Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Technology 6 (2012) 988 – 995

Procedia Technology

2nd International Conference on Communication, Computing & Security [ICCCS-2012]

Relational Database Watermarking for Ownership Protection

Udai Pratap Rao [a], Dhiren R. Patel [a], Punitkumar M. Vikani [a,*]

[a] *Department of Computer Engineering , NIT-Surat (395007), India*

Abstract

With the widespread popularity of Internet, the use of databases has increased tremendously and theft of the database is a main concern for the database owners. Therefore, it is essential to protect the ownership of the database. In this paper, a new technique of database watermarking is proposed which based on inserting the bits of a binary image in relational database. The proposed technique also minimizes the variation in watermarked database. Experimental results justify the feasibility of the proposed technique and its robustness against common database attacks.

Keywords: watermarking relational databases; copyright protection; binary image as watermark information

1. Introduction

There has been tremendous interest shown in digital watermarking technology during the past two decades. This technology is indeed for copyright protection, rights management and copy control of digital contents [1]. Most of the research works focused on watermarking for multimedia data such as video, audio and images [2-4] wherein, variety of approaches have been proposed. For protection of types of databases, database watermarking technique can be proposed and implemented. The techniques developed for multimedia data can‘t be directly used for relational database because of differences in characteristics between both [5]. The main objective is to know that which type of information may be embedded into the database for ownership protection. In our approach we select image as a watermark. In order to improve the effectiveness and conviction, watermark insertion process should be imperceptible, safe and reliable; also watermark detection should be a blind detection process. In case of Blind detection, we should not require the original database at

* Corresponding author. Tel.: +91-9428067307.
E-mail address: udaipratapprao@gmail.com (U. Rao), dhiren@coed.svnit.ac.in (D. Patel), punitvikaniit@gmail.com (P. Vikani).

doi:10.1016/j.protcy.2012.10.120

the time of detection process. The watermark techniques should be able to tolerate some malicious attacks like subset addition attack, subset deletion attack, subset selection attack, subset alteration attack, bit flipping attack, rounding attack [6].

In this paper, we investigate a technique for relational database watermarking in which binary image is used as watermark. The image bits are inserted into database, which represent the copyright information. The overall variation in the watermarked database is also minimized.

Nomenclature	
M	Name of the image (m x n)
P	The primary key in relational database
V	Attribute numbers to be marked
ξ	Number of LSB candidate bits to be modified
Ks	Secret key
F	Fraction of the tuple that can be marked
α	Significance level for detecting a watermark

1.1. Related work and motivations

First method [5, 7] for relational database watermarking was exposed by Agrawal and Kiernan in 2002. This method marks only numeric attributes and marking done at a bit level.

Zhang et al.[8] proposed the method for relational database watermarking which uses image as watermark information. The pixels of an image have the relative positions. This method is based on the tuples' order. The watermark can not be retrieved, if an attacker changes the order of the tuples. This method is not efficient against some subset attacks.

In 2008, Sun et al.[9] introduced another technique for inserting an image into the database as watermark information. In this method, they converted one or more images into flow of bits. They used hash value of database tuple to find the location of each pixel and marked bit. They considered mod of hash value and watermark's length. If someone takes large image as watermark information, then length of watermark increases. And this method cannot insert all the pixels into the database. Therefore this method is not efficient for small databases.

Wang et al.[10] describes an image based watermarking scheme which embeds a scrambled image based on Arnold transform. This technique converts a scrambled image into binary string. Suppose the length of this binary string is L. And whole database is divided into L groups. It computes 1hash value using private key, database's primary key and order of an image. According to this hash value, a particular group among all L groups are found. The i^{th} bit of binary string is inserted into the specific bit position of the attribute value. This i^{th} bit is chosen algorithmically [10]. The efficiency of this technique is improved because it depends on many factors like private key, scrambling number and order of an image. This technique uses only one fixed attribute to insert watermarks.

Cao et al. [11] introduced a new technique that uses EMC (Encrypted Mark Code) to convert the original image into bit flow and then, the similar steps are used as in [10]. This technique does not consider the order of the image also. At last, the usefulness of the database is checked. The modification is committed if acceptable, otherwise rolled back.

