International Journal Of Artificial Intelegence Research
Vol 3, No 1, Juny 2019, pp. 1-10
ISSN: 2579-7298

Design and Analysis of an Intelligent Integrity Checking Watermarking Scheme for Ubiquitous Database Access

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

Department of Information Technology, Institute of Graduate Studies and Research, Alexandria University, Egypt[1,2]
saad.darwish@gmail.com[*]

ARTICLE INFO

Article history:
Received: 11may 2018
Revised: 06 August 2018
Accepted:09 December 2018

Keywords:
Ubiquitous database accessing,
Distortion free watermarking,
Genetic algorithm,
intelligent Content Integrity mechanism

ABSTRACT

As a result of the highly distributed nature of ubiquitous database accessing, it is essential to develop security mechanisms that lend themselves well to the delicate properties of outsourcing databases integrity and copyright protection. Researchers have begun to study how watermarking computing can make ubiquitous databases accessing more confident work environments. One area where database context may help is in supporting content integrity. Initially, most of the research effort in this field was depending on distortion based watermark while the few remaining studies concentrated on distortion-free. But there are many disadvantages in previous studies; most notably some rely on adding watermark as an extra attributes or tuples, which increase the size of the database. Other techniques such as permutation and abstract interpretation framework require much effort to verify the watermark. The idea of this research is to adapt an optimized distortion free watermarking based on fake tuples that are embedded into a separate file not within the database to validate the content integrity for ubiquitous database accessing. The proposed system utilizes the GA, which boils down its role to create the values of the fake tuples as watermark to be the closest to real values. So that it's very hard to any attacker to guess the watermark. The proposed technique achieves more imperceptibility and security. Experimental outcomes confirm that the proposed algorithm is feasible, effective and robust against a large number of attacks.

I. Introduction

Information security in ubiquitous contexts remains to be challenging still underdeveloped. Several strategies either neglect security concerns completely or attempt to implement conventional tools that do not adjust themselves adequately to the profoundly dynamic and actually distributed nature of ubiquitous computing settings. A number of mechanisms have been stated to afford cyber data security, including cryptography and steganography [1]. Cryptography preserves data by encrypting them. Still, once the encrypted data are decrypted, the data are transparent and are no longer under protection. On the other hand, steganography hides the presence of the data by masking 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 methods do not stop copying rather it obstructs unauthorized copying by offering a means of building the original owners evidence inside the redistributed copies. 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. Nevertheless, watermarking databases have individual specifications that oppose from those claimed for watermarking digital audio or video [2-5]. These specifications cover: (1) Few redundant data: in multimedia data, there is a large quantity of space accessible to store the watermark, while the database consists of the tuples, each tuple denotes a separate object so, the watermark is expanded across these isolated objects, (2) out-of-order relational data: The relative spatial/temporal locations

of various components of multimedia objects do not replace, whereas there is no arrangement between the tuples in database as the combination of tuples is viewed as a set, and (3) Frequent updating: Multimedia objects typically reside unimpaired; any part of such an object is not cut or substituted arbitrarily without producing perceptual distortions in the object, whereas tuples may be inserted, deleted, or updated throughout regular database transactions. Due to these disagreements, we cannot immediately accept any of the routines as it is for the database, which formed 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 benign update attacks which may destroy or remove 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 location of the watermark is not discovered by an attacker. The fourth important property is detectability which means that the watermark should ascertain the owner's identification throughout the detection/extraction stage unless the watermark is ineffective. 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 interchange of tuples' locations using a linear permutation un-ranking procedure to rise the embedding dimensions [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 in which its values are recognized only by the data holder [14-19]. This type of watermarking has the facility to distinguish the up-to-date updates made by the 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 partition-based watermarks are securely created and enrolled in a trusted third party. Finally, (5) Zero-watermark technique in which the watermarks are created by means of using the internal features of database relation, like frequency distribution of several digits, and oscillates of data values [20-23].

Degrading content integrity checking in ubiquitous database access, these techniques are more powerful against little types of attacks such as deleting or updating cell values. Still, most of these distortion free database 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 structured as follows: Section 2 examines the related work and emphasizes the limitations of these techniques. An outline of our watermarking technique is explained 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 reported in Section 4. Finally, conclusions are conveyed in Section 5.

