A robust watermarking approach for non numeric relational database

Conference Paper · October 2012

DOI: 10.1109/ICCICT.2012.6398095

Vidhi Khanduja
Netaji Subhas Institute of Technology
14 PUBLICATIONS 138 CITATIONS

A Robust Watermarking Approach for Non Numeric Relational Database

Vidhi Khanduja[1], Anik Khandelwal[2], Ankur Madharaia[2], Dipak Saraf[2], Tushar Kumar[2]

1. Department of Computer Engineering, Netaji Subhas Institute of Technology, India
2. Department of Information Technology, Delhi Technological University, India

Abstract - **One of the major issues concerning all the web application is security of data and proving authority of database. In this paper we present a robust technique of embedding watermark in a relational database with non numeric attributes. In the proposed technique, an attribute is selected using a pseudorandom generator and the watermark is embedded in it. The vowel to be embedded is selected by calling a Key_Extraction( ) on primary attribute of the same tuple. Our technique develops a secret key of embedded vowels which is provided to the user. The technique is robust against different malicious attacks and is blind and imperceptible, as proved by our analysis.**

I. INTRODUCTION

With the advent of information technology in every aspect of daily life, it has become necessary to protect the privacy of the digital data. Due to the vast vulnerabilities in the security and privacy of data in the digital world, copyright protection is a major concern. To overcome this matter in question, a technique has been developed which is known as Digital Watermarking. A lot of work has already been done in the problem of watermarking multimedia and many reliable solutions have been come up with. Efforts are now being made to develop a solution to the problem of watermarking relational databases. In the field of information hiding, watermarking relational databases has become a crucial desideratum. Basically, a digital watermark is a pattern of bits embedded into a file, especially minute perturbations in the database, which is used to identify the source of illegal copies. The following are features of an idyllic watermark:

1. It should be difficult or impossible to remove a digital watermark without noticeably degrading the watermarked content, to ensure that the copyright information cannot be removed.

2. The watermark should be robust. It should remain in the content after various types of manipulations, both intentional and unintentional.

3. The watermark should be imperceptible. Embedding the watermark signal in the digital data produces alterations, and these should not degrade the perceived quality of the data.

4. The watermark should be blind. It should be easy for the owner or a proper authority to readily detect the watermark. Such decodability without requiring the original, unwatermarked document would be necessary for efficient recovery of the property.

5. A Watermark key enhances the effectiveness of the watermarking technique. It is beneficial to have a secret key associated with each watermark that can be used in the production, embedding, and detection of the watermark. It should be private, so that it is difficult for an attacker to know where the watermark signal is.

In this paper, we design a new algorithm to embed watermark in a database concerned with primarily non-numeric attributes and propose a neoteric method of watermarking relational databases using the non-repetitive nature of occurrence of a vowel in a selected attribute value. In this scheme, we append a vowel to the selected attribute depending upon the key value generated for a tuple. The method provides the user with a secret key depending upon the vowels appended, which can be used later to verify the identification of the original database. If the database has undergone tampering during transmission, it can be recognized by the authenticated user.

II. RELATED WORKS

In this section, we briefly discuss some of the previous approaches related to our work. A method proposed by Agrawal *et al.*, generates a random number which when concatenated with primary key of the tuple, determines the tuple to be marked then the attribute to be marked and then the bit where the watermark is to be embedded [1]. This method is primarily proposed for numeric data values. Apart from this, other methods were also proposed in this field [5], [6], [7].

D.Hanyurwimfura *et al.* developed the method where a hash function H(K || r.P ||K) is applied on a number generated by concatenation of a random number K with the primary key of the tuple r.P, and (K || r.P || K) is called message authenticated code (MAC) [2]. Using MAC, tuple is selected and the watermark bit is embedded in multiword attribute having the least Levenshtein Distance (LD). However, R.Bedi *et al.* proposed the method where a tuple-relation matrix is generated using the primary key of each tuple and watermark embedding is done by using Eigen values in a non-numeric attribute of a tuple either as prefix or as postfix[3]. In this case, the watermark embedded is very clearly visible and hence can be easily removed.

M.Shehab *et al.* proposed the technique for numeric attributes where all the tuples are partitioned into groups based on the value of the hash function H (Ks || H (r,P || Ks))mod m [4]. A hiding function is defined which depending upon the watermarking bit to be embedded is maximized or minimized. The detection of watermark is a threshold-based technique which minimizes the probability of decoding error.

III. PROPOSED METHOD

Watermarking helps prevent piracy of data and many methods have been developed to prevent it, but some of them lead to significant change in the database and hence make it unusable in many cases.

