Review of Watermarking relational database Using Optimization Techniques

Monika Gharu*, Deepak Sharma*

[*] *Kurukshetra Institute of Technology & Management, Kurukshetra-136119, Haryana*

Abstract— **Watermark describes information that can be used to prove the ownership of data such as the owner, origin or recipient of the content. Watermarking is the piece of data securely embedded and this is said to be imperceptible. We are discussing a proof of concept of watermarking technique and different types of optimization techniques like GA, BFA & PSO. This technique is resilient to tuple deletion, alteration, and insertion attacks. It includes the feature of detectability, i.e, the original data can be retrieved only with the help of the key value. The watermark data is robustly embedded into the original content .It means that the original values will be retained even if any alterations occurred. We can say it as a blind system, because the knowledge of the original content is no need for the process of watermarking. Furthermore, the watermark detection is blinded, that is, it neither requires the knowledge of the original data nor the watermark.**

Keywords— **Put your keywords here, keywords are separated by comma.**

I. Introduction

Proving ownership rights on outsourced relational databases is a crucial issue in today's internet-based application environments and in many content distribution applications. In this paper, a general approach is proposed for proof of ownership based on the secure embedding of a robust imperceptible watermark in relational data. The steps of the proposed mechanism of watermarking relational database mainly involve decoding and encoding on numerical attribute of relational database. The first phase is to partition the original data and assign partition number to each and every tuple of the relation using Cryptographic Hashing Function (MD5). In the second phase , while changing the data , select the desired watermark and bit bi is selected from the partitioned data and then that bit bi is changed using watermark W. When the original value of data gets changed dueto the watermark bit, it always checks the data usability constraints. In the. third phase, after inserting the watermark in the partition, merge all partitions and get the complete watermarked data. While decoding, use majority voting algorithm to get the correct watermark. Watermarking is the piece of data securely embedded and this is said to be imperceptible. Imperceptible embedding means that the presence of the watermark is unnoticeable in the data

2.RELATED WORK, MOTIVATION AND PRELIMINARY BACKGROUND

R. Agrawal et.al. [1] identified the need of protecting rights over relational databases by digital watermarking. They proposed that relational database can be watermarked in some algorithmically selected attributes out of several candidates attributes in a tuple. This algorithm embeds watermarks in LSBs of a selected attribute, such that changes in these values will not affect their applicability. The watermarking bits are generated using pseudorandom generators and secret key. The technique does not provide mechanism for multibit watermarks. These bits are then embedded into specific bit locations determined under the control of a secret key known only to the owner of the data. The LSB based data hiding technique is not resilient as simple shifting of the least significant bit by one position leads to a significant loss of watermark without much loss to the database hence are prone to bit based attacks.

V. Khanduja and O. P. Verma proposed a more imperceptible embedding mechanism that securely and randomly selects multiple attributes out of the selected candidate attributes for inserting watermarks in varying number of least significant bits. The technique resolves the two important concerns namely: owner identification and proof of ownership. This algorithm embeds the identity of a work's copyright holder as a watermark and this watermark can be used to provide evidence in ownership disputes making it useful in applications where proof of ownership is required. However, the resilience of this technique is also affected by changing LSB. Several image-based watermarking mechanisms [3, 5, 17, 19, 20, 21] have been proposed utilizing various types of attributes. Zhi-Hao Zhang et.al. proposed a novel image-based watermarking method for numerical data. In their method, an image is embedded into relational data which represents copyright information. In this technique the Relational database is divided into various chunks of uniform size. The pixel values of the image were embedded into corresponding locations of attributes by lowering the planer image dimension. Due to the fact that the pixels of image have relative position, and they are sure to be placed in order, this method is vulnerable to conflicting order of embedded marks (pixels) by some subset attacks [19]. J. Sun et.al. used two identification images that are embedded into numeric attribute of relational data in least significant bits of a selected attributes. This technique is also not resilient to

LSB attacks. Ali Al-Haj and Ashraf Odeh [3] proposed similar technique by inserting a binary image watermark in the non-numeric attribute of database tuples. The embedding process of each short string of the binary image is based on creating double-space at a location determined by the decimal equivalent of the short string. However, if kerckhoffs principal for public-system is followed i.e. embedding algorithm is publically known, then it is easy to detect the places where watermarks are embedded i.e. where double-spaces are inserted and can be easily removed.