In our technique, we establish a relation between an image and a relational database. This relation represents the copyright information. Our technique uses only numeric attributes for marking. We assume that slight modifications in certain numeric attribute values are tolerated. The data owner decides which numeric attributes are appropriate for marking among all the attributes, because modification in some numeric attributes are not appropriate like license no., vehicle no., account no., etc. In our approach, we attempt to remove the drawbacks of the existing approaches as mentioned above.

The rest of the paper consists of three sections. Section 2 describes our proposed technique, which includes watermark insertion and watermark detection algorithm. Section 3 includes the performance evaluation of proposed technique. Section 4 includes the conclusions and references at the end.

2. Proposed Work

Let us take a relational database R. The numeric attributes available for marking are represented as A_0, A_1,....., A_{v-1}, where v is the numeric attributes available for marking. The data owner will also decide two other important parameters F and ξ and these parameters define the limit of modification on databases.

2.1. Watermark insertion algorithm

The stepwise watermark insertion algorithm is given below:

// we insert a watermark information into the original database R, and return marked R.
// the parameters M, F, v, ξ and Ks are all known to the data owner only.

1. Convert an image (m x n) into matrix of 0 & 1, and store this matrix into **W[m][n]**.
2. For each tuple r in R do
3. t = HASH(Ks concatenate r.P)
4. if(t mod F == 0) then // this tuple is available for marking
5. attribute_index i = t mod v // mark attribute Ai
6. bit_index j = t mod ξ // mark j^{th} bit
7. select row of an image a = (i * v) mod m
8. watermark_index k = t mod length(a) // it gives some bit position in a^{th} row of watermark(image)
9. h = (HASH(t concatenate k(row value))) mod m // h is the position for selected mark bit from M
10. w = (HASH(t concatenate k(col value))) mod n // w is the position for selected mark bit from M
11. Replace the j^{th} LSB of r.Ai with **W[h][w]** bit
12. Now, apply the minimize variation
13. Update R;
14. End loop;

Initially the watermark image is transformed into a binary matrix and then the secret key and the primary key of the tuple is considered together. At the next step, its hash value is computed by the use of MD5 algorithm. The computed hash value and value of F determine whether the tuple will be marked or not.

After that, the attribute index (i) and bit index (j) is determined where attribute index specifies that particular attribute is selected for watermarking amongst all available numerical attributes A_0, A_1, ..., A_{v-1}.

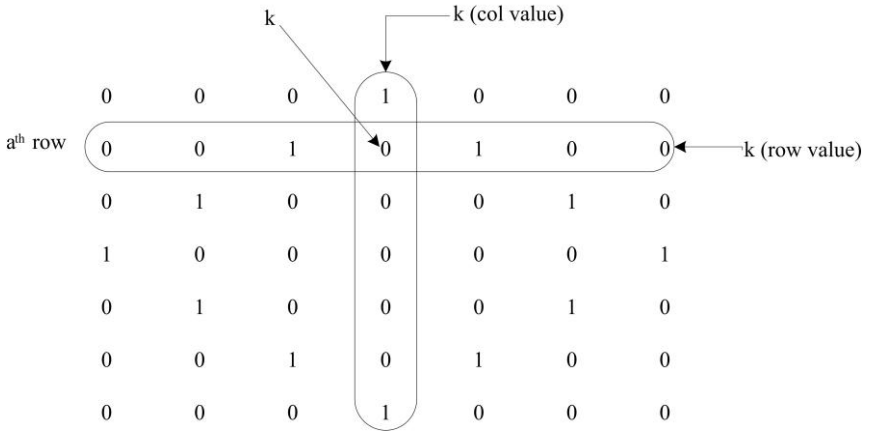

Fig. 1 Select k(col value) and k(row value) from watermark image

Bit index specifies LSB bit position of the selected attribute e.g. if the bit index is 3 then the 3rd LSB bit is chosen for watermark injection.

To understand how the algorithm works, we consider binary matrix of an image. We select one row (a) from the image using the formula in line 7 of our insertion algorithm. Line 8 obtains the watermark index and k (col value) is a binary string composed of the k^{th} bits of each row of the image, k (row value) is the value of a^{th} row of the image. The overall operation is shown in Figure 1.

