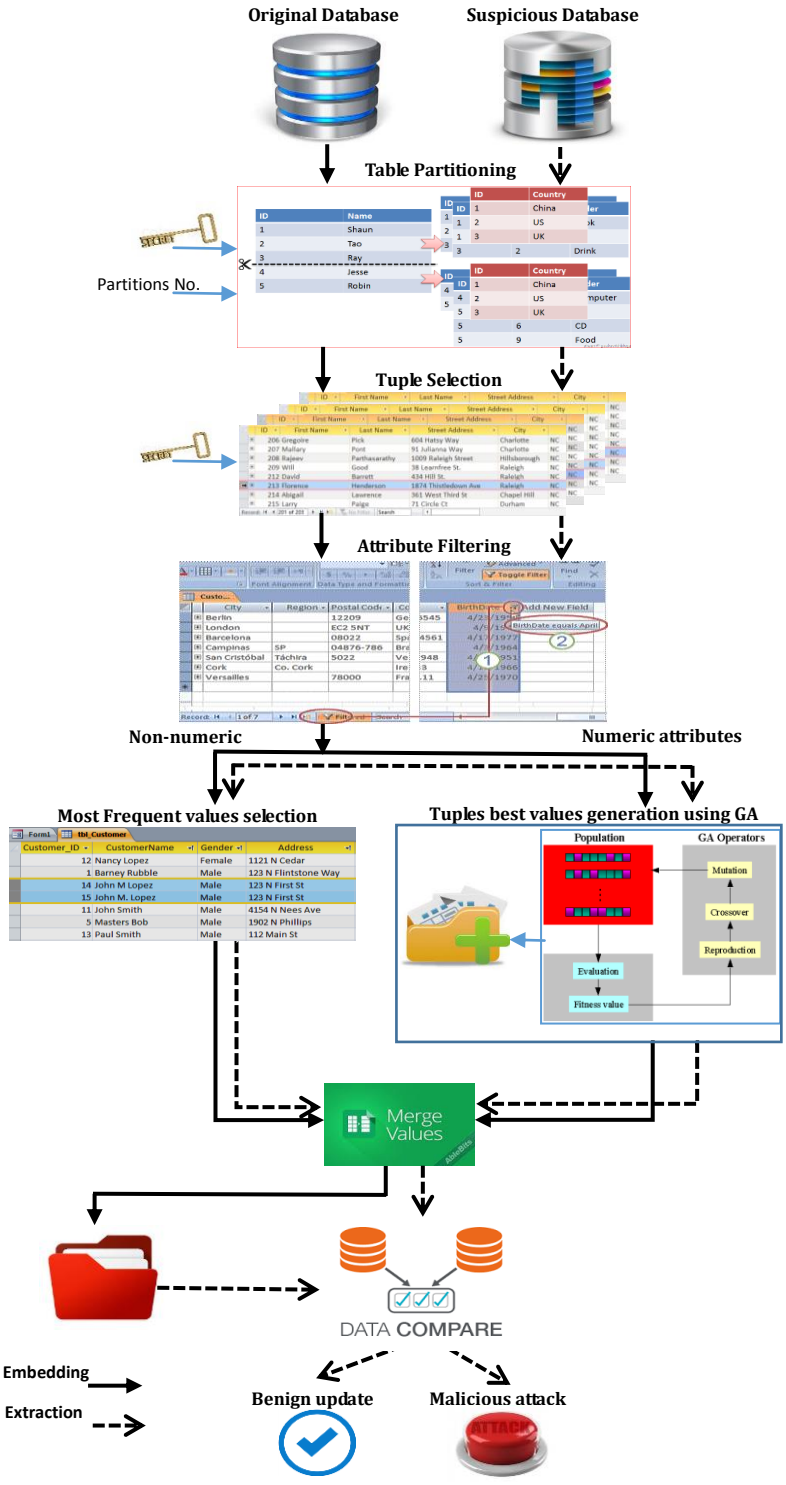

Fig. 1. The proposed distortion free database watermarking using GA.