S. Bhattacharya and A. Cortesi [4] build watermark after partitioning tuples as a permutation of tuples. A hash function is built on the top of this grouping. As the ordering of tuples does not affect the original database, this technique is distortion free. C. Jiang et.al. [12] proposed a watermarking algorithm, which can embed the watermark into a relational database transformed in the DWT domain. They provide an analysis of the wavelet's high frequency coefficients, defined an intensive factor and employed the linear correlation detecting method. The watermark can be distributed to different parts of the relational database. H. Cui et.al. [7] proposed a public key cryptography based algorithm. In this algorithm, asymmetric keys are used in inserting and detecting database watermarks in LSB. Private keys are decided by users and public key by trusted center IPR. Users can not destroy the database watermark through public key. Watermark detection can be completed by the third party using public key without secret leaking. D. Hanyurwimfura et.al. [11] embed watermarks in non-numeric multi words data based on lavenshtein distance. A mark is embedded in the selected attribute of selected tuples by horizontally shifting the location of a word depending on watermark bit. The lavenshtein distance between two successive words within an attribute decides the location where the mark is to be inserted. Certain cloud based techniques [14, 30] were proposed in literature. Based on the concept of similar clouds and N-D normal compatibility cloud generators, the numeric attributes in relational database whose schema is persistent is watermarked. R. Sion et.al. proposed technique for watermarking numeric attribute that selects subsets of the relational database and for each subset; a watermark bit is embedded under data usability bounds. The technique stores the marker tuples in order to accurately recover the partitions. This technique violates the principle of blind watermark detection. Most of the techniques discussed above are based on the use of special marker tuples, which makes them vulnerable to watermark synchronization errors resulting from tuple deletion and tuple insertion. Thus, such techniques are not resilient to deletion and insertion attacks. M. Shehab et.al. formulated watermarking of relational databases as a constrained optimization problem and discussed techniques based on Pattern Search and Genetic Algorithm to solve the optimization problem. This makes it resilient to watermark synchronization errors because it uses a partitioning approach that does not require marker tuples. However the technique is primary key dependent, not resilient to linear transformation attack and is not computationally efficient. One of the recent works carried out by Farfoura et.al. [8] proposes reversible technique for watermarking databases. The proposed technique is primary key dependent and not resilient to linear transformation attack. Moreover, technique selects certain tuples into which watermark is to be embedded, thus if those tuples are altered or deleted, the watermark will be lost. Hence such techniques are prone to subset alteration and subset deletion attacks. This technique is also prone to attribute re-order attack.

In order to increase efficiency of the watermarking system, we have different optimization technique .In this work we have designed a novel, improved watermarking relational database system with enhanced robustness, efficiency and imperceptibility using optimization algorithms. Moreover, improved hash partitioning approach independent of primary key is proposed in this work. Watermark to be embedded is cryptographically made secure and then embedded into multiple attributes within a tuple of a partition. Further, various parameters are tuned to make the watermarking system computationally efficient. To make this paper self explanatory, we briefly explain the bacterial foraging algorithm in the following sub-section.

