IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 3, Ver. III (May-Jun. 2014), PP 102-107
www.iosrjournals.org

Watermarking Relational Database Using Hindi Phonemes and Hill Cipher Technique

Rajneeshkaur K. Bedi[1], Dr. V. M. Wadhai[2]
[1]*(Computer Dept, MIT College of Engineering, Pune,India)*
[2]*(MIT College of Engineering, Pune,India)*

Abstract : *Digital watermarking now-a-days become more and more important due to tremendous availability of digital data on internet. The use of databases in various internet base applications has increased tremendously and theft of the data from database is a main concern for the database owners. Therefore, it is crucial to protect the piracy of the database. Most important technique in watermarking is its secret key and insertion location. In this paper, a new relational database watermarking method for non-numeric data is proposed based on nonnumeric attributes. A mark is computed based on the hill cipher technique. The position where mark is to be inserted is taken by the user. Our method is effective as it is robust and secure against different forms of malicious attacks.*

Keywords: *Copyright, Digital Watermarking, Hill Cipher, Phonemes, Relational Database.*

I. Introduction

Recent development in Internet application raised few questions on data security, copyright, authentication and integrity. As lots of data is being collected and stored in database through these applications. Knowingly or unknowingly user provides their personal, professional, educational, medical or financial data in these application which is supposed to be utilized by the user or the trusted party only but other people like researcher, marketing organization, hackers, enterprises etc are also interested to use this data which could be a great threat to their social or private life. Such personal data is sell or share hacked, alter gives rise to above mentioned question. Digital watermarking is one technique which addresses to tamper detection, ownership proof and traitor tracing.

The simple idea is to insert some identity as an error called marks to the original content such that it is not destroying the usefulness of data and not easily detected by malicious user, which later on can be used for ownership proof. All these marks together called watermark. Digital watermarking technique was widely used in images, videos, audios, VLSI and machine design.[1]. R. Agarwal[2] came with an idea of watermarking relational database using numerical attribute where tolerance of small error is acceptable. Such approach is not suitable for categorical data so, Sion et al[3] proposed another approach for certain type of application. Now the focus turned to non numeric, non categorical data for watermarking.

To briefly summarize we say that relational database watermarking is divided in distortion based and distortion free technique and various approaches are proposed based on data type like numeric, non-numeric(alphabetic), categorical and images.

Hill cipher is one of the poly graphic substitution multi-letter cipher based on linear algebra, developed by the mathematician Lester Hill in 1929 [4][5][6]. It is expressed as equation (1):

$$\mathbf{C = KP \bmod 26} \quad \ldots\ldots\ (1)$$

It is strong against cipher-text only attack but weak to known plaintext attack. This weakness is overcome by the modification we made in our approach as:

- Unique way for designing S-box based on Hindi phonemes
- Every individual is defining their own initial S-box
- generating K at a runtime based on above S-box
- replacement of character is in two phase

The key size is the binary logarithm of the number of possible keys. There are 74^{n^2} matrices of dimension n × n. Thus $\log_2(74^{n^2})$ or about $6.2n^2$ is an upper bound on the key size of our algorithm using n × n matrices. As we studied this is only an upper bound because not every matrix is invertible and thus usable as a key. Chinese Remainder Theorem is used to find invertible matrices [4].

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 gives an illustration with example of our proposed method and finally, section VI draws conclusions and future work and section VII is an acknowledgement.

II. Related Work

The survey paper by Raju Halder et. al [7], gives an insight on various approaches, attacks and issues for relational database watermarking technique. Accordingly our approach falls in distortion based category based on content characteristics and non-numeric multiword attribute for watermarking.

Another distortion based approach is proposed by [8], where they present new approach of adding a new “fake” tuple in the database. Their approach is effective watermarking technique for relational data and robust against various attacks. But adding a new tuple increases the size of database.

Watermark embedding proposed by [9] is by horizontally shifting the location of a word within selected attribute of selected tuples; a word is displaced right or left unmoved depending on watermark bit. The location where the mark to be inserted is determined by the Levenshtein Distance between two successive words within an attribute.

Distortion free watermarking scheme is proposed by [13] where a reversible data embedding algorithm referred as histogram shifting of adjacent pixel difference is used. This approach is robust but only applicable to numeric value.

