Volume 3, Issue 12, December 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Modern Watermarking Approach for Non- Numeric Relational Database

Bhupendra Nath Chaudhary, Awadesh Kumar Sharma

CSE & MMMEC Gorakhpur,
U.P. (India)

***Abstract -** In this paper, a modern watermarking approach is proposed for data authentication and integrity of Relational Database. For integrity verification of tables in the database, the watermark has to depend on a secret key and on the original copy of that table. It is important that the dependence on the key should be sensitive. The proposed approach makes use of the concept of Eigen values by constructing a Tuple -Relation Matrix for each tuple. The Eigen values are used for generating the watermark for a record in the table. Watermark embedding is done by using Eigen values in a non numeric attribute of a tuple. Detection of the watermark prove the authenticate and integrity of data. We will show that our approach leads to an effective way that is robust against different forms of malicious attacks as well as benign updates to the data. All the tuples in relational databases are first divided into different subsets and then the tuples in each subset, are sorted secretly. In each subset, there are attribute watermarks and tuple watermarks generated dynamically by using one way Hash Function and relational data for watermark security and blind extraction. Subset watermarking grid made up of attribute watermarks and tuple watermarks are construted for localizing modifications in a subset of relational databases. Theoritical analysis and experimental results show that the proposed method can detect tuple insertion, attribute value modification, tuple deletion and attribute deletion for providing authenticity verification of relational data.*

***Keywords-** Relational Database, Attribute Watermarks, Tuple Watermarks, Eigen values, Watermarking Algorithm.*

I. INTRODUCTION

Digital images, video and audio are examples of digital assets which have become easily accessible by ordinary people around the world. However, the owners of such digital assets have long been concerned with the copyright of their digital products, since copying and distributing digital assets across the Internet was never easier and possible as its now a days. Digital watermarking technology was suggested lately as an effective solution for protecting the copyright of digital assets[2,3]. This technology provides ownership verification of a digital product by inserting imperceptive information into the digital product. Such 'right witness' information is called the watermark and it is inserted in such a way that the usefulness of the product remains, in addition to providing it with robustness against attempts to remove the watermark

Here we briefly discuss three of the previous approaches related to our work for watermarking relational databases. First, the method given in Agrawal et al[1] utilizes the pseudorandom number generator algorithm to identify the marked tuples and Attributes, and also the degree of error to the marked attributes. Second, is the approach proposed by Zhang et al using embedded images [4]. In other words, in their approach, they embed images into relational database as the watermarks.

While previous techniques have been mainly concerned with introducing errors into the actual data. The approach proposed by T. Rethika , Ivy Prathap, R. Anitha and S.V. Raghavan these authors contributes a novel secure and efficient algorithm using the mathematical concept of Eigen values for text watermarking. This concept motivated use to create tuple Relation matrix [6]. Other approach proposed by Vahab Pournaghshband[5] inserts new tuples that are not real and they call them "fake" tuples, to the relation as watermarks. This approach uses fake tuples and utilizes the insertion and detection watermarking algorithms. Evaluating watermarks for relational database is a challenge and requires further consideration. However, the persistency of the watermark after both malicious and benign updates, as a sub problem, might be evaluated by acquiring access to a log of user queries on a particular database over a reasonably long period of time, and then run the log on the watermarked database and observe whether the watermark detection algorithm will confirm.

II. RELATED WORK

In the last few years some watermarking scheme have been proposed to protect relational database. Agarwal and kierman originally present a robust watermarking scheme based on bit-level for databases. Li further extend this scheme to embed multiple watermarking bits. Sion put forward a statistical- property watermarking algorithm. For relational data Shehab propose a resilient watermarking method based on genetic algorithm and pattern search technique. These robust watermarking schemes for relational databases are designed for copyright protection. However there are few fragile watermarking schemes geared for relational data integrity. Devanbu and pang present watermarking schemes based on a merkle hash tree for databases. Though these schemes can detect modifications, they can not localize the modification.

