A Unique Approach for Watermarking Non-numeric Relational Database

Rajneesh Kaur Bedi
Assistant Professor,
Comp Department, MITCOE
Pune, India.

Purva Gujarathi
B.E. Student,
Comp Department, MITCOE
Pune, India.

Poonam Gundecha
B.E. Student,
Comp Department, MITCOE
Pune, India.

Ashish Kulkarni
B.E. Student,
Comp Department, MITCOE
Pune, India.

ABSTRACT
In today's transient world, securing digital assets is very important. For data authentication and integrity of the relational database we have proposed a secure method which uses both semantic and syntactic techniques to watermark the tuple in a relation. The Watermarking technique is dependent on secret key and on the relation. The proposed algorithm is based on the concept of predefined signals for ASCII characters. A secret key is generated by using these signals only. To embed a watermark we are using the concept of abbreviations for words and also one syntactic approach, if former technique is not possible. Detection of watermark leads to the authentication and integrity to data. Experimental result shows that, our approach is robust and secure against the various malicious attacks.

General Terms
Security, Algorithms

Keywords
Watermarking, Relational Database, Signals.

1. INTRODUCTION

1.1 Digital Watermarking
The arrival of Internet raised many new opportunities for the creation and delivery of content in digital form. Eventually the threats of tampering, forgery, falsification and piracy of the digital assets such as multimedia, software, and database also increased. This has led a developing interest in copyright protection and authentication of digital assets. Digital Watermarking is one of the solutions against destruction of these digital assets [3], [4], [5]. It provides promising methods and technologies that hide information by inserting a secret key directly into the data.

The key idea behind introducing a watermark is to insert some errors in the original data such that it does not affect usefulness of the data. In addition to it the watermark should be robust enough to survive against all cyber-attacks.

1.2 Digital Watermarking for Relational Data
Initially most of the watermarking work was related to still images, audio, videos and VLSI designs. [1], [2] Nowadays watermarking relational database has become a hotspot due to increasing use of databases in many real-time applications.

A database system can be easily duplicated or exchanged. So there is need to develop a unique watermark technique which is secured and robust to deal with issues like piracies.

In general, the database watermarking techniques comprises of two stages:

i. Watermark Insertion.
ii. Watermark Detection.

During insertion stage, a secret key K is used to insert the watermark W in the database and the key is kept private. To detect the incredulous database, the database is taken as input and the private key K is used for the same. Then this incredulous database is then compared with the original database.

Some efforts are already invested in watermarking relational database [6], [7], [8], [9], [10], [11] .In this paper; we propose a scheme of watermarking non-numeric relational database that is robust against malicious attacks and benign updates of the data.

The rest of the paper is organized as follows. Section II describes previous related work. Section III specifies our proposed system along with watermarking algorithms. Section IV gives results for some conducted experiments. And section V draws some conclusions.

2. RELATED WORK
Now we will discuss about the previous work done related to the field of watermarking relational database. Rakesh Agrawal and Jerry Kiernan [7] proposed a database watermarking scheme for watermarking numeric values of the database. This scheme is based on watermarking LSB of selected attributes of a selected subset of tuples and considerably less robust. R. Sion, M. Atallah and S. Prabhakar [11] also present a different approach

to robust watermarking scheme for databases. In their scheme, all tuples are securely sorted and divided into non-intersecting subsets. A single watermark bit is embedded into tuples of a subset by modifying the distribution of tuples values. But the scheme is not favorable for database which needs frequent updating, because it is very expensive to re-insert watermark into the updated database in this scheme. Other approach proposed by Vahab Pournaghshband [8] 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.

Ding Huang and Hong Yan [4] proposed a scheme where interword spaces of different text lines are slightly modified. After the modification, the average spaces of various lines have the characteristics of a sine wave and the wave constitutes a mark. But watermarking by this approach will increase the memory overhead in relational database.

Mahmoud E. Farfoura,Shi-Jinn Horng, Jui-Lin Lai, Ray-Shine Run, Rong-Jian Chen and Muhammad Khurram Khan[16] proposed a method which embed a watermark bits in the fractional portion of numeric attributes of the database by using a reversible data-embedding technique. But use of numeric attributes for watermarking makes some of the attacks intensive to the system.