Approached proposed by [10] uses non numeric attribute to compute eigen matrix and eigen value to generate secrete key and to identify position to watermark. This method limits itself as fewer eigen matrix may have fewer than real roots, or no real roots at all and ASCII value cannot be computed for Hindi phonemes.

Using non numeric attribute another method is proposed by [11] perform matrix operation by forming the matrix using number of vowels, consonants and ASCII value of each character the result is used to scale the image related to that record (profile image). In this method it is not necessary that every relational database has an image associated to each record. Again use of ASCII is problem and restricted by 3*3 matrix only.

Using predefined signals of each ASCII character and a set of abbreviation of words a novel method is proposed by [12]. But this method depends on signal processing tools and to maintain and compute data in signal form is bulky and time consuming.

Another approach of embedding watermarks in non numeric attribute is proposed by [14], where hash function and secrete key is used to generate watermarks.

III. Our Proposed System

Our approach is doesn’t require primary key. It works on the concept of Hill cipher. As we know in basic hill cipher we take 26 English alphabets and to add more complexity some additional characters are added. Here in our approach we add Hindi phonemes (vowels and consonants) along with English alphabet. The list is as follows:

Hindi Vowels:

A, E, I, O, U, EE, OO, RI, RE, AY, AN, AI, AU, UN, UH

Hindi Consonants:

KA, CK , KHA, GA, GH, NG, CH, CHHA, JA, JU, JHA, NY, TA, THA, DA, DHA, NA, PA, PH, BA, BU, BH, MA, YA, R, RA, RU, LA, LU, VA, VE, SH, TIO, SHA, SA, HA, KSH, TRA, GY

User had a choice to arrange these 74 characters (E26+H48) in his own sequence we termed it as S-box. This index in S-box is used in this computation. Here, in our case it is represented as equation (2):

C = KP mod 74 ………. (2)

Each Hindi phoneme is represented by a number modulo 74 in our case (as currently only 74 phonemes were identified later on we can add more). As to watermark a message, each block of *n* letters is multiplied by an invertible *n* × *n* matrix, again modulus 74 same we will do for watermarking with some additional steps.

The matrix used for watermarking is the secret key, and it is computed from the given attributes using the number of vowels, consonants, their length, frequency etc abbreviation is given in Table 1, and formed invertible *n* × *n* matrices (modulo 74).

Table 1: Abbreviations for algorithm

V_e,	No. of vowels in English
V_h,	No. of vowels in Hindi
C_e	No. of Consonants in English
C_h	No. of consonants in Hindi
FV_h	Frequency of occurrence of Hindi vowels
FV_{dcv}	Frequency of occurrence of double character of Hindi vowels
FV_{dcc}	Frequency of occurrence of double character of Hindi consonants
S_e,	Sum of English character
S_h	Sum of Hindi phoneme characters

Watermark Key Generation Algorithm:

User will define his S-box (i.e sequence vowels and consonants in his list) and store it with him.

Step 1: Forming matrix as K

Extract a set of vowel and consonant as per English alphabet and Hindi phonemes, then take a count of each as V_e, V_h, C_e, C_h, FV_h, FV_{dcv}, FV_{dcc}, S_e, $S_{h.}$. Form a matrix as:

$$\begin{bmatrix} Ve & Ce & Se \\ Vh & Ch & Sh \\ \mathrm{FVh} & \mathrm{FVdcv} & \mathrm{FVdcc} \end{bmatrix}$$

Step 2: Choosing P

1. Choose most unique letter from V_h (preferably two letter) / one frequently appearing vowel letter.
2. Choose one/two most unique consonants from C_h
3. Take index value of the identified letters. Form a single matrix P

Step 3: Compute C = KP mod 74

Step 4: Rearrange Alphabet list

From Step 3 we obtained a substitute letter for P,
User can repeat the above two steps (2 & 3) for more times (its' an optional)
now replace the obtained new letter in the same index of S-box (i.e reorder the main list)
Here, your key is ready. Store this key with the trusted third party.

Watermark Insertion Algorithm:

- Now choose the attribute where to insert watermark.
- Split the character according to Hindi phonemes
- Replace that attribute value character by character with the characters from new S-box.