II. Research Method

Despite the intensive research in the area of content integrity checking in ubiquitous database accessing; building a robust distortion free database watermarking for content integrity remains a holy grail. This paper proposes a new fake tuples-based distortion free watermarking based bio-

inspired generic algorithm that copes with the difficulties stated above. unlike other approaches of the same branch, this new method creates a so-called Separable Fake Tuples Zero Watermark (SFTZW) based on hash function to select tuple for each partition and adopts genetic algorithm to build 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 approach has several gains in excess of current content integrity checking techniques. (1) It is accessible from any database schema. (2) It is not feasible to remove the watermark as it has been embedded into a separate file identified only by the data owner (may be encrypted for more security). (3) It does not depend on any particular type of attributes (categorical, numerical). (4) There is no need for original data for watermark detection so it is a fully blind scheme. (5) Yet, it does not have complicated prerequisites or infrastructure on either database design or management. (6) It is partition based, anyone able to identify and localize changes as he can follow the group which is perhaps influenced once a tuple tampers. (7) Neither watermark formation nor detection is subject to any correlation or costly arrangement among data items. Each tuple in a table is individually processed; so, the scheme is mainly well-organized for tuple oriented database processes. (8) Here, only three fake tuples for each partition are sufficient to protect all types of data. The proposed system has two stages: watermark embedding and watermark extraction 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.

The input to the watermark encoding algorithm is dataset *D*, secret key *Ks* and the number of partitions *m* that are known only to the owner. The dataset *D* is a database relation with scheme *D* (*P, A0, . . .,AN-1*), where *P* is the primary key attribute, *A0, . . .,AN-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 3*m* fake tuples) without any modification for the data or the schema.

The dataset *D* is to be divided into *m* non-overlapping groups, namely, $S_0,\ldots,S_{m-1}$ such that each partition S_i holds on the average |*D*|/*m* tuples from the dataset *D*; see [10][24][26-35] for more details. In this case, an attacker cannot guess the tuples-to-partition transfer without the awareness of the secret key K_s and the number of partitions *m*, which are reserved secret. This step helps to achieve localized tampering detection. After that, 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 (research assumption; other value from 1 to 15 can be used provided that each partition must have a watermark). 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.

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.

Unlike previous approaches, the on-hand procedure concentrates on the whole tuple's attributes rather than a subset of these attributes. The approach purposes to build fake tuples as watermarks and appends them into a separate file not inside the database (pure zero-watermarking). In general, it is a big problem to realize what and how many fake tuples should be injected into the separate file. For fake tuples number, we assume that this number is determined by the database owner. Our aim has been to exploit a zero watermarking algorithm that, with little management, can successfully create the fake tuples.

In this case, three fake tuples are created per each partition. One of them is the selected tuple extracted from each partition with a zero hash value. 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.

Regarding the non-numeric attributes, the suggested procedure has a close kin with marking frequency. The higher marking frequency we catch, the more robust it is [36]. The proposed technique is algorithmically based on assessing the internal features of database relation such as 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 regulates the distinct values for A_i in D and build the frequencies by the relative frequency (most frequency) of each value for A_i. For example, for the attribute "carrier type" in the flight scheduling database, the distribution of the frequencies is assembled 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].

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]:

Determine the fitness function: in this case the fitness function is computed as:

$f = \sum_{j}^{N-N_t} \left| v_{A_{r,j}} - v_{A_{s,j}} \right|$, where $v_{A_{r,j}}$ 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. In this case, the objective is to minimize f.

Initialization, random selection $r \epsilon D$ to create the initial generation of chromosomes (one population for each section inside the partition S_i). Each chromosome has a corresponding value of the objective function, referred to as the fitness of the chromosome. *Roulette selection operation*: pass the initial group of chromosome for roulette selection to keep the good individuals. *Crossover operation:* with a certain crossover likelihood (between 0.5-0.8) individual, a single-point crossover operator is used to vary chromosomes from one generation to the next by taking more than one parent solution and producing a child solution from them. *Mutation operation:* with a certain probability of individual; generally not more than 0.5, binary mutation is used to discover the search space. *Terminate condition:* if the number of generations that are identified by the database owner is completed, the loop is terminated. Otherwise, go to Roulette selection operation.