The dataset D is to be partitioned into m non-overlapping partitions, namely, $S_0,...,S_{m-1}$ as illustrated in Algorithm 1 such that each partition S_i contains on the average $|D|/m$ tuples from the dataset D [10], [24], [26]-[35]. At line 3, the data partitioning algorithm computes a message authenticated code (MAC) for each tuple $r \in D$, which is considered to be secure and is given by $H(Ks||H(P||Ks)$, where $H(\ )$ is a secure hash function, and $||$ is the concatenation operator. In general, using the computed MAC, tuples are assigned to partitions. For a tuple r, its partition assignment is given by Partition(r) = $H(K_s||H(P||K_s))$ mod m. Using the property that secure hash functions generate uniformly distributed message digests this partitioning technique, on average, places $|D|/m$ tuples in each partition. The MAC value protects both a message's data integrity as well as its authenticity, by allowing verifiers (who also possess the secret key) to detect any changes to the message content. Furthermore, an attacker cannot predict the tuples-to-partition assignment without the knowledge of the secret key K_s and the number of partitions m, which are kept secret.

Algorithm1: *Data Partitioning*

---

1. $S_0,...,S_{m-1} \leftarrow \{\ \}$
2. For each tuple $r \in D$
3. Partition(r) $\leftarrow H(Ks\ ||H(P||Ks))$ mod m
4. Insert r into $S_{partition(r)}$
5. Return $S_0,\ldots S_{m-1}$

3.1.2 Tuple selection

In this step, a cryptographic hash function Message Digest 5 (MD5) is applied to each partition S_i to select only one tuple (located in the middle locations inside the partition) which has a zero hash value (see Algorithm 2, where Ns is the number of tuples inside the partition). This tuple is used later to create the other two fake tuples using the genetic algorithm. In our case, the selected tuple for each partition is the first tuple achieves the condition $H(P \| Ks) \bmod N_s$ equals zero within the second third of the partition. The tuple selection aims to find informative and representative tuple to assist the genetic algorithm in creating other two fake tuples. There are primarily three main reasons for performing tuple selection [30]: (a) to reduce computation time by ignoring redundant tuples, (b) to reduce the cost of labeling as superfluous tuples are discarded, and (c) to increase the efficiency of the learning algorithm by focusing only on relevant and informative tuples.

Algorithm 2: *Tuple Selection*

---

1. For each S_i ($i=0$ to $m-1$)
2. For each tuple $r \in S_i$ do
3. Tuple _hash = $H(P \| Ks) \bmod N_s$
4. If (Tuple _hash=0) then
5. Loc =Tuple_location (r) // *determine the tuple location*
6. If ($N_s \times (1/3) < Loc_r < N_s \times (2/3)$) then
7. Selected Tuple= $r(Loc)$; break

3.1.3 Attribute filtering

In general, the current distortion free database watermarking methods are limited by numerical and to some extent categorical attributes. However, non-numeric attributes occupy a large space within databases; so priority should also be given to these attributes during the watermarking process. The suggested system employs both types of attributes so that it can deal with a variety of databases. To achieve this goal, the proposed model performs attributes filtering by means of reading table schema that contains metadata regarding attribute type. In general, Filtering is a useful way to see only the data that you want to be processed. The suggested system depends on a static filter, not parameter-based filter in which the system defines the fields' type in order to be passed by the filter.

3.1.4 Build fake tuples

In this approach, unlike previous approaches, the algorithm concentrates on tuples with their entirety rather than a subset of their attributes. The approach aims to generate fake tuples and insert them into a separate file not inside the database (pure zero-watermarking; marks should not by any means degrade the quality of the data). In general, it is a big challenge to figure out what and how many fake tuples should be inserted into the separate file. For the number of fake tuples, we expect that this number is decided by the database owner. Regarding the creation of fake tuples, although this can be done manually by the database owner which is a viable approach, our effort is to make this process as automatic as possible. Therefore, our goal has been to develop an insertion algorithm that, with little supervision, can effectively generate the fake tuples. Moreover, this step does not require any additional time because the calculations required for the new records are done offline.