There is one more approach proposed by R. Bedi, A. Thengade and V. Wadhai [6]. They proposed a scheme that introduced the concept of Eigen value based watermark generation to watermark non numeric attributes in the relational database.

Their approach proves to be an effective technique that is robust against different forms of malicious attacks as well as benign updates to the data. But the watermark inserted can be easily noticed by malicious users as it is clearly visible in the database.

This leads to a need for generating a feasible algorithm for watermarking relational database which is robust, persistent and most importantly invisible. Here we present our scheme for watermarking relational database.

Table 1. Table of Related Work

Key Points	Our Approach	Previous Approach
Watermarking with single or multiple bits	Multiple Bits	Single Bit[7]
Robustness	More Robust	Comparatively Less Robust[7]
Spread Spectrum	On Multiple Tuple	Only on Single Tuple [11]
Persistent Watermarking	Persistent	Non-Persistent[11]
Watermark Insertion	Using signals	By inserting fake tuples[8] and Eigen values[6]
Invisible Watermarking	Using Abbreviations	Using Spaces[4]
Embedding Watermark Bits	In Non-Numeric Attributes	In Numeric Attributes[12]
Watermarking Based on	Predefined signals	Discrete Wavelet Transform[16]

3. PROPOSED SYSTEM

3.1 Technique for Generation of Secret Key

To Watermark the Relational Database we generally require a secret key. A secret key is generated by using a logic signals for low impact attributes.

Consider the Student's database, select low impact non numeric attributes such as address and Hometown to be watermarked. Compute the set of low impact vowels V, consonants C, special characters P and highest length H_l occurring in each tuple for selected non numeric low impact attributes of the relation.

Select the logic signals for set of all vowels V, for set of all consonants C and for set of all special characters P from the predefined library of signals A. Now combine all signals of set of vowels to form single signal g_v. Similarly combine all signals of set of consonants and special characters to form single signal g_c and g_p respectively.

Table 2. Table of Notations

T	Number of tuples in the relation
N	Number of non-numeric low impact attributes in the relation
V	Set of vowels of selected non numeric attribute
C	Set of all the other alphabets apart from vowels
P	Set of special characters of selected non numeric attribute
H_l	Highest length of a selected non numeric attribute
A	Predefined signals for all the characters
g_v	Combined logic signal of V
g_c	Combined logic signal of C
g_p	Combined logic signal of P
K	Secret key

If the combination of set signals of vowels doesn't produce a common graph g_v, then take the disjoint of signals of set of vowels. Similarly take the disjoint of signals of set of consonant and set of special character to form single signal g_c and g_p respectively.

Now compare these signals with the predefined signals A and evaluate the ASCII values of those characters which match with maximum with these signals. Store these ASCII values in S_1, S_2 and S_3.

1) For each tuple t € R do
2) Select the Low Impact Non-numeric attributes.
3) Compute the set of low impact vowels(V), Consonants(C) and Special characters(P) and H_l from selected non numeric attributes
4) Select a logic signals for set of vowel (V), set of Consonant(C) and set of special character (P) from A.
5) Compute the logic signal g_v, g_c and g_p.
6) Compute corresponding ASCII values of g_v, g_c and g_p and store it in S_1, S_2 and S_3 respectively.
7) Compute packed ASCII value of H_l and store it in S_4.
8) Concatenate S_1, S_2, S_3 and S_4 and assign it to secret key K.
9) If K < 8 digits then padding of zeroes to right

Fig 1: Key Generation Algorithm

As we have evaluated the highest length occurring in a tuple and stored in H_l. Now evaluate packed ASCII value of H_l and store it in S_4.Now concatenate the S_1, S_2, S_3 and S_4 to form secret key K.

Now we will discuss about searching complexity required to match the signals to particular ASCII value. If we use B-tree as a searching procedure then it requires $O(t\log_t n)$ CPU time to search, where t is time taken by while loop within each node and n is the number of keys in the B-tree.