In the proposed method, we embed minor changes in the attribute values such that they don't degrade the usability of the database thereby preventing piracy and ensuring copyright protection. Our method is robust against malicious attacks such as subset deletion, addition and modification. The proposed technique inserts a vowel adjacent to another vowel which is selected using Key_Extraction( ). The meaning of the word is not changed as only a sound of vowel to the word is added. Table I describes the symbols used in this paper.

TABLE I. SYMBOL TABLE

Symbol	**Description**
G	Pseudorandom generator.
S	Sequence of candidate attribute generated by pseudorandom generator.
K	List containing embedded vowels in original database.
L	Character embedded.
T	Threshold percentage i.e. minimum percentage required for detecting watermark.
R	Set containing all the tuples in the database.
C	Set containing all the candidate attributes selected by the user.
F	Function used for watermark verification, taking modified list K as input and giving output P as authenticity percentage.
P	ASCII Sum for each primary attribute.
M_i	ASCII value of the character under consideration of index i.
ω	Total number of tuples that have been watermarked.
μ	Total number of tuples in which watermark has been detected.
α	Significance level of the test.

A. THE KEY_EXTRACTION FUNCTION

The Key_Extraction( ) is an algorithm or subroutine that maps large data sets of variable length to smaller data sets of a fixed length [8], [10]. It calculates the key value for all the tuples and also the character to be embedded. The key value is calculated using a weighted ASCII sum and the character to be embedded depends entirely on the key value generated.
Let R be the relational database. To calculate the weighted ASCII sum, the prime attribute of every tuple of the set R is used. It is calculated by taking the sum of index value of characters of the prime attribute raised to the power of their ASCII values. The key is then used to decide the character, i.e. the vowel, to be embedded. Algorithm for Key_Extraction( ) is shown in figure 1.

$$P=\sum_{i=0}^{n}(index\ of\ character\ in\ primary\ attribute)^{M} \qquad (1)$$

Where, M is ASCII value of characters, and n is the length of primary attribute.

Example:

Primary Attribute: DCE

$P= (1^{68} + 2^{67} + 3^{69})$

Weighted ASCII sum (P) calculated using equation (1) = 832

```
Key_Extraction( )
1) For each tuple r ∈ R
2) set P = 0
3) for i =1 to length(primary_attribute)
4) P = P + (power ( i , $M_i$ ))
   [end of for loop of line3]
```

```
5)  key= P mod 5
6)  L is selected depending on the value of key.
7)  Return L
  [end of for loop of line 1]
```

FIGURE 1. ALGORITHM OF KEY_ EXTRACTION ( )

B. Watermark Insertion

A vowel is embedded in a candidate attribute which is selected by a pseudorandom generator. Pseudorandom generator uses a bit pattern depending upon the weighted ASCII sum and generates a sequence in which watermark is embedded.

The Key_Extraction( ) is called on prime attributes of all tuples in R. The vowel to be embedded and key value is returned by the function. The tuples R are partitioned into groups depending on this key value and all the tuples in a particular partition embed the same vowel. Occurrence of vowel is checked in the first candidate attribute and if no consecutive occurrence of the same vowel is found then the vowel is embedded in the position just following the vowel thereby increasing the length of attribute value by 1. However if consecutive occurrence (of the vowel to be embedded) is found, then next candidate attribute is selected from the sequence generated by the Pseudorandom generator. Pseudorandom generator generates computationally infeasible sequence of numbers depending on initial seed. If fixed initial seed is given Pseudorandom generators generate same fixed sequence of numbers. Pseudorandom generators are implemented in such a way that the output of this generator is a vector with number of states equivalent to candidate attributes [10]. These states decide what all attributes in a tuple are selected for embedding watermark. After this, occurrence of the vowel is searched in the new candidate attribute and the above process is repeated, however in this case the attribute value is prefixed with a single wide space.

```
1) Pseudorandom generator G is used to get a sequence S
   of candidate attribute names.
2)  for each tuple r ∈ R
3) call Key_Extraction ( )  //generates the character L to
           be embedded.
4) for each candidate attribute c ∈C
5)   for  i = 1 to c.length
6)          j = c.length
7)          if c(i)  =  L and c(i+1)  !=  L
8)          while ( j  !=  i )    //embed the character
9)                    c(j+1) = c(j)
10                    j = j-1
            [end of while loop]
11)         c(j+1)  =  L
 12)        else
13)                   go to line 4
            [end of if]
            [end of for  loop of line 5]
            [end of for  loop of line 4]
            [end of for  loop of line 2]
```