In this case, 3 fake tuples are created per each partition. One of them is the selected tuple extracted as illustrated in section 3.1.2. The other two tuples are created using the genetic algorithm for numeric attributes and most frequent values selection concept for non-numeric attributes; one tuple for each section above and down the selected tuple. The rationale of choosing 3 tuples/partition is to track any malicious changes that may be occurred inside the partition and precisely identify the section inside this partition that contains these attacks.

3.1.4.1 Based on non-numeric attributes

To be noted, regarding the non-numeric attributes, the robustness of the suggested algorithm has a close relation with marking frequency. The higher marking frequency we get, the more robust it is [36]. The proposed technique is algorithmically based on evaluating the local characteristics of database relation like data values frequency distribution [24]. The technique takes the relation D and a non-numeric attribute A_i; (i may take values from 0 to N_t-1, N_t is the number of non-numeric attributes); and determines the distinct values for A_i in D and construct the frequencies by the relative frequency (most frequency) of each value for A_i. For instance, for the attribute "carrier type" in the flight scheduling database, the distribution of the frequencies is constructed as F(Boeing) = 3/4 and F(Airbus)=1/4. So, the value for this attribute inside the fake tuple is set to Boeing. If these attributes have the same frequency, one of them is chosen randomly [14].

3.1.4.2 Based on numeric attributes

Regarding the numeric attributes, the suggested system exploits GA as an exhaustive search algorithm to create numerical values of fake tuples. A GA uses a series of steps to reach the optimum solution (the optimal values of fake tuples' numeric attributes). The first step is to select and initialize the population i.e. from where the solution (tuples inside each section; two sections in this case) is obtained and preceded to what is collectively known as the generation. Then the objective function for the problem is evaluated and the fitness function corresponding to that objective function is found. After that, a set of genetic operators comprising of reproduction, mutation, and crossover are applied. These steps are continued until the

desired criterion is reached and the optimum solution is obtained. Optimality here means that the numerical values of the fake tuples are closer to the real values and can be used to measure the potential average values that may occupy by the attributes. Furthermore, the difference between these values and values within the selected tuple is little as possible. In details, the steps of GA are as follows [37]:

- **Define the fitness function**: in this case the fitness function is computed as: $f = \sum_{j}^{N-N_T} \left| v_{A_{r,i}} - v_{A_{s,j}} \right|$, where $v_{A_{r,i}}$ is the numeric attribute's value of the highlighted tuple inside the population, and $v_{A_{s,j}}$ is the numeric attribute's value of the selected tuple.
- **Initialization, random selection** $r \in D$ and $|r| > \zeta$, to produce the first generation of population of chromosomes (one population for each section inside the partition S_i). ζ represents a threshold for the optimal number of tuples inside the population. Each chromosome has a corresponding value of the objective function, referred to as the fitness of the chromosome
- **Roulette selection operation**: operate the initial group for roulette selection firstly, keep the good individuals.
- **Crossover operation***:* according to a certain crossover probability of individual single-point crossover operator, conventional genetic algorithm crossover probability is constant, generally between 0.5-0.8.
- **Mutation operation***:* according to a certain probability of individual variability binary arithmetic. Variability factor is made a choice between 0 and 1, where the mutation rate is not too large, generally not more than 0.5. Unlike the process of crossover, the role of mutation is to explore the search space.
- **Terminate the conditional***:* There are two conditions to stop loop termination, one is to set the maximum termination of algebra (if the number of generations which determined by the database owner is completed); another is that when the population variance between individual fitness is less than a set value, the loop is terminated. Otherwise, go to Roulette selection operation.