Guo propose a fragile watermarking scheme which can detect, localize and characterize three kinds of malicious modification (Inserting tuples, deleting tuples and modifying values) made to relation data. However the method of constructing watermarking is very complicated. Meanwhile it's simulation results don't indicate that it can detect, localize and characterize attribute variation. Our proposed approach overcomes these defects.

III. OUR APPROACH

Process of Generating Watermark:

The Watermark generation process involves secret key generation using Eigen values of Tuple-Relation matrix for a tuple in the given relation. We use the Employee's personal Database as an example.

Secret Key Generation:

Consider the Employee database, select low impact non numeric attributes such as address and city to be watermarked. Then compute the no. of vowels, consonants and special characters occurring in each tuple for selected non numeric low impact attributes of the relation. As per the notation defined in figure 1, the weighted consonant sum C and weighted vowel sum V of high impact non numeric attribute of a tuple is calculated. Now the weighted ASCII sum A of each tuple is computed as below,

$$\mathbf{A} = \frac{\sum_{i€n} \mathbf{ASCII(c)} \div \mathbf{n}}{\mathbf{K}} \qquad \mathbf{0< i <n} \qquad \mathbf{(1)}$$

where ASCII(c) is ASCII value of the character c in the tuple. Tuple vectors are constructed with V, C, P, A as its components.

k	Number of tuples in the relation
n	Number of non numeric low impact attributes in the relation
V	ASCII values of the vowels of selected non numeric attribute are summed up to give the weighted vowel sum V
C	ASCII values of the consonant of selected non numeric attribute are summed up to give the weighted consonant sum C
P	the count of special characters of selected non numeric attribute
A	The weighted ASCII sum of the all character of selected non numeric attribute is calculated using formula (1) to give the weighted sum A
X	Concatenate all the Eigen value
m	Secret key

The Tuple-Relation matrix D is

$$\mathbf{D = [\,d\,i\,j\,]_{n\times 4}} \qquad \mathbf{(2)}$$

Where di1, di2 and di4 denote the weighted vowel sum, consonant sum, and ASCII sum of selected non numeric attribute in the given relation respectively and di3 denote the number of special characters of selected non numeric attribute in the given relation respectively. Each vector in the tuple-Relation matrix is not a unit vector and Hence it is normalized as below,

$$\mathbf{N = [n\,i\,j\,]_{n\times 4}} \qquad \mathbf{(3)}$$

$$\mathbf{n\,i\,j} = \frac{\mathbf{d\,i\,j}}{\mathbf{(\,\Sigma d2\,i\,j\,)1/2}}$$

The normalized tuple-Relation matrix N is then pre-multiplied with its transpose NT to yield the watermark matrix W which is a square matrix of order 4. Let e1, e2, e3, e4 be the 4 Eigen values of the watermark matrix W. Some of the Eigen values may be zero when the rank of the matrix is less than or equal to 4. The precision of the Eigen values is increased by multiplying each of the Eigen values by 10. The generated secret key is easily computable once we arrive at the tuple-Relation matrix. In O(n) time, the secret key can be generated from the relation. On the other hand it is not easy to form the Tuple-Relation matrix even if the secret key is known. It is hard to find the tuple-Relation matrix even from the Eigen values.

Key Generation Algorithm:

A. For each tuple r € R do.
B. Get number of non numeric low impact attributes in n.
C. Compute the count of special characters P for selected non numeric attribute in the tuple. Also calculate the weighted vowel sum V, consonant sum C and weighted ASCII sum A for selected non numeric attributes of tuple in the relation.
D. Construct the tuple-Relation matrix D for the selected tuple in given relation.
E. Normalize the matrix to get N.
F. Compute the watermark matrix W=NT*N, where NT denotes the transpose of N.
G. Find the Eigen values of the watermark matrix, W. If they are floating points, convert into integers by multiplying by 10 to get two digit eigen value .

H. Concatenate all the Eigen value in X.
I. If $X < 8$ digits then padding of zeroes to right
else
Take first 8 digit from right
J. Secret key m = X

This algorithm produces a unique secret key of a tuple in given relation.