Watermark Detection Algorithm:

- Retrieve the key from the trusted third party
- Split the character according to the Hindi phonemes
- Accordingly compare the character from the given attribute or in case of bigger dispute repeat the procedure for the secrete key generation using the basic S-box with user for the number of round he performed
- Compare the result

To understand it properly let's see an example in next section.

IV. Illustration Of Proposed Method

Consider the message 'RAJNEESHKAUR KARAMSINGH BEDI', and the key evaluated from:

Step 1: Forming matrix as K

V_e= {A, E,E,A,U,A,A,I,E, I } = **10** C_e= { R, J N S H K R K R M S N G H B D} = **16** **Total: 26**

V_h= {A,EE, AU, E , I, I} = **6** C_h={ RA, J, N, SH, K, R, KA, RA,M,S,NG,H, B, D } = **14** **Total: 20**

Frequency FV_h= 2 FV_{let} = 2 FC_h = 9

Now, **K** = $\begin{bmatrix} 10 & 16 & 26 \\ 06 & 14 & 20 \\ 02 & 02 & 09 \end{bmatrix}$

Step 2: Choosing P

Now, **P** = {EE, AU, SH} = $\begin{bmatrix} 5 \\ 12 \\ 48 \end{bmatrix}$

Thus the enciphered vector is given by:

Step 3: Computing C = K P mod 74

$$\begin{bmatrix}10 & 16 & 26\\06 & 14 & 20\\02 & 02 & 09\end{bmatrix}\begin{bmatrix}5\\12\\48\end{bmatrix} = \begin{bmatrix}1490\\1158\\466\end{bmatrix} = \begin{bmatrix}10\\48\\22\end{bmatrix} \text{mod } 74$$

Which, corresponds to a cipher -text of {AU, SH, CHHA}.

Step 4: Rearrange Alphabet list

From Step 2 we obtained
EE = AU
AU=SH
SH = CHHA
Replace them and according to it re-arrange the array list as follows:

0	1	2	3	4	5	6	7	8	9	10	11	12	13
A	**E**	**I**	**O**	**U**	**AU**	**EE**	**OO**	**RI**	**RE**	**AY**	**AN**	**SH**	**AI**
14	15	16	17	18	19	20	21	22	23	24	25	26	27
UN	**UH**	**KA**	**CK**	**KHA**	**GA**	**GH**	**NG**	**CH**	**JA**	**JU**	**JHA**	**NY**	**TA**
28	29	30	31	32	33	34	35	36	37	38	39	40	41
THA	**DA**	**DHA**	**NA**	**PA**	**PH**	**F**	**B**	**BA**	**BU**	**BH**	**MA**	**YA**	**R**
42	43	44	45	46	47	48	49	50	51	52	53	54	55
RA	**RU**	**LA**	**LU**	**VA**	**VE**	**CHHA**	**TIO**	**SHA**	**SA**	**HA**	**KSH**	**TRA**	**GY**
56	57	58	59	60	61	62	63	64	65	66	67	68	69
C	**D**	**G**	**H**	**J**	**K**	**L**	**M**	**N**	**P**	**Q**	**S**	**T**	**V**
70	71	72	73										
W	**X**	**Y**	**Z**										

Now accordingly replace the attribute character.
For example: City attribute: Value : CHANDIGAR
CH AN D I GA R = NG AY D I KHA R
Watermarked value: NGAYDIKHAR

V. Experiment And Results Analysis

The objective of our experimental testing is to check the robustness of our proposed approach against various attacks on watermarked data. Our experiment is implemented on Windows 8 run on Intel core i5 – 2430M CPU @ 2.40GHz, 4GB memory using Java Netbeans IDE 7.3.1 and Access in backend. We utilized our student database available with College administration section with the collection of 7000+ records. In the following, we present experimental result with respect to various attack and watermark detection.

Similar to the example mentioned illustration section we have processed a file containing, performed similarity check and obtained the following result.

The result shows that even if 60% of data is altered we have 50% of watermark detection successfully. The proposed method is resilient to sorting, brute force, addition, substitution and subtraction as explained in next section.