3.1.5 Embedding into a separate file

After obtaining the optimal values for both numeric and non-numeric attributes to be exploited for building a fake tuple, these values are combined into one tuple. As mentioned above, the system generates 3 tuples per partition. These tuples are embedded into a separate file to generate the watermark; so the file contains 3m tuples for each relation. The system can efficiently deal with the benign update by building a new watermark using the previous steps (offline); because this does not require a great time as will be explained in the experiments later.

3.2. Watermark Extraction and Verification

In the decoding phase of the watermark, the suspicious database is taken as input to the algorithm to extract the embedded watermark using the secret parameters including K_s and m. It is assumed that the primary key attribute has not been changed or else can be recovered. The watermark detection is blinded that means it neither requires the knowledge of the original data nor is the watermark. The procedure of detecting watermark is similar to the procedure of embedding watermark with an extra step that involves the watermarking verification phase. The watermarking verification phase is the process of comparing the data set watermark found in the separate file and the data set watermark from suspicious data set. The correlation coefficients of statistics are calculated. If the correlation coefficient is detected higher than the presetting value, the detection is successful, and the copyright information is determined. A very important problem in a watermarking scheme is synchronization, that is, we must ensure, that the watermark extracted is in the same order as that generated. If synchronization is lost, even if no modifications have been made, the embedded watermark cannot be correctly verified.

4. Attacks Analysis and Experiments

This section reports the intensive experiments of this study. The purpose of these experiments is to test the proposed scheme in term of robustness against several database attacks like deletion, insertion, and modification attacks. The experimental setup included a 2 GHz CPU and 12 GB RAM PC running Windows 10. Algorithms are implemented in MATLAB environment using SQL server database. We applied our algorithms to a real database (https://technet.microsoft.com/en-us/library/ms124425 (v=sql.100).aspx) with a set of 18 attributes, one is the primary key and the others are numerical and non-numerical attributes. The size of the database is 100,000 tuples (approximately 83 Mbyte). The average time required for the watermark embedding was 7 minutes (using Azure cloud system based on a server with 4 Cores, 28 GB RAM, 8 Data disks, 12800 Max IOPS, 56 GB SSD hard disk, this times is reduced to 117 seconds only). For the watermark verification, the required time is 7 minutes on average (120 seconds on Azure cloud system). These results indicate that our algorithm performs well enough to be used in real-world applications.

4.1. Subset Deletion Attack

In this attack, violators randomly drop α tuples from the database. To simulate this attack, we randomly deleted tuples in various locations with different ratios from the watermarked data. If the tuples are randomly deleted, then, on average, each partition loses α/m tuples. Fig. 2 depicts the resilience under the deletion attack for several values of the tuple dropping (0.001%, 0.002%, 0.003%... 0.14%) for the proposed algorithm. As shown in Fig. 2, the embedded watermark can be 100% lost (i.e. the matching between the embedded watermark inside the separate file and the extracted one from the suspicious database equals zero) after deleting 0.14% of the total tuples (140 tuples from 100000 tuples). We can see that the algorithm shows higher resilience when α decreases.

The successful deletion of the selected (marker) tuple inside each partition could result in a large number of errors in the decoding phase and susceptible to watermark synchronization error. To avoid watermark synchronization errors, the *m* marker tuples should be stored. Already the proposed system stores the selected tuple as one of the three tuples representing watermark for each partition. Note that, our partitioning technique is resilient to such synchronization errors as it does not rely on marker tuples to locate the partition limits; instead, our partitioning technique assigns tuples to partitions based hash function. One advantage of the proposed system is that the deletion of tuples from the database does not cancel any tuple from the watermark because it is kept in a separate file.