So, secret key generation process is differential and generates a unique secret key.

3.2 Watermarking Insertion

In the above section we have discussed about the generation of secret key. In this section we will provide a technique to watermark a tuple. As we are considering the low impact attributes as our main focus towards watermarking, therefore change in their values should not affect much.

Consider there are n low impact attributes. We have secret key K of m digits, where (m>n). Select the n digits from right of secret key K and store it in W. Select first and last digits of the W. And find its average and store it in W_k. If the value of W_k is less than equal to n then W_k is the attribute which to be watermarked.

If W_k is greater than n then we will change its value by taking its MOD with n and storing the value in W_k itself.

Up till now we got the attribute to be watermarked. We will now use one of two techniques to watermark the attribute. In first one we have a metadata 'm' which stores the list of words and their abbreviated form. For example in address field we can abbreviate 'block' as 'blk.', 'Apartment' as 'apt.' and 'Road' as 'rd.'

1) For each tuple r € R do
2) Use secrete key K, select the rightmost n values from K and store in W, where m>n.
3) Compute average of first and last digit of W and store in W_k.
4) If W_k is greater than n then take its mod and store result in it.
5) Now scan the W_kth attribute and find if there is any word whose abbreviation is possible.
6) If words are found then write it in its abbreviated form. Else select the digits between first and last digits of W and split them into two digit integer values.
7) Select first two values find its ASCII equivalent and embed first value at the beginning and second value at the end.

Fig 2: Key Insertion Algorithm

Scan the attribute to be watermarked and search for the words whose abbreviation is present in the metadata. If abbreviation is present then replace those words by their abbreviated text. So in this way the attribute get watermarked and also the overall database size is getting reduced.

But while scanning if we don't get any word which is therein metadata, then we have to use second technique. In this technique, select the digits between the first and last digits of the W, split it into two digit integer to get N_W values.

Select first two values and find its ASCII characters. Now append first character at the beginning and second character at the end of attribute data.

So in this way the insertion of watermark can be done in by using both syntactic and semantic approaches.

3.3 Watermarking Detection

As we are dealing with the two techniques so watermark detection is done in two phases. In first phase we are checking for an abbreviated text and in second phase we are checking for the first and last character of the watermarked attribute.

If there is difference in the secret key we automatically come to know that the user is not the authenticated user.

So in this way the integrity and authentication of the data is ensured while watermark detection.

1) For each tuple r € R do
2) Use secrete key K, select the rightmost n values from K and store in W, where m>n.
3) Compute average of first and last digit of W and store in W_k.
4) If W_k is greater than n then take its mod and store result in it.
5) Now scan the W_kth attribute and find if there is any word in the abbreviated form.
6) If it is present then watermark is detected.
7) Else select the digits between first and last digits of W and split them into two digit integer values. Select first two values find its ASCII equivalent.
8) Extract the first and last character of attribute.
9) If two values match with these characters then watermark is detected else suspicious data found.

Fig 3: Key Detection Algorithm

4. EXPERIMENTAL RESULTS

To implement the proposed watermark we have use MATLAB 7.6.0 to prepare a library of predefined signals for ASCII characters.

Consider a student database, where 'name' is considered as low impact attribute to be watermarked. Let us consider a tuple whose name field contains 'Julien Boutoille' as a name. Now we compute the secret key by using this tuple value.

1) Compute the set of low impact vowels (V), Consonants(C) and Special characters (P) and H_l from selected non numeric attributes. That is,

   V= {o}

   C= {J, B, n, t}

   S= {space between Julien and Boutoille}

   H= {15}

2) Now in set of low impact vowels we only have 'o'. So, our S_1 will directly become the ASCII of 'o'. that is,

   S1= 111

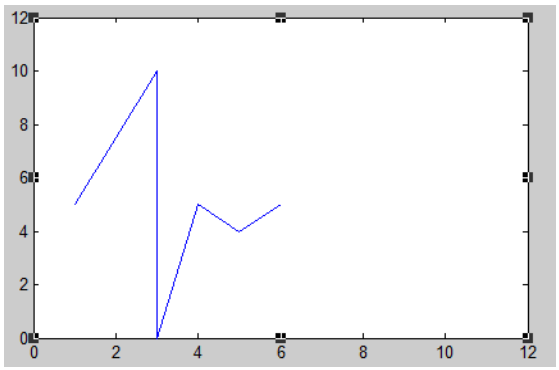

Fig 4: Signal for vowel o

3) Combining of signals of set of consonant is carried out by taking stem plot of the signals and then the stemmed signal is compared with library of predefined signals to map the correct ASCII value.