4. CONCEPT OF OPTIMIZATION ALGORITHM

We have different optimization techniques like GA, BFA and PSO. During foraging of the real bacteria, locomotion is achieved by a set of tensile flagella. Flagella help an E.coli bacterium to tumble or swim, which are two basic operations performed by a bacterium at the time of foraging. When they rotate the flagella in the clockwise direction, each flagellum pulls on the cell. That results in the moving of flagella independently and finally the bacterium tumbles with lesser number of tumbling whereas in a harmful place it tumbles frequently to find a nutrient gradient. Moving the flagella in the counterclockwise direction helps the bacterium to swim at a very fast rate. In the above-mentioned algorithm the bacteria undergoes chemotaxis, where they like to move towards a nutrient gradient and avoid noxious environment. Generally the bacteria move for a longer distance in a friendly environment. When they get food in sufficient, they are increased in length and in presence of suitable temperature they break in the middle to from an exact replica of itself. This phenomenon inspired Passino to introduce an event of reproduction in Bacteria Foraging Optimization algorithm. Due to the occurrence of sudden environmental changes or attack, the chemotactic progress may be destroyed and a group of bacteria may move to some other places or some other may be introduced in the swarm of concern. This constitutes the event of elimination dispersal in the real bacterial population, where all the bacteria in a region are killed or a group is dispersed into a new part of the environment. The original Bacterial Foraging Optimization system consists of three principal mechanisms, namely, chemo taxis, reproduction, and elimination-dispersal. These are described as follows [18].

1. Chemotaxis

2. Reproduction
3. Elimination and Dispersal
4. Optimization Algorithm

4. PROPOSED WATERMARKING TECHNIQUE

In general, any watermarking database algorithm can be represented as shown in Fig. 1. While designing the watermarking database system the major concern is not to avoid data alteration, but to limit the change within usability limits i.e. to acceptable levels with respect to the intended use of the data [20]. Watermark insertion is performed by watermark encoder system and detection of embedded watermarks is performed by watermark decoder. The proposed watermarking system consists of two subsystems:

1. Watermark Encoder
2. Watermark Decoder

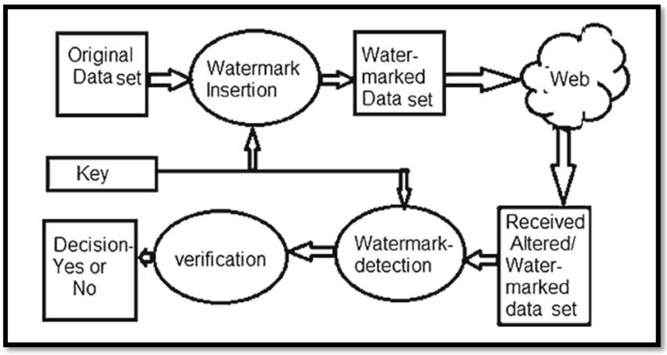

Fig 1: Block diagram of digital watermarking scheme

4.1 Watermark encoder

The watermark encoder embeds desired watermarks into a given relational database. The relational database S is a database relation with scheme S (A0, . . . ANa -1), where Na is the total number of attributes in the relation S out of which, ν attributes are candidates for watermarking such that ν < Na and Nt is the number of tuples in relation S such that Nt = |S|. The target attributes are selected by the owner of the database in a manner such that these attributes can tolerate a small amount of error without affecting the usage of the database. The task is achieved using four steps as shown in Fig. 2

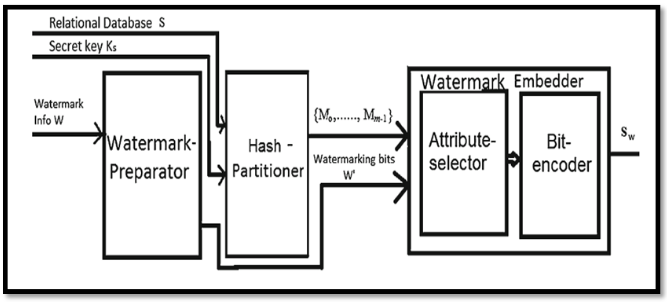

Fig. 2 Watermark encoder

4.2 Hash-Partitioner

An improved hash-partitioning technique is used to partition the database S into m non overlapping partitions M = {M0,.....,Mm-1} in-order to secure the ordering of tuples. This strategy designates one or more attributes from the database S as the partitioning attribute. A hash function is chosen in the range {0,1,......,m-1}. Each tuple on the original relation is hashed on partitioning attributes [20]. If the hash function returns i, then the tuple is placed on partition Mi. The secret key Ks which is selected by the owner of the database, is concatenated with Most Significant Bits (MSB) of thenormalised partitioning attributes in order to make tuple-to-partition assignment more secure. Since owner knows Ks, he can reproduce result but Mallory, an attacker does not know Ks, can't. This makes it more difficult for an attacker to predict the assignment. Such one-way hash function with secret key is referred as Message Authentication Code (MAC) [20]. Figure 4 shows the flowchart for the function partitions that implements the Hash-Partitioner.