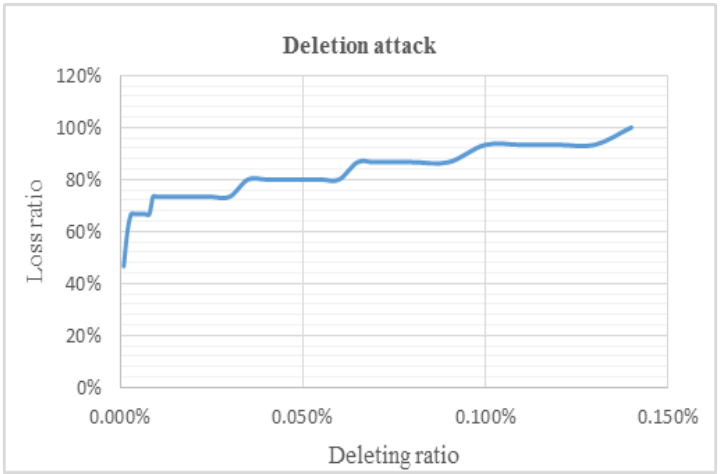

Fig. 2. Loss in watermark detection after deletion attack.

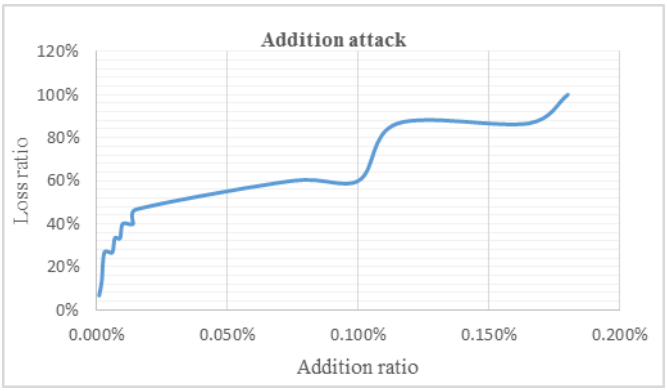

Fig. 3. Loss in watermark detection after addition attack.

4.2. Subset Addition Attacks

In this attack, the attacker adds a set of α tuples to the database. To simulate this attack, we randomly add different ratios of the watermarked data in different locations. Also, if the tuples are randomly inserted, then, on average, each partition increases with α/m tuples. Fig. 3 depicts the resilience under the addition attack for several values of the tuple insertion (0.001%, 0.002%, 0.003%... 0.18%) for the proposed algorithm. We can see that the results are quite similar to the deletion attack. The embedded watermark can be 100% lost after inserting 0.18% of the total tuples (180 tuples from 100000 tuples). We can see that the algorithm shows higher resilience when α decreases. The added tuples won't change valuable attributes of the original data set, but they make examining watermark information more difficult.

In our approach, if the insertion reaches a certain threshold, by comparing the watermarks stored in the separated file and the extracted watermark for each partition, the tampering can be easily detected. Using the number of groups should not argument our technique blindness because the number of the group is preserved by the owner; it is used in watermark integrity checking process as an input to the watermark function. In this case, the tuples addition does not affect the watermark because it is added to a separate file but may result in a different watermark during the extraction process.

4.3. Subset Modification Attacks (Case Alteration Attacks)

Table 1. Loss in Watermark Detection after Modification Attack in 4 Attributes Using 5 Partitions

No. of Tuples	Change ratio	Effect ratio
100	0.02%	0%
150	0.03%	1%
200	0.04%	2%
300	0.07%	10%
600	0.13%	10%
700	0.16%	12%
800	0.18%	12%
900	0.20%	13%
1000	0.22%	14%
1100	0.24%	15%
1200	0.27%	16%
1300	0.29%	16%
1450	0.32%	16%
2100	0.47%	16%

In this attack, violators try to randomly select and modify random attributes A_j in α tuples in the watermarked relation (the relation in our case doesn't contain the watermark). Those modified tuples may

perturb the watermark extraction process, but in our method, the result shows high resilience against this attack. In this experiment, we modified four attributes randomly for the whole data set. Table 1 shows the result of this attack. The system's ability to counterfeit this attack is very high. This can be explained due to the utilization of both the genetic algorithm and most frequent values selection mechanisms. In general, an attacker cannot destroy the watermark because it is stored in a separate file and not in the tuples itself. In the case when the change affects the selected tuple inside each partition, the watermark in the extraction process changed. Any situation resulting from this case can be solved by retrieving the selected tuple from the stored watermark. In the case of benign attacks that change the selected tuple, the watermark embedding process can be recalculated in time schedule to reflect the changed attributes. In summarization, the results show that only 16% distortion in the watermark is observed with 0.47% random modifications. So it is more robust against such kind of attacks.