Figure 1: performance for additive attack

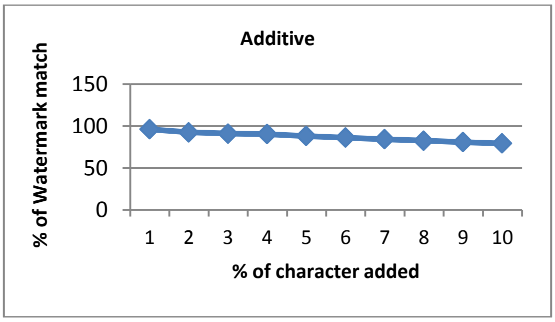


Figure 2: performance for deletion attack

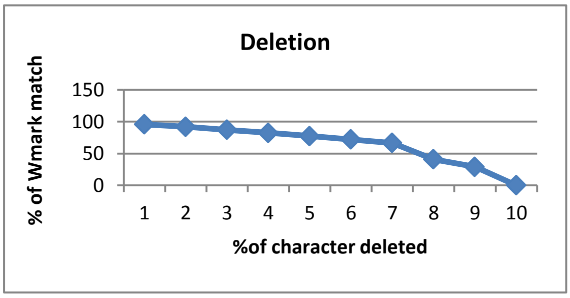


Figure 3: performance for substitution attack

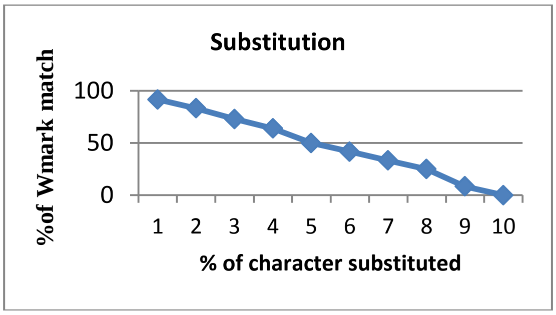


Figure 4: robustness analysis

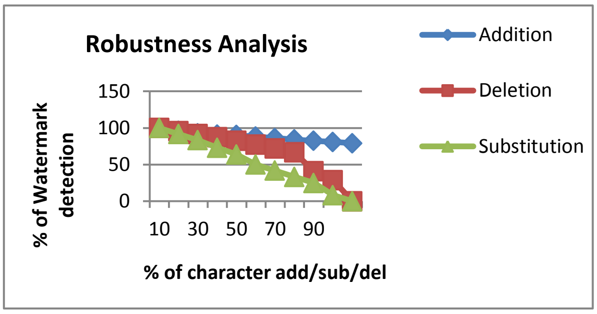
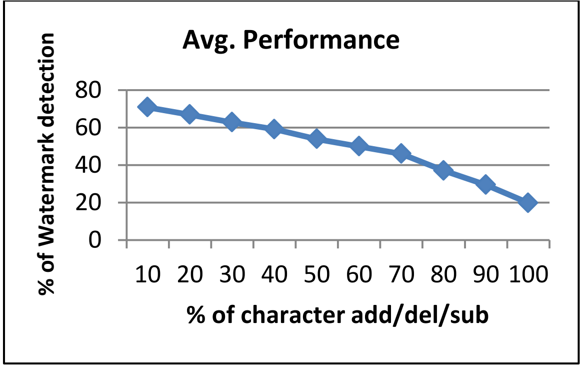


Figure 5: Avg. performance



Attacks on Watermarking:

In watermarking the aim of attacker is to remove or distort the original watermark inserted in database. Watermarking scheme need to be robust against such things. Possible attacks on any database could be any of the following:

Sorting attack:

Our embedding approach is based on individual tuple, so any kind of sorting attack will not affect our watermark.

Brute force attack:

In this case, attacker tries to guess secrete key by some know method or parameter. This attack has been overcome due to individual S-boxes and permutation of alphabetic character is 74!,. False Hit / False Miss: It is the probability of detecting / not detecting a valid watermark from the non watermarked database. In our approach it is difficult as the choice of S-box is user specific and marks are for individual tuple only.

Additive Attack:

In additive attack the attacker tries to add few characters to distort our mark. The proposed method uses its unique way to analyze the character based on Hindi phonemes so it withstands upto 75% of addition in the mark, refer Fig 1.