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. In the extraction phase, the suspicious database is passed as input to the algorithm to extract the watermark with the knowledge of secret parameters including Ks, and m. It is supposed that the primary key attribute has not been altered or else can be recovered. The watermark extraction is blinded that means it does not need the original data. 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 stage is the procedure of comparing the watermark data set found in the separate file and the watermark data set from suspicious data. The correlation coefficients of statistics are computed. If the correlation coefficient is detected higher than the presetting value, the detection is successful, and the content integrity (tampering) is determined. A very important issue in a watermarking scheme is synchronization, that is, we must confirm, that the watermark extracted is in the same order as that generated. If synchronization is missing, even if no changes have been made, the embedded watermark cannot be appropriately verified.

In this, the suggested system can tune the parameter such that with the lower value, that system can be used for copyright protection confirmation. For the current implementation, the suggested system uses this parameter with value equal 100 for content integrity checking such that during the comparison between the two watermarks (original and extracted), there must be the full similarity between the corresponding attributes to achieve verification. To summarize, the pseudo code of the suggested watermark extraction algorithm is as follows

III. Analysis and Result

This section reports the experiments and analyses of this study. The aim of these investigations is to evaluate the suggested scheme regarding its strength (content integrity) against a number of database attacks like deletion, insertion, and modification. The experimental configuration controlled within a 2 GHz CPU and 12 GB RAM PC running Windows 10. The system modules are implemented in MATLAB environment using SQL server database component. The system is applied to a real database (https://technet.microsoft.com/en-us/library/ms124425 (v=sql.100).aspx) with a set of 18 attributes containing numerical and non-numerical values. The size of the database was 100,000 tuples (approximately 83 Mbyte). The average time required for 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, and 56 GB SSD hard disk, this times is reduced to 117 seconds). For the watermark verification, the involved time is 7, 5 minutes on average (120 seconds on Azure cloud system). These results indicate that, the suggested system carries out in a good way to be applied in real-world applications. The utilized GA parameters configuration is illustrated in Table 1.

Table1: Genetic algorithm parameters.

GA Parameter	Commonly used Value	Value for Implemented System
No. of Generations (N_g)	100	100
Populations Size (S_p)	50	50
Chromosome Length	3	No. of database' numeric attributes Dataset 1= 10 , Dataset 2= 13
Selection Mechanism	Tournament Selection Tournament Size = 5	Tournament Selection Tournament Size = 5
Crossover	Type: Single Point Fraction: 0.7	Type: Single Point Fraction: 0.7
Mutation	Type: Uniform Rate: 0.1	Type: Uniform Rate: 0.1
Elitism Count	2	2

In this attack, attackers arbitrarily delete α tuples from the suspicious database. To simulate this attack, we randomly deleted different ratios of the watermarked data. If the tuples are arbitrarily erased, then, on average, each section drops $\frac{\alpha}{m}$ tuples. Fig. 2 depicts the system resilience under the deletion attack for several values of the tuple dropping (0.001%, 0.002%, 0.003%… 0.14%). 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 only 0.14% of the total tuples (140 tuples from 100000 tuples). The experiments show that the suggested approach is resilient to deleting attack even if the rate of the deleted tuple is small. In this case, based on the partitioning process, the system can easily realize that the suspicious database tuples in each group are smaller than the expected one; then, the tampering is proved.

The successful deletion of the selected (marker) tuple inside each partition leads to a large number of errors in the decoding phase and vulnerable 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 robust to such synchronization errors as it does not depend on marker tuples to locate the partition boundaries; as an alternative, the utilized partitioning technique allocates tuples to the 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.