4.4. Security Analysis

Watermarking techniques can be classified as robust or fragile. Robust watermark, which is used mainly for copyright protection, should be able to withstand modification attempts like cropping, compression, transformation, etc. On the other hand, fragile watermarks which are used for data integrity should be sensitive to modifications. 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. In this case, any change that may occur in the database' data does not effect on the stored watermark and thus facilitates the process of integration. Furthermore, the secret key used to generate the database partitioning and tuple selection for each partition is very important as far as security aspects are concerned. In order to reduce the chances of security leakage, this secret key is changed every time we generate the watermark after each benign update.

For the role of GA in the security inside the process of generating the watermark. GA has introduced randomness in the creation of the data within each fake tuple; so, it is very difficult for an attacker to predict the fake tuple. This helps to improve the overall security of the watermark. Thus, GA itself provides help to guard against an attacker. The invisibility of the watermark also adds to the overall security of the watermark, because it helps to hide the distortion according to the neighborhood values. Therefore, the attacker will find it hard to predict the attributes that are marked. The proposed system relies on a new idea to differentiate between benign update and the malicious attacks on the basis that the owner will recreate the watermark in the case of the benign update and save it in the independent file because the time required to recalculate a new watermark is very small. Therefore, any changes in the database will be considered as a malicious attack.

5. Conclusion

In this paper, we identified the problem of tamper detection and localization for a database relation with different types of attributes and proposed a novel fragile watermarking scheme to address this problem based on intelligent creation of fake tuples in a separate file not inside the database; thus, it is distortion free. In the proposed scheme, all tuples in a database relation are first securely divided into groups according to some secure parameters. The watermarking process has an advantage over hash function, as it does not depend on the ordering of the tuples. The fact that this algorithm is group based, yields to the following strength points: (1) we are able to detect and locate modifications as we can trace the group which is possibly affected when a tuple tampers. (2) this watermarking technique can be tuned according to different security levels, by returning the partitioning through the use of multiple attributes. (3) Neither watermark generation nor detection depends on any correlation or costly sorting among data items. Each

tuple in a table is independently processed; therefore, the scheme is particularly efficient for tuple oriented database operations. (4) The proposed system introduced a new method for distinguishing between benign updates and malicious updates. Any modification in the watermark was considered the result of malicious updates. In the case of benign updates, a new watermark will be calculated. The future research will be directed towards increasing the level of attack resilience against several sources of attacks in the watermarking method. Finally, different applications for our proposed technique could be envisioned and enforced.

References

[1] Guo, H., & Jajodia, S. (2004). Tamper detection and localization for categorical data using fragile watermarks. *Proceedings of the ACM Workshop on Digital Rights Management* (pp. 73-82).

[2] Agrawal, R., Haas, P. J., & Kiernan, J. (2003). Watermarking relational data: Framework, algorithms and analysis. *Very Large Data Bases Journal*, *12(2),* 157-169.

[3] Agrawal, R., & Kiernan, J. (2002). Watermarking relational databases. *Proceedings of the Very Large Database Conference* (pp. 155-166).

[4] Mehta, B., & Rao, U. (2011). A novel approach as multi-place watermarking for security in database. *Proceedings of the International Conference on Security and Management* (pp. 703-707).

[5] Lei, Z., & Li, R. (2013). Research of applications in relational database on digital watermarking technology. *International Journal of Engineering Science Invention, 2(9),* 84-89.

[6] Bilapatte, S., Bhattacharya, S., & Sawarkar, S. (2014). A review on watermarking relational databases. *International Journal of Applied Engineering*, *4(2),* 89-96.