Contribution by this work:

In this work we have handle the non numeric data by numeric watermarking technique. In which creation of secret key generation by Eigen values using low impact non numeric attributes such as address and city to be watermarked. Then compute the no. of vowels, consonants and special characters occurring in each tuple for selected non numeric low impact attributes of the relation.

Outline of documentation:

This dissertation is devised into eight section. Section-1 is introductory section about the basics of our research. Section-2 describes related work. Section -3 describes the our approach. Section-4 describes watermarking algorithm. Section-5 discusses experiments and results of relational database watermarking. Section-6 the algorithm analysis**.** Section-7 the application. Section -8 the conclusion and future scope.

IV. WATERMARKING ALGORITHM

To avoid the attention of attackers an effective solution is to merge watermarking into relational data. This technique marks only numeric attributes and assumes that the marked attributes are such that small changes in values are accepted and non-obvious. For protecting each data from modifying all numeric attributes of relational need to marked. We are watermarking a relational database R whose scheme is R(P, A_1, A_2,…..A_S) where r.P is the primary key attribute denoted by $group_i$ r_xA_y the value A_y in the tuple r_x €$group_i$ €R. Where $group_i$ is referred to the ith subset. For simplicity, assume that attributes A_1, A_2……..A_S are selected to embed watermarking bit and each attribute value can tolerate modification of at least two least significant bits. The parameter g is a control parameter that presents the number of subsets of tuples. We denote by g_i the actual number of tuples in a subset. The complete algorithm consists of three modules: watermarking constructing, watermarking embedding and watermarking detection.

A. Watermarking Constructing:

The watermarking constructing procedure consists of the following operational steps:

- The complete set of tuples making up the database are partitioned into a number of unique nonintersecting subsets of tuples according to the number of subset g, the hash value of a watermarking key k and their primary key. Only the database owner has the knowledge of g and k for security reason.
- The tuples in each subsets are the sorted on the basis of their corresponding primery key.
- Construct attribute watermarks: Attribute watermarks are extracted from the hash value generated according to the watermarking key k and all values of the same attributes of all tuples in each subset.
- Construct tuple watermarks: Tuple watermarks are extracted from the hash value generated according to the watermarking key k and all attribute values of the same tuple in each subset.

B. Watermarking Embedding:

The watermarking procedure is very simple. Attribute watermarks and tuple watermarks are embedded into each other subset independently. The embedding procedure includes the following operational steps:

- In attribute watermarks embedding for any value r_xA_Y in $group_{i,}$ the least significant bit is set to the most significant bit of the corresponding attribute watermarks.
- In tuple watermarks embedding for any values r_xA_y in $group_i$, the next least significant bit is set to the significant bit of the corresponding tuple watermarks.
- Repeat the above operations for each subset of the database under watermarking until all watermarks are embedded.

In this embedding procedure, the column of the tuple be marked by attribute watermarks, while the row of tuple be marked by tuple marked. In this way, the watermark grid made up of attribute watermarks and tuple watermarks are useful to describe malicious attacks.

C. Watermarking Detection:

Watermark detection is verified based on each subset the same as in watermarking embedding. For this reason we need to the embedding key k and the number of subsets g. The detection procedure mainly consists of the following operational steps:

- Construct attribute watermarks WA and tuple watermarks WT: Construction method of watermarking is the same as used by watermarking embedding.
- Extrat attribute watermarks WBA and tuple watermarks WBT: Attribute watermarks are extracted from the least significant bit of any value r_xA_y, while tuple watermarks are extracted from the next least significant bit of any value r_xA_y.

Through comparing WA and WBA, WT and WBT respectively, the verifying result is given.

V. EXPERIMENTS AND RESULTS

The experiments were performed on a computer with the Windows XP professional 2002 operating system 1.73 GHz intel processor, 1.0G of memory, and 40-GB hard disk. We used the real life forest cover type dataset. Available from http://kdd.ics.uci.edu/databases/covertype/covertype.html. We choose the five integer-valued attributes for watermarking. The subset number is 8 and the number of the least significant bits used to embed watermarking. The watermarking algorithm is implemented with Borland Delphi 7 Enterprise Edition. In the following using a subset of a database as an example, We give the detecting results on four kind of modifications: modify attribute values, insert tuples, delete tuples and delete attribute. A_i(l ≤ i ≤ 5) is the attribute name. In this section, we briefly enumerate some assumptions and notation used.