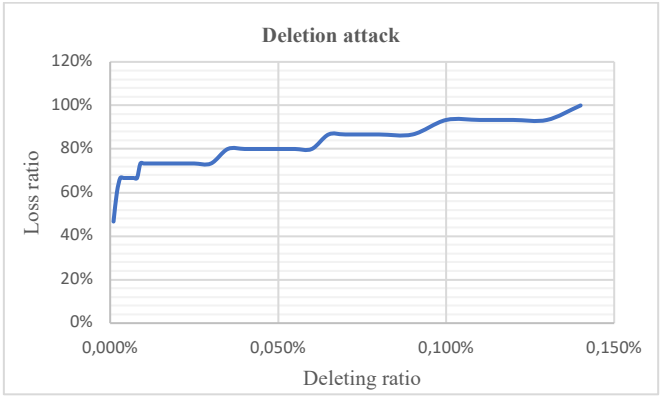

Fig. 2: Loss in watermark detection after deletion attack

In this attack, the invader adds a set of α tuples to the original database. To simulate this attack, we randomly add different ratios of the watermarked data. If the tuples are randomly inserted, then, on average, each partition increases with $\boxed{\frac{\alpha}{m}}$ 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 algorithms. Fig. 3 reports the result for this attack. We can see that the results are quite similar for the same reason explained in 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. In this case, if the insertion extents a certain threshold, by comparing the watermarks stored in the separated file and the extracted watermark for each partition, the tampering can be effortlessly perceived. Herein, the tuples addition does not affect the watermark because it is stored to a separate file but may result in a different watermark during the extraction process.

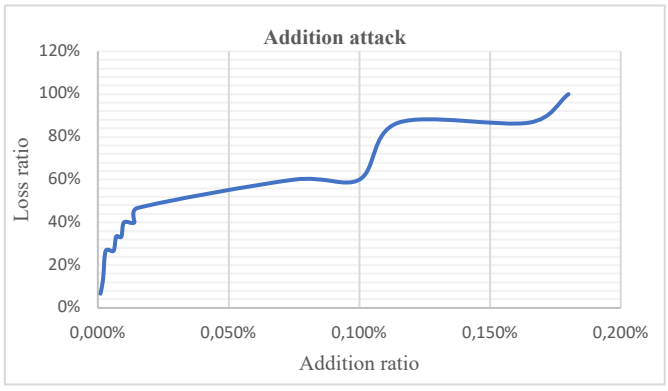

Fig. 3: Loss in watermark detection after addition attack

In this attack, attackers try to arbitrarily choice and change random attributes A_j in α tuples in the watermarked relation (the relation in our case doesn't contain the watermark). Those altered tuples may disturb 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 2 show the result of this attack respectively. It can be seen that the results in this attack are

very high. This can be explained due to utilizing both of the genetic algorithm and most frequent values selection mechanisms. In general, any 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 can be recalculated in time scheduling to reflect the changed attributes. In summarization, the results show that only 16% distortion in watermark is observed with 0.47% random modifications. So it is more robust against such kind of attacks.

Table 2: 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%

This experiment compares the suggested GA-based distortion free watermarking approach with L. Camara technique [24]. These results shown in Table 3obviously validate that L. Camara method is not resilient enough to common attacks, while with the suggested scheme even a minor data change has a significant impact on the extracted watermark. By identifying the number of groups used in the watermark generation procedure, after immense deletion or insertion attacks, anyone may have one or more imperfect groups than the expected one. For all types of attacks in fragile watermarking, the aim of an attacker is to change the original database by keeping the watermark intact. The superiority of the proposed system is due to the exploiting of a genetic algorithm to build watermark. As genetic operators are not purely random but use randomness to (often slightly) change individuals, it becomes difficult for the attacker to modify database content without changing the watermark.

Table 3: Comparison with L. Camara approach for database integrity checking

	The proposed system		L. Camara approach		Comment
	No. of tuples	Similarity Loss (%)	No. of tuples	Similarity Loss	
Insertion attacks	**1**	7	**1**	0.5	Detect and localize the tampering at group level
	90	100	**90**	19	
Deletion attacks	**1**	40	**1**	5	
	1300	100	**1300**	23	
Modification attacks	**150**	1	**150**	0.01	
	2100	16	**2100**	2	
Database partitioning	Partition based technique		Partition based technique		
Security	Secret key based technique		Secret key based technique		
Attribute Type	All types		Numeric attributes		

In general, distortion-free usually utilizes a fragile watermarking scheme and is used for tamper detection and content integrity, whereas distortion-based is considered robust and is recommended for ownership protection. As the proposed approach is fragile in nature, the suspicious database can

be matter to numerous attacks with the aim to cruelly change the protected data while not touching the certified watermark. Herein, the realization of the use of chance to alter the content of the database while retaining the watermark undamaged is examined. 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 trickle, this secret key is reformed every time we create 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 complicated for an invader to guess the fake tuples. This assistances to increase the intact security of the watermark. Therefore, GA offers aid to protect against an attacker. The hiddenness of the watermark as well increases to the whole security of the watermark, since it services to conceal the distortion conferring to the locality values. Consequently, the attacker will realize it is difficult to expect the attributes that are used to generate the fake tuples. 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.