[7] Rathva, M., & Sahani, G. (2013). Watermarking relational databases. *International Journal of Computer Science, Engineering and Applications*, *3(1),* 71-79.

[8] Singh, P., & Chadha, R. (2013). A survey of digital watermarking techniques applications and attacks. *International Journal of Engineering and Innovative Technology*, *2(9),* 165-175.

[9] Bhattacharya, S., & Cortesi, A. (2009). A distortion free watermark framework for relational databases. *Proceedings of the International Conference on Software and Data Technologies* (pp. 229-234).

[10] Bhattacharya, S., & Cortesi, A. (2009). A generic distortion free watermarking technique for relational databases. *Proceedings of the International Conference on Information Systems Security* (pp. 252-264).

[11] Bhattacharya, S., & Cortesi, A. (2010). Distortion-free authentication watermarking. *Proceedings of the International Conference of Software and Data Technologies* (pp. 205–219).

[12] Li, M., & Zhao, W. (2011). An asymmetric watermarking scheme for relational database. *Proceedings of the International Conference on Communication Software and Networks* (pp. 180–184).

[13] Arun, R., Praveen, K., Bose, D., & Nath, H. (2012). A distortion free relational database watermarking using patch work method. *Proceedings of the International Conference on Information Systems Design and Intelligent Applications* (pp. 531-538).

[14] Pournaghshband, V. (2008). A new watermarking approach for relational data. *Proceedings of the Annual Southeast Regional Conference* (pp. 127-131).

[15] Zawawi, N., El-Gohary, R., Hamdy, M., & Tolba, M. (2012). A novel watermarking approach for data integrity and non-repudiation in rational databases. *Proceedings of the International Conference on advanced Machine Learning Technologies and Applications* (pp.532-542).

[16] El-Bakry, H., & Mastorakis, N. (2009). A new watermark approach for protection of databases. *Proceedings of the International Conference on Applied Informatics and Communications* (pp. 243-248).

[17] El-Bakry, H., & Hamada, M. (2010). A novel watermark technique for relational databases. *Proceedings of the International Conference on Artificial Intelligence and Computational Intelligence* (pp. 226-232).

[18] El-Bakry, H., & Hamada, M. (2010). A developed watermark technique for distributed database security. *Proceedings of the International Conference on Computational Intelligence in Security for Information Systems* (pp. 173-180).

[19] Gamal, G., Rashad, M., & Mohamed, M. (2008). A simple watermark technique for relational database. *International Journal of Intelligent Computing and Information Science*, *8(1),* 92-101.

[20] Bhattacharya, S., & Cortesi, A. (2010). Database authentication by distortion free watermarking. *Proceedings of the International Conference on Software and Data Technologies* (pp. 219-226).

[21] Kamel, I. (2009). A schema for protecting the integrity of databases. *Computers and Security*, *28(7),* 698-709.

[22] Wu, H., Hsu, F., & Chen, H. (2008). Tamper detection of relational database based on SVR predictive difference. *Proceedings of the International Conference on Intelligent Systems Design and Applications* (pp. 403-408).

[23] Hamadou, A., Sun, X., Gao, L., & Shah, S. (2011). A fragile zero-watermarking technique for authentication of relational databases. *International Journal of Digital Content Technology and its Applications, 5(5),*189–200.

[24] Camara, L., Li, J., Li, R., & Xie, W. (2014). Distortion-free watermarking approach for relational database integrity checking. *Mathematical Problems in Engineering*, *2014(1),* 1–10.

[25] Iqbal, S., Rauf, A., Javed, H., & Ahmad, S. (2011). Distortion free algorithm to handle secondary watermark attack in relational databases. *Proceedings of the European Conference on Information Management* (pp. 214-221).

[26] Mayekar, A., Jha, M., Mule, S., & Shridattopasak, C. (2014). Relational database watermarking. *International Journal of Engineering Research and Technology*, *2(7),* 248-252.