Subtraction Attack:

In subtractive attack the attacker tries to delete few characters randomly to distort our mark. The proposed method resists upto 63-60% below that the performance drops, refer Fig 2.

Substitution Attack

In substitution attack the attacker assumes he knows marking character and accordingly he made changes in attribute to distort our mark. The proposed method resists upto 80%, refer Fig 3.

Other type of attack like subset addition or deletion of tuple or attribute does not affect to our scheme as our method is based on individual secrete key generation and attribute choice for his own record.
Our approach is robust (refer Fig 4) against all of the above attacks, in worst case also we have 22-24% chance to detect our marks, refer Fig 5 and also to it overcome attribute value transposition.

VI. Conclusion

In the conclusion, let us summarize with the main features of the technique presented in this paper,

- Proposed method does not depend on primary key.
- It depends on non-numeric attribute.
- Secret key is stored with trusted party and/or once again can be computed with original values.
- It is robust against various malicious attacks.
- It is secure and easy to implement.
- Each tuple is individually processed based on user choice.
- It modify the attribute which are preferred by user (generally of less important in all) so, does not affect much to the correctness of database.
- We are also able to detect modification.

This study can be further extended to any text data. Further approaches can be derived from more secure and proven algorithm in cryptography.

Acknowledgements

This is the research work associated to the PhD work going on under Sant Gadge Baba Amravati University (SGBAU) and PG Department of Computer Science and Engineering, Amravati. My sincere thanks to the centre Head Dr. V. M. Thakre and the faculty for their extensive support..

References

[1]. 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), Peth, Australia.IEEE Press, 2005, pages 709–716.

[2]. R. Agrawal and J. Kiernan, Watermarking Relational Databases, Proceeding of the 28th VLDB Conference, VLDB Endowment Press, 2002, pp. 155-166.

[3]. Radu Sion , Mikhail Atallah , Sunil Prabhakar, Rights Protection for Categorical Data, IEEE Transactions on Knowledge and Data Engineering, 7(7), July 2005,p.912-926.

[4]. http://en.wikipedia.org/wiki/Hill_cipher

[5]. Forouzan B.A. and Mukhopadhyay D., Cryptography and Network Security, McGraw Hill, Second Edition

[6]. William Stallings, Cryptography and Network Security, Pearson, Fourth edition.

[7]. Raju Halder, Shantanu Pal, Agostino Cortesi, "Watermarking Techniques for Relational Databases: Survey, Classification and Comparison", Journal of Universal Computer Science, 16(21), 2010, pp. 3164-3190

[8]. Vahab Pournaghshband, "A New Watermarking Approach for Relational Data". ACM-SE '08,March 28-29,2008,Auburn AL,USA, pp 127-131.

[9]. H. Damien, Liu Y., and Liu Zh., "Text format based relational database watermarking for non-numeric data", 2010 International Conference on Computer Design and Applications (ICCDA 2010), IEEE Computer Society Press, June 2010, pp. 4312-4316,

[10]. R Bedi, A Thengade & V M Wadhai, "A New Watermarking Approach for Non-numaric Relational Database", International Journal of Computer Applications, 13(7)., 2011, pp. 37-40.

[11]. Bedi, R., Wadhai, V.M., Sugandhi, R., Mirajkar, A.: Watermarking Social Networking Relational Data using Non-numeric Attribute. International Journal of Computer Science and Information Security (IJCSIS) 9(4), 2011, pp 74–77.

[12]. Rajneeshkaur Bedi, Purva Gujarathi, Poonam Gundecha, and Ashish Kulkarni, "A Unique Approach for Watermarking Non-numeric Relational Database", International Journal of Computer Applications (0975 – 8887), 36(7), December 2011, pp.9-14.

[13]. Chin-Chen Chang, Thai-Son Nguyen, and Chia-Chen Lin, "A blind reversible robust watermarking scheme for relational database", The Scientific World Journal, vol 2013, Article ID 717165, 12 pages.

[14]. Nahla El_Hahhar, M. M. Elkhouly, Samah S. Abu El Alla, "Blind watermarking technique for relational database", COMPUSOFT, an International Journal of Advanced Computer Technology, 2(3), May 2013, pp. 121-126.