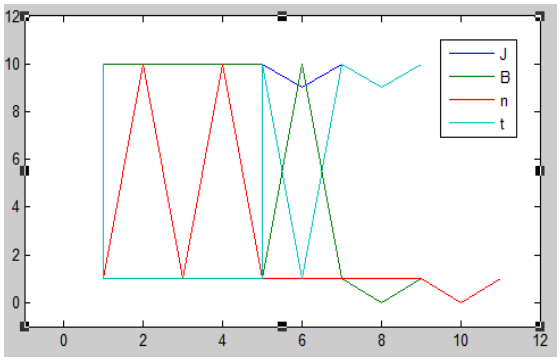

Fig 5: Signal for consonants J, B, n and t.

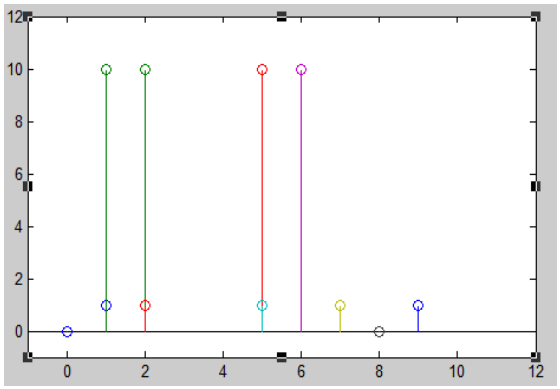

Fig 6: Stemmed Plot of J, B, n and t.

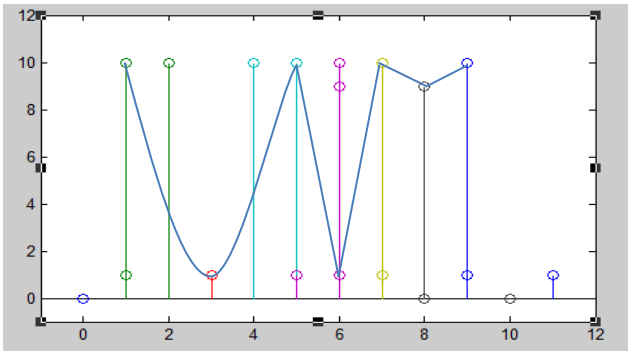

Fig 7: Mapping to predefined signals

4) Thus set of low impact consonant maps to the ASCII character 'l'. Now S_2 will be the ASCII equivalent of 'l'. That is,

   S_2= 108

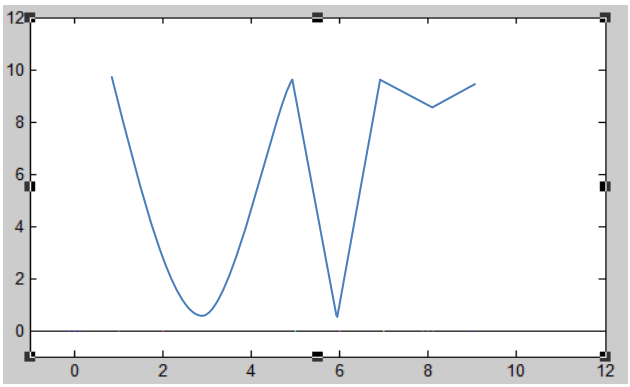

Fig 8: Mapped to consonant l

5) As there is only one special character. So the value of S_3= 32(ASCII equivalent of space).

6) The packed ASCII of 15 is H_l = 4953.