[27] Kamran, M., Suhail, S., & Farooq, M. (2013). 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.

[28] Radha, R., Sankari, K., & Devi, S. (2016). Implementation of invisible watermarking technique in relational database. *International Journal of Recent Scientific Research*, *7(4),* 10034-10037.

[29] Sonupriya, S., & Rani, R. (2014). The digital watermarking technique for numerical relational databases. *International Journal of Innovations & Advancement in Computer Science*, *3(9),* 14-21.

[30] Panimalar, S., & Srinath, D. (2015). Reversible watermarking technique based on time stamping in relational data. *International Journal of Innovations & Advancement in Computer Science, 2(1),* 961-967.

[31] Waichal, S., Bhandure, M., Waghmare, U., & Meshram, B. (2013). Watermarking databases. *Journal of Engineering, Computers & Applied Sciences*, *2(6),* 81-88.

[32] Khanduja, V., Verma, O., & Chakraverty, S. (2015). Watermarking relational databases using bacterial foraging algorithm. *International Journal of Multimedia Tools and Applications*, *74(3),* 813-839.

[33] Melkundi, S., & Chandankhede, C. (2015). A robust technique for relational database watermarking and verification. *Proceedings of the International Conference on Communication, Information & Computing Technology* (pp. 1-7).

[34] Camara, L., Li, J., Li, R., Kagorora, F., & Hanyurwimfura, D. (2014). Block-based scheme for database integrity verification. *International Journal of Security and its Applications*, *8(6),* 25-40.

[35] Khanduja, V., Chakraverty, S., & Verma, O. (2015). Watermarking categorical data: Algorithm and robustness analysis. *Defense Science Journal*, *65(3),* 226-232.

[36] Cui, X., Sheng, G., & Zheng, J. (2006). A robust algorithm for watermark numeric relational databases. *Lecture Notes in Control and Information Sciences*, *344(1),* 810-815.

[37] Jawad, K., & Khan, A. (2013). Genetic algorithm and difference expansion based reversible

watermarking for relational databases. *Journal of Systems and Software*, *86(11),* 2742-2753.

Mohamed M. Elsherbiny received his B.Sc and M.Sc at EED, AU, Faculty of Engineering, EED and Ph.D at McMaster University, Hamilton Ontario, Canada, (EED). Currently, he works at Alexandria University (AU), Institute of graduate studies and research (IGSR). He has around 31 MSc and PhD theses and 18 papers in IT field and 10 theses and 5 papers in materials science field. He is a visiting professor at Waterloo University, Canada (EED). 1995, he was working on electrical safety grounding system, software development. He worked as a chairman of computer engineering department, University of Beirut – Lebanon, Faculty of Engineering, Sultan Qaboos University, and Dean of engineering (computer science) - Canadian University of Dubai. His present fields of interests are PV solar cells, software development, information technology, computer real time measurements, and safety grounding systems.

Saad M. Darwish received the B.Sc. degree in statistics and computer science from the Faculty of Science, Alexandria University, Egypt in 1995. He held the M.Sc degree in information technology from the Institute of Graduate Studies and Research (IGSR), Department of Information Technology, University of Alexandria in 2002. He received his Ph.D degree from the Alexandria University for a thesis in image mining and image description technologies. He is the author or coauthor of 50+ papers publications in prestigious journals and top international conferences and also received several citations. He has served as a reviewer for several international journals and conferences. He has supervised around 60 M.sc and Ph.D students. His research and professional interests include image processing, optimization techniques, security technologies, database management, and machine learning. Since July 2017, he has been a professor in the department of information technology, IGSR.

Hosam A. Selim received his M.Sc degree in information technology from the Alexandria University, Egypt. Currently, he is a Ph.D student in the Department of Information Technology, Institute of Graduate Studies and Research, Alexandria University, Egypt. He worked as a lecturer to teach many computer science subjects such as databases and programming. His research interests are in the area of database security.