In lines 9 and 10 of our insertion algorithm, we find h and w values, which give us the position of one bit of the image. At this point, we already know the value of i and j. So, the j^{th} LSB bit of the selected attribute is replaced by **W[h][w]**. This watermark is inserted into the relational database. After this, we minimize the variation in the numerical attribute's value. This is done if the selected LSB is not the first LSB. We have ξ number of LSB bits available for modification.

If the selected LSB bit is 0, then it is replaced by 1 during the watermark insertion. As shown in Figure 2, the attribute's original numeric value is 2730 and after watermark insertion it's value becomes 2746. There is a change of 16 in the numerical value. If we want to reduce this difference, then we can minimize the variation in the attribute's value. In addition, the new value is greater than the original value that means we have to reduce the value of the watermarked attribute. To do this, we invert all the right side bits of selected LSB bits whose value is 1 in the watermarked attribute. Then this value becomes 2736, which is shown in Figure 2. In this example, we reduced the variation by 62.5%.

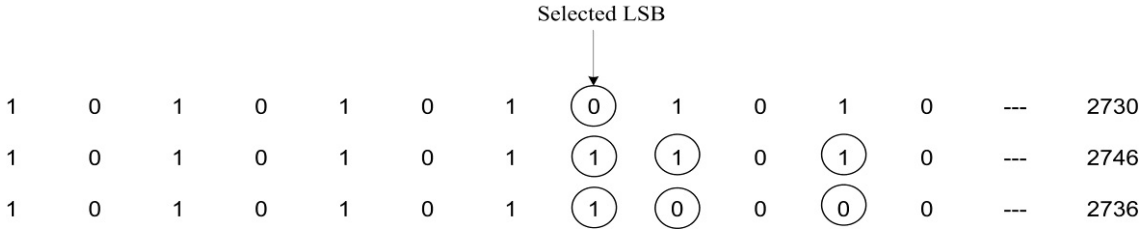

Fig. 2 Minimize Variation 0->1

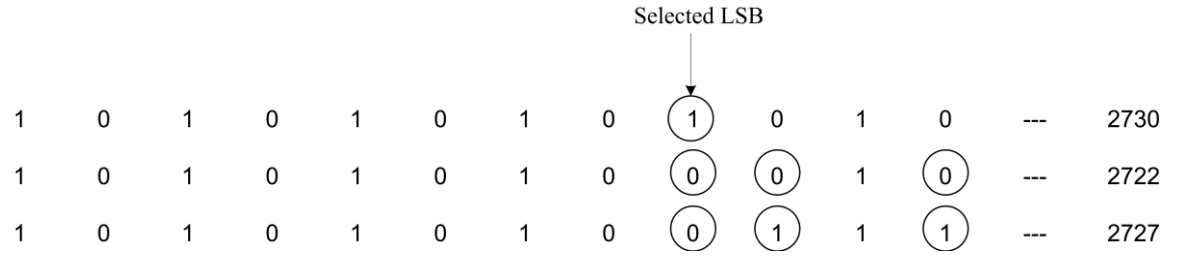

Fig. 3 Minimize Variation 1->0

If the selected LSB bit is 1, it is replaced by 0 as shown in Figure 3. Initially, the attribute's numeric value is 2730 and after watermark insertion, it becomes 2722. In this case, the new value is lesser than the original value. Therefore, we have to increase the value of the watermarked attribute. To do this, we invert all the right side bits of selected LSB whose value is 0 in the watermarked attribute and then, this value becomes 2727 which is shown in Figure 3. We can minimize the variation by this wrinkle.

2.2. Watermark detection algorithm

The same technique (insertion algorithm) is used to identify the marked tuples. For each marked bit, we observe the total count and match count. Line 16 in this algorithm decides the ownership of database.

// we use two integer parameters *total_count*=0 (to count the total no of watermarked bits) and
// *match_count*=0 (to count the total no of matched watermarked bits).