IV. Conclusion

In this paper, we discussed the problem of content integrity checking for ubiquities database accessing and recommended a new fragile watermarking scheme to handle this problem based on intelligent fake tuples formation in a separate file (distortion free) not inside the database with different kinds of attributes. GA as a meta-heuristic algorithm is utilized to find the optimal numerical attributes values for the created fake tuples, whereas the most frequent approach is used to select the non-numerical values. The suggested system is partition -oriented to embed and verify in each group independently. The watermarking procedure has an improvement over hash function, as it does not be subject to the arrangement of the tuples. The suggested system can be adjusted in keeping with different security levels, by recurring the partitioning across the use of several attributes. Neither watermark generation nor detection is contingent on any relationship or costly arrangement of data items. Each tuple in a table is independently processed; therefore, the scheme is particularly efficient for tuple oriented database operations. Furthermore, 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 focused in the direction of boosting the level of attack resilience against numerous resources of attacks in the watermarking method. Lastly, different applications for our proposed technique could be intended and applied.

References

[1] Y. Li, H. Guo, and S. Jajodia, "Tamper Detection and Localization for Categorical Data using Fragile Watermarks", ACM workshop on Digital Rights Management, pp. 73-82, 2004.

[2] R. Agrawal, P. J. Haas, and J. Kiernan, "Watermarking Relational Data: Framework, Algorithms and Analysis", Very Large Data Bases Journal, 12(2):157-169, 2003.

[3] R. Agrawal, and J. Kiernan, "Watermarking Relational Databases", Very Large Database Conference, pp. 155-166, 2002.

[4] B. Mehta, and U. Rao, "A Novel approach as Multi-place Watermarking for Security in Database", International Conference on Security and Management, pp. 703-707, 2011.

[5] Z. Lei, and R. Li, "Research of Applications in Relational Database on Digital Watermarking Technology", International Journal of Engineering Science Invention, 2(9):84-89, 2013.

[6] S. Bilapatte, S. Bhattacharya, and S. Sawarkar, "A Review on Watermarking Relational Databases", International Journal of Applied Engineering, 4(2):89-96, 2014.

[7] M. Rathva, and G. Sahani, "Watermarking Relational Databases", International Journal of Computer Science, Engineering and Applications, 3(1):71-79, 2013.

[8] P. Singh, and R. Chadha, "A Survey of Digital Watermarking Techniques Applications and Attacks", International Journal of Engineering and Innovative Technology, 2(9):165-175, 2013.

[9] S. Bhattacharya, and A. Cortesi, "A Distortion Free Watermark Framework for Relational Databases", International Conference on Software and Data Technologies, pp. 229-234, 2009.

[10] S. Bhattacharya, and A. Cortesi, "A Generic Distortion Free Watermarking Technique for Relational Databases", International Conference on Information Systems Security, pp. 252-264, 2009.

[11] S. Bhattacharya, and A. Cortesi, "Distortion-Free Authentication Watermarking", International Conference of Software and Data Technologies, pp. 205–219, 2010.

[12] M. Li, and W. Zhao, "An Asymmetric Watermarking Scheme for Relational Database", International Conference on Communication Software and Networks, pp. 180–184, 2011.

[13] R. Arun, K. Praveen, D. Bose, and H. Nath, "A Distortion Free Relational Database Watermarking using Patch Work Method", International Conference on Information Systems Design and Intelligent Applications, pp. 531-538, , 2012.

[14] V. Pournaghshband, "A New Watermarking Approach for Relational Data", Annual Southeast Regional Conference, pp. 127-131, 2008.

[15] N. Zawawi, R. El-Gohary, M. Hamdy, and M. Tolba, "A Novel Watermarking Approach for Data Integrity and Non-repudiation in Rational Databases", International Conference on Advanced Machine Learning Technologies and Applications, pp.532-542, 2012.