A. Localization Results of Modifying a Values:

In table 1, r_2A_3 is modified, the attributes and tuple watermarks related to the value also change. As shown in table 3, the verification results FF indicates the location of the modified attribute value. When multiple values are altered the similar verification result can form.

TABLE 1: A Subset of a Database

	A_1	A_2	A_3	A_4	A_5
r1	3255	1687	224	208	902
r2	3205	155	228	230	2089
r3	3134	5623	222	225	843
r4	3353	2016	252	207	811
r5	3099	2295	214	187	2291

TABLE 2: Table of Symbols

Symbols	Meaning
TT	No watermarks change
TF	Only tuple watermarks change
FT	Only attributes watermarks change
FF	Attribute watermarks and tuple watermarks change

TABLE 3: Localization results of Modifying a Value

	A1	A2	A3	A4	A5
r1	TT	TT	FT	TT	TT
r2	TF	TF	**FF**	TF	TF
r3	TT	TT	FT	TT	TT
r4	TT	TT	FT	TT	TT
r5	TT	TT	FT	TT	TT

B. Localization Results of Inserting a Tuples:

A new tuple is inserted, as shown in table 4 (table r4 is inserted) all attribute and the tuple watermarks related to the tuple change. As shown in table 5 the verification result indicates that the fourth tuple in the subset is added. If multiple tuples are added the similar verification results can be seen. Perhaps, the added tuples are nothing but into different subsets.

TABLE 4: Insert a tuple in table 1

	A1	A2	A3	A4	A5
r1	3255	1687	224	208	902
r2	3205	155	228	230	2089
r3	3134	5623	222	225	843
r4	**3200**	**1801**	**255**	**199**	**2345**
r5	3353	2016	255	199	2345
r6	3099	2295	214	187	2291

TABLE 5: Localization results of Inserting a tuple

	A1	A2	A3	A4	A5
r1	FT	FT	FT	FT	FT
r2	FT	FT	FT	FT	FT
r3	FT	FT	FT	FT	FT
r4	**FF**	**FF**	**FF**	**FF**	**FF**
r5	FT	FT	FT	FT	FT
r6	FT	FT	FT	FT	FT

C. Localization Results of Deleting a Tuple:

r_2 is deleted as shown in table 6, all attribute watermarks in the subset change, while all tuple watermarks don't change . The verification results indicate that some tuples in the subset are deleted. Before watermark verification, even if database owners do not know the number of tuples in each subset, they are unable to estimate the number of deleted tuples in a subset. But though the algorithm, tuples deletion may be localized in one or more subsets.

TABLE 6 Localization results of deleting a tuple

	A1	*A2*	*A3*	*A4*	*A5*
r1	FT	FT	FT	FT	FT
r2	FT	FT	FT	FT	FT
r3	FT	FT	FT	FT	FT
r4	FT	FT	FT	FT	FT

D. Localization Results of Deleting Attributes:

Attackers adjust the order of attributes besides the primary key attribute. Our algorithm still can be used to detect the above malicious attacks. Even though the order of attributes changes, all values of the same attributes have no change. In fact, the values of all attributes have no change to the change of attributes of databases owner firstly restore the original order of attributes and then extract watermarks. For example, If the last attribute A_5 of table 1 is deleted the verification results are given in the Table 7. As shown in table 7, the verification results of all attribute watermarks are true, while the verification results of all watermarks are false. The verification indicates each tuple must suffer the same attacks. If multiple attribute are deleted, the similar verification results can be seen figure 1. shows that the detection failure rate is significantly decreased with the number of rest attributes which indicates that less attributes are deleted, the higher the detection success rate is affected.