1. Convert an image(m x n) into matrix of 0 & 1, and store this matrix into **W[m][n]**.
2. For each tuple r in R do
3. t = HASH(Ks concatenate r.P)
4. if(t mod F == 0) then // select this tuple
5. attribute_index i = t mod v // mark attribute Ai
6. bit_index j = t mod ξ // mark j^{th} bit
7. select row of an image a = (i * v) mod m
8. watermark_index k = t mod length(a) // it gives some bit position in a^{th} row of watermark(image)
9. h = (HASH(t concatenate k(row value))) mod m // h is the position for selected mark bit from M
10. w = (HASH(t concatenate k(col value))) mod n // w is the position for selected mark bit from M
11. *total_count*++;
12. if **W[h][w]** matched with j^{th} LSB
13. *match_count*++;
14. End if;
15. End loop;
16. if(*match_count* / *total_count* >= α)
17. Has watermark

18. else
19. No watermark

3. Performance Evaluation

The proposed technique has been tested and evaluated with the help of experimental database. We created this experimental database by random number generator. This database consist 15000 tuples and 8 numeric attributes. We performed our experiments on Windows XP operating system with 1.6 GHz Intel CPU and 2GB RAM. We used Sun JDK 1.7 and MYSQL 4.1 in our experiments. In proposed work, we claimed that F, ξ are two important parameters to define the limit of modification on database. F is a control parameter that determines the selected number of tuples for marking, same as given in table 1. As the value of F increases, selected number of tuples for watermarking decreases. In addition, we insert watermark in one of the ξ LSB bits in selected attribute. If we choose $\xi = 5$, means one of the last five bits is available for modification. In addition, if we choose small ξ, then the value of that attribute changes slightly.

Table 1 Relation between F and selected no. of tuples

Value of F	**Selected no. of Tuples for Watermarking(out of 15000)**
F=1	14995
F=2	7425
F=5	2981
F=10	1479
F=15	976
F=20	746

We applied some attacks on our technique like subset deletion attack, subset addition attack, subset alteration attack, subset selection attack. In these attacks, we select the subsets randomly and results of these attacks are shown in below figures [4-7].

3.1. Subset deletion attack

This is a simple attack technique, which attacker uses most of the time. In subset deletion attack, attacker deletes some part of the watermarked database randomly. Main aim of the attacker is to remove the watermarked. To verify this attack, we randomly deleted some part of the database. The result is shown in the form of chart in figure 4. According to the chart we can say that, 50% watermark would be detected even if an attacker deletes 50% of the database.

3.2. Subset alteration attack

In this type of attack, the attacker not only deletes some original data but also adds some new data. It may produce double data damage. The experiment result of subset alteration attack is shown in figure 5. It shows that even if 50% of the data is altered then also significant amount of watermark is detected.