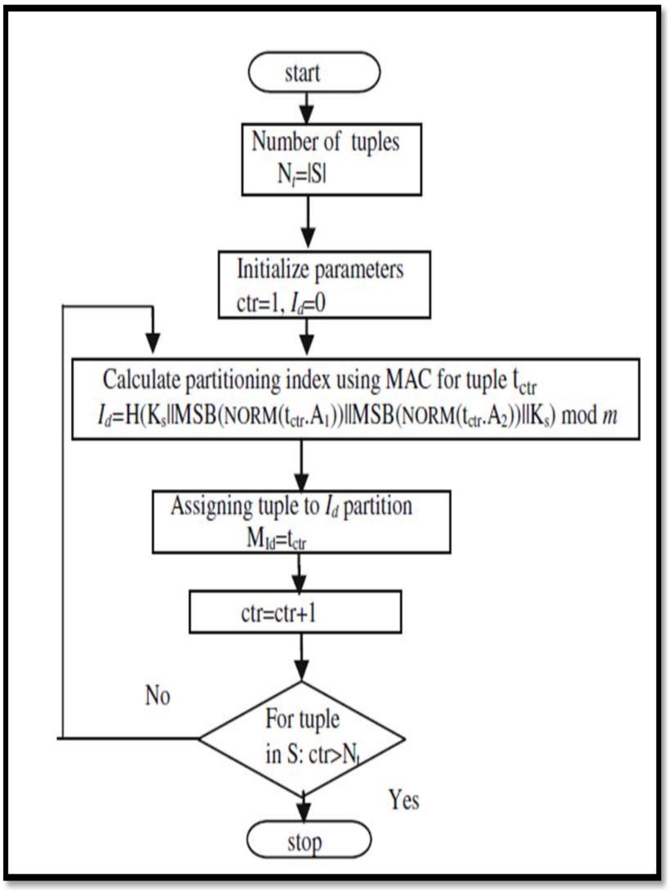
start
Number of tuples N_t=|S|
Initialize parameters ctr=1, I_d=0
Calculate partitioning index using MAC for tuple t_{ctr}
I_d=H(K_s||MSB(NORM(t_{ctr}.A_1))||MSB(NORM(t_{ctr}.A_2))||K_s) mod m
Assigning tuple to I_d partition M_{Id}=t_{ctr}
ctr=ctr+1
No
For tuple in S: ctr>N_t
Yes
stop

Fig 3: Flow Chart Of Partitioning

5.EXISTING SYSTEM

In the already existing technique, the original data is encrypted with the public and private keys. RSA is an algorithm used for encryption. Encryption is the process of converting the original content into cipher text to make it unreadable to anyone except those possessing special knowledge, usually referred to as a key. Public key and private key are used for encryption. Public key is a key which is used for encryption and is known for both the authors and the users where the private or secret key is an encryption/decryption key known only to the party or parties that exchange secret messages. The use of combined public and private keys is known as asymmetric encryption. It is not much secure, because the original data can be easily decrypted with the simple manipulations. So that we can retrieve the original content easily. This is the main drawback in the existing system.

CONCLUSIONS

Watermarking algorithms are often used in larger system designed to achieve certain goals e.g., prevention of illegal copying. Watermarking database can be used to prevent database piracy, where somebody takes somebody else's database, slaps their name on it and then goes into competition with the original database producer. Protection from the piracy of digital assets is usually based upon the embedding of digital watermarks into the data. Watermarking approaches do not prevent copy rather it deter illegal copying by providing a means of establishing the original owners a redistributed copy. In this paper, the new watermarking technique is proposed for relational data that embeds watermark bits in the data by maintaining its meaning and value as it uses usability matrix while changing its original value. The data partitioning is done by using cryptographic hash function (MD5). The proposed technique handles the alteration, deletion and insertion attack more effectively. The watermark resilience was improved by the repeated embedding of the watermark and using majority voting technique in the watermark decoding phase. Moreover, the watermark algorithm can be applied for more than one attribute of the same table.