FIGURE 2. ALGORITHM OF WATERMARK INSERTION

C. Watermark Detection

In order to detect the watermark, we search for an embedded vowel in the candidate attributes in the order provided by the pseudorandom generator. The vowel embedded is generated by calling the Key_Extraction( ) on the primary attribute. Pseudorandom generator uses a bit pattern depending upon the weighted ASCII sum and generates a sequence in which watermark is embedded. The Key_Extraction ( ) is called on prime attributes of all tuples. The vowel to be detected and key value is returned by the function. We then search for consecutive occurrence of that vowel amongst the candidate attributes. If we only find one such attribute then it is considered as the one with the watermark. In case where we find multiple candidate attributes with consecutive occurrence of that vowel then the attribute value that is prefixed with a space is the one that has been watermarked. A threshold value is then calculated using the technique mentioned in [1] to detect the piracy of database. The threshold is denoted by τ and is calculated as:

$$\tau = \max\left\{ t \in \left[0, \frac{\omega}{2}\right] : \sum_{i=t}^{\omega - t} b\left(i; \omega, \frac{1}{2}\right) \geq 1 - \alpha \right\}$$

Where ($b(i; n, p) = nkp^{i}(1-p)^{n-i}$)   (2)

If the value of tuples in which watermark is detected is within a predefined range then the database is suspected to be a pirated copy of the original database.

```
1) Pseudorandom generator G is used to get the same sequence of attribute name as used in the insertion.
2) A list K containing all the embedded characters is maintained.
3) For each tuple r ∈ R
4)    Key Extraction function is used to find the key and the value L is returned.
5)    Traverse through every candidate attribute depending on the value of S, until a double occurrence of character L is found.
6)    If found, list K is modified by removing L and replacing it by some consonant.
   [end of for loop]
7) $\tau$ = threshold ($\omega$, α)
8) If ( ( $\mu$ < $\tau$ ) or ($\mu$ > ($\omega$ - $\tau$) )
        PIRACY SUSPECTED
```

FIGURE 3. ALGORITHM OF WATERMARK DETECTION

IV. EXPERIMENTAL RESULTS

The system we used to perform the experiments works on Windows 7 OS with 2.2 GHz Intel i5 core processor and 4 GB RAM. The database required for the proposed algorithm is created using Microsoft SQL Server Management Studio v10.0.1600.22. The watermark insertion and deletion procedures are implemented using MatLab 7.9.0 (R2009b). The watermark is then embedded on the database using the above mentioned algorithms.

The set of attributes we embed the watermark in, are chosen by the owner of the database such that fiddling with them leads to minimal loss of data. So, altering of a subset of such attributes during insertion goes unnoticed. So, the proposed watermark came out to be imperceptible. Then, to test the robustness For the algorithm, malicious attacks are performed on the watermarked database. The performance evaluation for robustness is based primarily upon two major attacks: Subset Addition Attack, Subset Deletion Attack and Subset Modification Attack

A. Subset Deletion Attack

In this category, the attacker may delete a subset of tuples from the watermarked database, so that the resulting database contains distorted watermark. However, this attack is undesirable by the attacker himself as it perturbs the watermark significantly at a cost of data loss. In the experiment, a random subset of tuples from the watermarked database is deleted. The ratios of the subsets selected are varied from 10% to 100% considering the interval of 10%. The watermark is again detected and respective efficiencies for different ratios are recorded. It is noticed that the efficiency of detecting the watermark in database after deletion of 50% of the tuples from the watermarked database is 53.33%, which is very high as compared to previous proposed methods. The efficiencies for different ratios are shown in the figure 4.

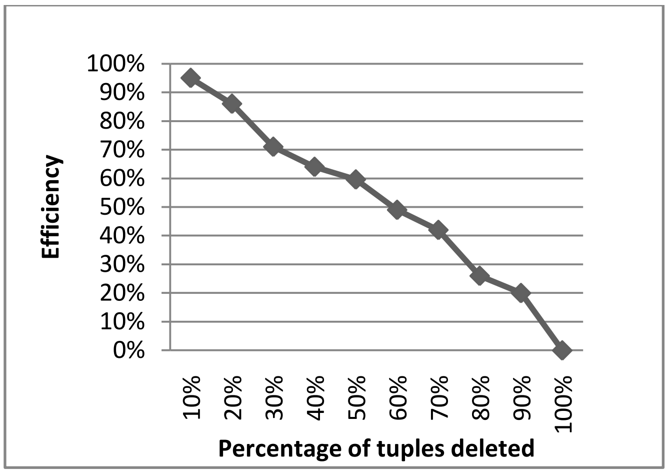

FIGURE 4. EFFICIENCY ON SUBSET DELETION ATTACK

B. Subset Addition Attack

In this category of attack, the attacker may add a subset of tuples to the watermarked database, with a hope that watermark will be removed or tampered. Random tuples are generated with the same properties as the tuples in the original database and are added to the database in the ratios of 10% to 100% with interval of 10%. The watermark is detected and recorded with different values of efficiency of the detected watermark on the database. The efficiency is calculated as the ratio of detected tuples to the total number of tuples which are watermarked.