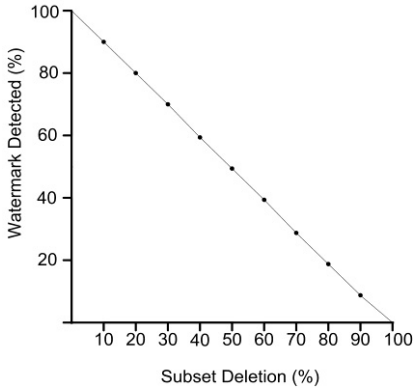

Fig. 4 Subset Deletion Attack

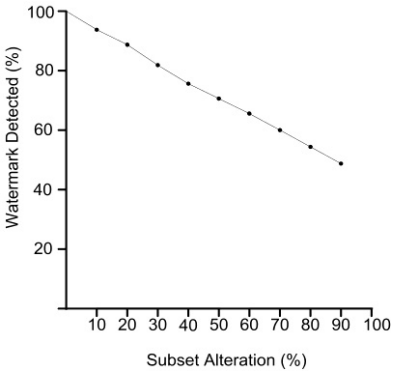

Fig. 5 Subset Alteration Attack

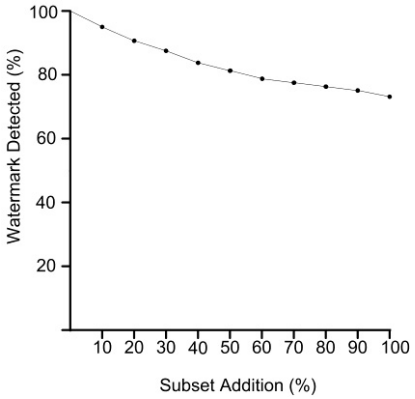

Fig. 6 Subset Addition Attack

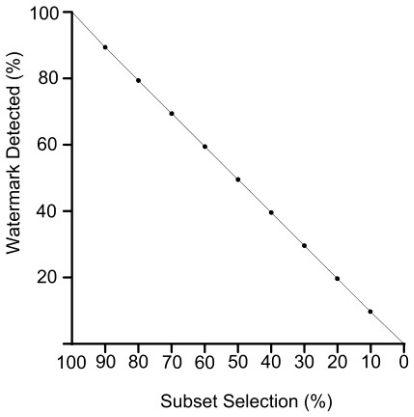

Fig. 7 Subset Selection Attack

3.3. Subset addition attack

It is obvious that the watermark can be affected with the data increasing. In this type of attack, the attacker adds some random or duplicate data to the original data. Figure 6 shows our experiment result on this attack. We can see that, Significant amount of watermark would be detected even if the attacker added as many tuples as the original tuples.

3.4. Subset selection attack

The attacker randomly selects and uses only subset of the original database. From that subset, attacker gets what he wants. The attacker thinks that small subset of large database is selected, and it does not contain watermark. The result of our experiment is shown in figure 7. It shows that the watermark will remain in the selected database even if the attacker selects a subset whose size is 10% of the original database. This is because our proposed algorithm inserts watermark in the whole database.

4. Conclusions

In this paper, we proposed an effective technique for database watermarking in which a suitable tuple in the database is chosen for marking and then the chosen bits of an image replace some bits of the selected attributes of particular tuple. The proposed technique minimizes the variation by inverting some bits of the watermarked attribute. The experimental results demonstrate the robustness of our technique against deletion, alteration, selection and insertion attacks and shows that proposed technique is irrespective to the tuples' order.

References

[1] Zhang Yong, Niu Xia-mu, Wu Di, Zhao Liang, Li Jun-cao, Xu Wei-jun, 2002. A Method of Verifying Relational Databases Ownership with Image Watermark. Multidiscipline Scientific Research Foundation of Harbin Institute of Technology Project, Project Number: HIT.MD-2002.11.

[2] Langelaar, G.C., Lagendijk, R.L. and Setyawan, I., 2000. Watermarking Digital Image and Video Data, In IEEE Signal Processing Magazine, vol. 17, p. 20-43.

[3] Arnold, M., 2000. Audio Watermarking: Features, applications and Algorithms, In Proceedings of the 5th IEEE International Conference on Computer and Multimedia and Expo, vol.2, p. 1013-1016.

[4] Potdar, V.M., Han, S. and Chang, E., 2005. A Survey of Digital Image Watermarking Techniques, In Proceedings of the IEEE 3rd International Conference on Industrial Informatics, p. 709-716.

[5] Agrawal, R. and Kiernan, J., 2002. Watermarking relational databases, In Proceeding of the 28th International conference on Very Large Databases, p. 155-166.

[6] Raju Halder, Shantanu Pal, Agostino Cortesi, 2010. Watermarking Techniques for Relational Databases: Survey, Classification and Comparison, Journal of Universal Computer Science, vol.16, no.21, p. 3164-3190.

[7] Agrawal, R., Haas, P.J. and Kiernan, J.,2003. Watermarking relational data: framework, algorithms and analysis, VLDB Journal, vol.3.

[8] Zhi-hao Zhang, Xiao-ming Jin, Jian-min wang, De-yi li, 2004. Watermarking relational database using image, In Proceedings of International Conference on Machine Learning and Cybernetics, vol. 3, p. 1739-1744.

[9] Jianhua Sun, Zaihui Cao, Zhongyan Hu, 2008. Multiple Watermarking Relational Databases Using Image, In IEEE International Conference on MultiMedia and Information Technology, p. 373-376.

[10]Chaokun Wang, Jianmin Wang, Ming Zhou, Guisheng Chen, Deyi Li, 2008. Atbam: An Arnold transform based method on watermarking relational data, In Proceedings of the 2008 International Conference on Multimedia and Ubiquitous Engineering, p. 263-270.

[11]Zhongyan Hu, Zaihui Cao, Jianhua Sun, 2009. An Image Based Algorithm for Watermarking Relational Databases, In Proceedings of the 2009 International Conference on Measuring Technology and Mechatronics Automation, p. 425-428.