REFERENCES

[1] Agrawal R, Haas PJ, Kiernan J (2003) Watermarking relational data: framework, algorithms and analysis. VLDB J 12(2):157–169

[2] Ali YH, Mahdi BH (2011) Watermarking for relational database by using threshold generator. Eng Tech J 29(1):33–43

[3] Ali A-H, Odeh A (2008) Robust and blind watermarking of relational database systems. J Comput Sci 4(12):1024–1029

[4] Bhattacharya S, Cortesi A (2009) A distortion free watermark framework for relational databases. In: Shishkov B, Cordeiro J, Ranchordas A (eds) ICSOFT 2009: Proceedings of the 4th international conference on software and data technologies, vol 2, Sofia, Bulgaria, INSTICC Press, pp 229–234

[5] Chen X, Chen P, He Y, Li L (2008) A self-resilience digital image watermark based on relational database. Int Symp Knowl Acquis Model 698–702

[6] Cox IJ, Miller ML, Bloom JA (2002) Digital watermarking. Morgan Kaufmann, Academic Press

[7]Cui H, Cui X, Meng M (2008) A public key cryptography based algorithm for watermarking relational databases. Proc IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Harbin, 1344–1347. doi:10.1109/IIH-MSP.2008.194

[8] Farfoura ME, Shi-Jinn H, Jui-Lin L, Ray-Shine R, Rong-Jian C, KhanMK (2012) A blind reversible method for watermarking relational databases based on a time-stamping protocol. Expert Syst Appl 39(3):3185–3196

[9.] Gupta VK (1997) Copyright issues relating to database use. DESlDOC Bull Inf Technol 17(4):11–16

[10.] Hanmandlu M, Verma OP, Kumar NK, Kulkarni M (2009) A novel optimal fuzzy system for color image enhancement using bacterial foraging. IEEE Trans Instrum Meas 58(8):2867–2879

[11] Hanyurwimfura D, Liu Y, Liu Z (2010) Text format based relational database watermarking for nonnumeric data. Proc IEEE International Conference on Computer Design and Applications (ICCDA), vol 4. Qinhuangdao, 312–316. doi:10.1109/ICCDA.2010.5541119

[12]. Min H, Jia-heng C, Zhi-yong P, Cheng Z (2004) A new mechanism based on similar clouds watermark for database's information security. Wuhan University. J Nat Sci 9(4):415–419. doi:10.1007/BF02830434

[13] Mishra S (2005) A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation. IEEE Trans Evol Comput 9(1):61–73

[14] National geochemical survey database of the US, http://tin.er.usgs.gov/geochem/

[15] Odeh A, Al-Haj Ali (2008) Watermarking relational database systems. First International Conference on the Applications of Digital Information andWeb Technologies, Ostrava, 270–274. doi:10.1109/ICADIWT.2008.4664357

[16] Passino KM (2002) Biomimmicry of bacterial foraging for distributed optimization and control. IEEE Control Syst Mag 22(3):52–67

[17]Sardroudi HM, Ibrahim S (2010) A new approach for relational database watermarking using image. 5th International Conference on Computer Sciences and Convergence Information Technology (ICCIT), Seoul, 606–610. doi:10.1109/ICCIT.2010.5711126

[18] Schneier B (2008) Applied cryptography, protocols, algorithms and source code in C. Wiley-India

[19] Shehab M, Bertino E, Ghafoor A (2008) Watermarking relational databases using optimization-based techniques. IEEE Trans Knowl Data Eng 20(1):116–129

[20] Silberschatz A, Korth HF, Sudarshan S (2005) Database system concepts. McGraw-Hill Int. Edition

[21] Sion R, Atallah M, Prabhakar S (2004) Rights protection for relational data. IEEE Trans Knowl Data Eng
16(12):1509–1525