7) Now secret key S will be

$S= S_1 S_2 S_3 H_l$

S=111108324983.

Now we will deliberate how our system is robust against the various types of attacks like Bit Flipping attack, Mix and Match attack, Invertibility attack and Additive attack.

In Bit Flipping attack malicious user flips the values of bits that he guesses as a marked. But in proposed method the watermark is inserted as an abbreviation, so to guess a correct mark he should randomly select the tuples and flips the values of particular attribute. But the watermark of each tuple is not inserted in the same attribute, so he will not be able to detect the watermark.

Likewise in Mix and Match attack the malicious user select some tuples of relation R and mix it with tuples of other relation to create a new relation S same as R. If the watermark is inserted at same position for each tuple, then he will get expected matching bits. As watermark is not at same position for each tuple, therefore expected matching bits will not be obtained.

In additive attack malicious user tries to insert his watermark to tamper the original watermarking bits. The proposed method uses two techniques to watermark a tuple, so additive attack has no such effect on original watermark.

In Invertibility attack malicious user tries to get a key K which satisfactorily yields the watermark. It's not necessary that K should match with actual secret key. In other words K should approximately yield the watermark position. In the proposed algorithm generation of secret key is based on the signals, so mapping is somewhat difficult.

5. CONCLUSION

Thus we propose a unique approach for watermarking tuples in a relational database. Our approach makes uses signals. It inserts watermark with multiple bits on multiple tuples.

The proposed algorithm is robust against the various type of attacks like Bit Flipping attack, Mix and Match attack, Invertibility attack and Additive attack.

The results shown in this paper about the generation of the secret key are promising. In future we will try to optimize the proposed algorithm.

6. REFERENCES

[1] Lee, S. and Jung, S. "A survey of watermarking techniques applied to multimedia". In Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE '01), pages 272–277, Pusan, South Korea. IEEE Press, 2001.

[2] Potdar, V. M., Han, S., and Chang, E. "A survey of digital image watermarking techniques". In Proceedings of the 3rd IEEE International Conference on Industrial Informatics (INDIN '05), pages 709–716, Peth, Australia.IEEE Press, 2005.

[3] 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.

[4] 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.

[5] Y. Zhang, D. N. Zhao, D. Y. Li. "Digital Watermarking Techniques and Progress", Journal of PLA University of Science and Technology (Natural Science Edition) (In Chinese). 2003.6 4(3): 1-5.

[6] Bedi R., Thengade A., Wadhai V., "A New Watermarking Approach for Non Numeric Relational Database", 2011.

[7] Agrawal, R., Haas, P., and Kiernan, J. "Watermarking relational data: framework, algorithms and analysis". The VLDB Journal 12, 2 (Aug. 2003).

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

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

[10] 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.

[11] R. Sion, M. Atallah, S. Prabhakar, "Watermarking Relational Databases", Technical Report. Indiana: the Center for Education and Research in Information Assurance and Security of Purdue University, 2002.

[12] Mahmoud E. Farfoura, Shi-Jinn Horng, Jui-Lin Lai, Ray Shine Run, Rong-Jian Chen, Muhammad Khurram Khan, "A blind reversible method for watermarking relational databases based on a time-stamping protocol", 2011.

[13] Prof. R. Manjula, Nagarjuna Settipalli, "A new Relational Watermarking Scheme Resilient to Additive Attacks", International Journal of Computer Applications (0975 – 8887) Volume 10– No.5, November 2010.

[14] Adnan Abdul-Aziz Gutub, "e-Text Watermarking: Utilizing 'Kashida' Extensions in Arabic Language Electronic Writing", Journal of Emerging Technologies in Web Intelligence, vol. 2, no. 1, February 2010.

[15] Damien Hanyurwimfura, Yuling Liu and Zhijie Liu, "Text Format Based Relational Database Watermarking for Non-numeric Data", International Conference On Computer Design And Applications (ICCDA 2010).

[16] Chuanxian Jiang, Xiaowei Chen and Zhi Li, "Watermarking Relational Databases for Ownership Protection Based on DWT". Fifth International Conference on Information Assurance and Security (2009).