As per the tests performed, the efficiency of watermark detection resulting from any ratio of subset addition attack never falls below 100%. This result solidifies our argument that our watermark is robust against subset addition attacks.

C. Subset Modification Attack

In this type of attack, the attacker tries to remove the watermark by modifying the values present in the database that he guesses have been marked. As the attacker has no idea which attributes have been marked, he randomly selects the attributes and modifies them. The ratio of number of tuples modified to the number of tuples in the database varies from 10% to 90% considering the intervals of 10%. The watermark detection method is applied on the modified database and efficiencies for different ratios are recorded. It is noticed that the efficiency does not fall below 66% even when 90% of the tuples were modified randomly. The efficiencies for different ratios are shown in the figure 5.

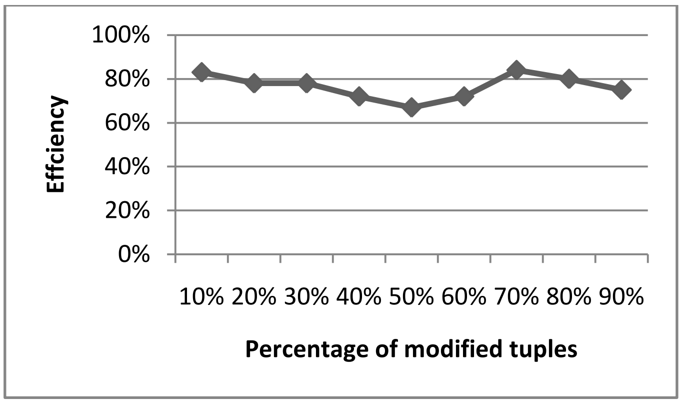

FIGURE 5. EFFICIENCY ON SUBSET MODIFICATION ATTACK

V. CONCLUSION

In this paper, we proposed a novel robust method suitable for watermarking non-numeric attributes in databases. We embed a vowel in the attributes depending upon the key value generated. The approach maintains integrity of the database.

Experimental results show that even if the database suffers from 50% deletion, 75% of the watermark is still retained and incase of insertion attack even after 50% insertion, the watermark detection percentage is still 100%, thereby ascertaining the robustness of the method.

A vowel is added to the existing data items thereby not changing the meaning of the data but adding an extra vowel sound to it, thereby maintaining the integrity of the database. Moreover a user can claim the ownership of database without requiring the original unwatermarked document but just using a secret key.

Hence the watermark has all the features which should be in an idyllic one and it proves the ownership of database without losing its integrity

REFERENCES

[1] Rakesh Agrawal,P.Haas,and J.Kiernan, "Watermarking relational data: framework, algorithms and analysis", The VLDB Journal 12, 2 (Aug. 2003), pp. 157-169.

[2] D.Hanyurwimfura, Y.Liu and Z.Liu, "Text Format Based Relational Database Watermarking for Non-numeric Data", 2010 International Conference On Computer Design And Applications (ICCDA 2010), pp-312-316.

[3] R.Bedi, A.Thengade and V.M.Wadhai,"A New Watermarking Approach for Non-numeric Relational Database",International Journal of Computer Applications (0975 – 8887) Volume 13– No.7, January 2011, pp-37-40.

[4] M.Shehab, E.Bertino and A.Ghafoor, "Watermarking Relational Databases Using Optimization-Based Techniques", IEEE Transactions On Knowledge And Data Engineering, Vol. 20, No. 1, January 2008, pp-116-129

[5] H.M.Sarsroudi, S.Ibrahim and O.Zanganeh, "Robust Database Watermarking Technique over Numerical Data", Journal of Communications and Information Sciences, Volume 1, Number 1, April 2011, pp-30-40.

[6] L.Zhang, W.Gao, N.Jiang, L.Zhang and Y.Zhang, "Relational Databases Watermarking for textualand numerical data", 2011 International Conference on Mechatronic Science, Electric Engineering and Computer August, pp-19-22, 2011, Jilin, China.

[7] N.S.Rani and M.V.V.S.N.P.J.Gopay, "Secure Embedding of a Robust Imperceptible Watermark in Relational data",(International Journal of Computer Science and Information Technologies, Vol. 2 (4) , 2011, pp-1780-1784.

[8] William Stallings, "Cryptography and network security", 4 th edition Pearson Education India, 2005

[9] Bruice Schneier, "Applied Cryptography, protocols, algorithms and source code in C", 2 nd ed. Wiley-India, 2008.

[10] Identification and Proof of Ownership by Watermarking Relational Databases," *International Journal of Information and Electronics Engineering, vol. 2,no.2,pp.274-277,2012*.