TABLE 7: Localization results of deleting a tuple

	A1	*A2*	*A3*	*A4*
r1	TF	TF	TF	TF
r2	TF	TF	TF	TF
r3	TF	TF	TF	TF
r4	TF	TF	TF	TF
r5	TF	TF	TF	TF

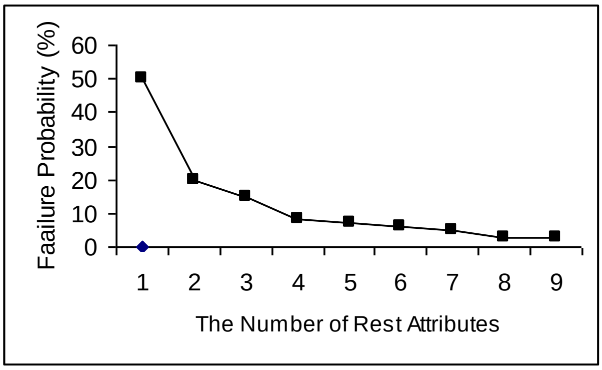

Figure 1: Failure Probabilities for Multiple Attributes Detection

VI. ALGORITHM ANALYSIS

A. Role of a Primary Key:

In the proposed algorithm, the primary key is used for dividing tuples, constructing watermarks and sorting tuples. So it is not allowed to be modified. For relational databases without primary key, we may construct a virtual primary key with reference to the literature[12] and then embed watermarks into databases.

B. Randomness of the Number Tuples in a Subset:

In the proposed approach, the number of tuples in a subset is dynamic and random. For example, there are 40 tuples needed to be divided into 8 different subsets. According to the proposed approach, the tuple divided into different subset is not known, even to the owner of the databases, so no everyone be able to sure the number of tuples in each subset. This promotes the security of watermarking databases.

TABLE 8 Total Error induced by Watermark Embedding

Attribute name	Modification rate of Mean (%)	Modification rate of variance (%)
A_1	0.0072	0.0033
A_2	0.0167	0.0084
A_3	0.0219	0.0806
A_4	0.0194	0.0377
A_5	0.0119	0.0345

C. Error Calculation of Relational Data:

In many cases, for relational data in databases, the difference is very small when the values of the same attribute are compared and even these values are highly similar. Pre and post embedding watermarking the variation range of the value of each attribute has little. Therefore using the mean and variance of relational data to measure total error of each attribute induced by watermarks embedding. As shown in table 8 watermarking databases has little influence on the original values of attributes.

D. Invisibility of Watermark:

In the proposed algorithm watermarking information controlled by private key directly are embedded into numeric present using numeric value, attackers are unaware of the entity of watermarks.

E. Effects of Changing the order of Tuples:

According to our algorithm, the order and division of tuples are uncorrelated, so disordering process to the order of tuple will not affect the extraction and detection of watermarks.

F. Blind Verification of Watermark:

In the proposed algorithm, the original database need not be required for watermark verification. The property can improve practical application of watermarking technique for copyright and integrity authentication of databases.

VII. APPICATION

A. In the semiconductor industry, parametric data on semiconductor parts is provided primarily by three companies: Aspect, IHS, and IC Master. They all employ a large number of people to manually extract part specifications from datasheets.They then license these databases at high prices to design engineers. Companies like Acxiom have compiled large collections of consumer and business data.

B. In military applications where the information of personnel, armaments have to be kept secure, this database can be used.

C. The ACARS meteorological data used in building weather prediction models. The wind vector and temperature accuracies in this data are estimated to be within 1.8 m/s and 0.5° C respectively. The errors introduced by watermarking can easily be constrained to lie within the measurement tolerance in this data.

D. Consider experimentally obtained gene expression datasets that are being analyzed using various data mining techniques. The nature of some of the data sets and the analysis techniques is such that changes in a few data values will not affect the results.

F. The customer segmentation results of a consumer goods company will not be affected if the external provider of the supplementary data adds or subtracts some amount from a few transactions.

VIII. Conclusion and Future Scope

In this study we proposed a watermarking approach based on Eigen values of non numeric attribute and watermarking algorithm based on bit shifting of attributes of subsets of tuples. The effectiveness of the proposed algorithm is verified against four kind of database attacks such that attribute values modifying, tuple inserting, tuple deletion and attribute deletion. A novel secure and efficient algorithm using the mathematical concept of Eigen values for non numeric relational database is proposed. This approach can be used effectively where a huge amount of relational data is transferred between owner and authenticated users. Ongoing and future research is designing a watermarking algorithm for non-numeric relational database using different secret keys.