[16] H. El-Bakry, and N. Mastorakis, "A New Watermark Approach for Protection of Databases", Proceedings of the International Conference on Applied Informatics and Communications, pp. 243-248, Russia, 2009.

[17] H. El-Bakry, and M. Hamada, "A Novel Watermark Technique for Relational Databases", International Conference on Artificial Intelligence and Computational Intelligence, pp. 226-232, 2010.

[18] H. El-Bakry, and M. Hamada, "A Developed Watermark Technique for Distributed Database Security", International Conference on Computational Intelligence in Security for Information Systems, pp. 173-180, 2010.

[19] G. Gamal, M. Rashad, and M. Mohamed, "A Simple Watermark Technique for Relational Database", International Journal of Intelligent Computing and Information Science, 8(1):92-101, 2008.

[20] S. Bhattacharya and A. Cortesi, "Database Authentication by Distortion Free Watermarking", International Conference on Software and Data Technologies, pp. 219-226, 2010.

[21] I. Kamel, "A schema for Protecting the Integrity of Databases", Computers and Security, 28(7):698-709, 2009.

[22] H. Wu, F. Hsu, and H. Chen, "Tamper Detection of Relational Database Based on SVR Predictive Difference", International Conference on Intelligent Systems Design and Applications, pp. 403-408, 2008.

[23] A. Hamadou, X. Sun, L. Gao, and S. A. Shah, "A Fragile Zero-Watermarking Technique for Authentication of Relational Databases", International Journal of Digital Content Technology and its Applications, 5(5):189 – 200, 2011.

[24] L. Camara, J. Li, R. Li, and W. Xie, "Distortion-Free Watermarking Approach for Relational Database Integrity Checking", Mathematical Problems in Engineering, 2014(1):1–10, 2014.

[25] S. Iqbal, A. Rauf, H. Javed, and S. Ahmad, "Distortion Free Algorithm to Handle Secondary Watermark Attack in Relational Databases", European Conference on Information Management, pp. 214-221, 2011.

[26] A. Mayekar, M. Jha, S. Mule, and C. Shridattopasak, "Relational Database Watermarking", International Journal of Engineering Research and Technology, 2(7):248-252, 2014.

[27] 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, 25(12):2694 – 2707, 2013.

[28] R. Radha, K. Sankari, and S. Devi, "Implementation of Invisible Watermarking Technique in Relational Database", International Journal of Recent Scientific Research, 7(4):10034-10037, 2016.

[29] S. Sonupriya, and R. Rani, "The Digital Watermarking Technique for Numerical Relational Databases", International Journal of Innovations & Advancement in Computer Science, 3(9):14-21, 2014.

[30] S. Panimalar, and D. Srinath, "Reversible Watermarking Technique based on Time stamping in Relational Data", International Journal of Innovations & Advancement in Computer Science, 2(1):961-967, 2015.

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

[32] V. Khanduja, O. Verma, S. Chakraverty, "Watermarking Relational Databases using Bacterial Foraging Algorithm", International Journal of Multimedia Tools and Applications, 74(3):813-839, 2015.

[33] S. Melkundi, C. Chandankhede, "A Robust Technique for Relational Database Watermarking and Verification", International Conference on Communication, Information & Computing Technology, pp. 1-7, 2015.

[34] L. Camara, J. Li, R. Li, F. Kagorora, and D. Hanyurwimfura, "Block-Based Scheme for Database Integrity Verification", International Journal of Security and its Applications, 8(6):25-40, 2014.

[35] V. Khanduja, S. Chakraverty, and O. Verma, "Watermarking Categorical Data: Algorithm and Robustness Analysis", Defense Science Journal, 65(3):226-232, 2015.

[36] X. Cui, X., G. Sheng, and J. Zheng, "A Robust Algorithm for Watermark Numeric Relational Databases", Lecture Notes in Control and Information Sciences, 344(1):810-815, 2006.

[37] K. Jawad, and A. Khan, “Genetic Algorithm and Difference Expansion based Reversible Watermarking for Relational Databases”, Journal of Systems and Software, 86(11): 2742-2753, 2013.