References

[1] Agrawal, R., Haas, P., and Kiernan, J. 2003. Watermarking relational data: framework, algorithms and analysis. The VLDB Journal 12, 2(Aug. 2003), 157-169. DOI= http://dx.doi.org/10.1007/s00778-003-0097-x.

[2] J.T. Brassil, S. Low, N.F. Maxemchuk , and L. O'Gorman, "Electronic marking and identification techniques to discourage document copying ", IEEE Journal on Selected Areas in Communications", vol. 13, No. 8, October 1995, pp.1495-1504.

[3] Ding Haung, Hong Yan, "Interword distance changes represented by sine waves for watermarking text images", IEEE Transactions on Circuits and Systems for Video Technology, vol. 11, No.12, pp. 1237- 1245, Dec 2001

[4] Zhang, Z., Jin, X., Wang, J., Li, D. 2004. Watermarking Relational Database Using Image. In Proceedings of the Third International Conference on Machine Leaning and Cybernetics, (Shanghai, August 26 – 29, 2004).

[5] Vahab Pournaghshband 2008 A New Watermarking Approach for Relational Data, ACM-SE'08 March 28-29,2008,Auburn,AL,USA,ACM ISBN 978-1-60558-105-7/08/03.

[6] T. Rethika , Ivy Prathap, R. Anitha and S.V. Raghavan 2009 ESRGroups France A Novel Approach to Watermark Text Documents Based on Eigen Values.

[7] L. Boney, A. H. Tewfik, and K. N. Hamdy. Digital watermarks for audio signals. In International Conference on Multimedia Computing and Systems, Hiroshima, Japan, June 1996.

[8] M. Atallah and S. Lonardi, "Authentication of LZ-77 Compressed Data," *Proc. ACM Symp. Applied Computing,* 2003.

[9] M. Atallah, V. Raskin, C. Hempelman, M. Karahan, R. Sion, K. Triezenberg, and U. Topkara, “Natural Language Watermarking and Tamperproofing,” *Proc. Fifth Int'l Information Hiding Workshop,* 2002.

[10] N. F. Johnson, Z. Duric, and S. Jajodia. Information Hiding: Steganography and Watermarking – Attacks and Countermeasures. Kluwer Academic Publishers, 2000.

[11] R. Agrawal and J. Kiernan.Watermarking relational databases. In proceedings of VLDB, 2002.

[12] Li. Y. V. Swarup and S. Jajodia ”Constructing a virtual primary key for fingerprinting relational data.” Proc the 3rd ACM workshop on digital rights management ACM Press. October 2003, pp 133-141. http://portal.acm.org/citation.cfm= 947380.947398.

[13] R. Sion, M. Atallah and S. Prabhaka. “Right protection for relational data.” Proc the ACM International Conference on Management of Data.San Diego, California USA, June 2003 pp 98 109. http://portal.acm.org/citation.cfm=872757872772.

[14] M Shehab, E. Bertino and A. Ghafoor. “Watermarking relational databases using optimization-based techniques.” Proc IEEE Transactions on knowledge and data engineering TKDE 08) IEEE Educational Activities Department Press January 2008 pp. 116-129 10 1109/ TKDE 2007 19068.

[15] P. Devanbu, M. Gertz, C. Martel and S Stubblebine,” Authentic data publication over the internet.” Journal of Computer Security vol.11 iss 3 march 2003, pp 2910314 ISSN 0926-227X.

[16] H. Pang and K. Tan. ” Authenticating query results in edge computing.” Proc the 20th International Conference on Data Engineering. IEEE Computer Society Press. April 2004 pp 560, doi: 10 1109/ICDI: 2004 1320027.

[17] Guo H. Li. Y. Lau A. and S Jajodia “A Fragile watermarking scheme for detecting malicious modification of databases relations.” Information Science vol 176. 2006 pp 1350-1378 doi: 101016 june 2005 06.003