IOSR Journal of Engineering (IOSRJEN)
ISSN: 2250-3021 ISBN: 2878-8719 PP 78-83
National Symposium on engineering and Research

Watermarking relational databases: A Review

Prof.Bhawana Ahire, Prof. Neeta Deshpande
Assistant Professor, Computer Department, GNSCOE, Nashik (MS), India
Associate Professor, Computer Department, MCERC, Nashik (MS), India

Abstract: *- With the wide use of internet and growing technologies, one needs to protect the confidential information and data from the intruders and data hackers. Watermarking is an information hiding technique which is used to embed a mark within some host content. One of the main applications of watermarking is to prevent non-compliance of copyrighted works by embedding a copyright mark within the content to be protected.*

There is a rich body of literature for watermarking techniques to address these issues. Steps of watermarking relational database basically include data partitioning watermark embedding, decoding threshold evaluation and threshold decoding. The desirable properties of the watermarking relational databases are robustness, imperceptibility, private key selection, etc. This paper strongly focuses on the review of four relational database watermarking techniques proposed by researchers [R.Agarwal, R.Sion, Zhi-Hao Zhang and M.Shehab].

Keywords: - *Digital watermarking, robustness, relational database.*

I. INTRODUCTION

With the tremendous use of World Wide Web, authors of digital media can easily distribute their works by making them available on Web pages or other public assembly. To overcome the problems of piracy, one method is to embed additional information in terms of image, text, etc and only distribute the media that contains this additional information [1].

This embedded data which is in terms of information about the media, author, copyright or license information is termed as watermark [2]. Tremendous growth in digital watermarking has increased interest in intellectual property and copyright protection[3]. There are a lot of watermarking techniques which have been developed for video, images, audio, text data as well as natural language text [2],[4],[6],[7]. But not too much attention has been given to the watermarking relational databases. Hence, in this paper we have reviewed some techniques based on relational databases using optimization based techniques [10],[4],[7]. In general, the database watermarking techniques consist of two phases: Watermark Embedding and Watermark decoding [3]. In watermark embedding process, a watermark is embedded into the dataset D by encoding the bits. For encoding the bits, secret key Ks is used which is required for security purpose as it is known only to the owner of the dataset D. The watermarked database Dw, is then made publicly available. For checking the ownership of the mistrustful database, decoding procedure is performed by giving watermarked data as input. The watermark should remain unaltered though tuples would be added or deleted from the relational database. Figure 1 depicts the basic database watermarking technique.

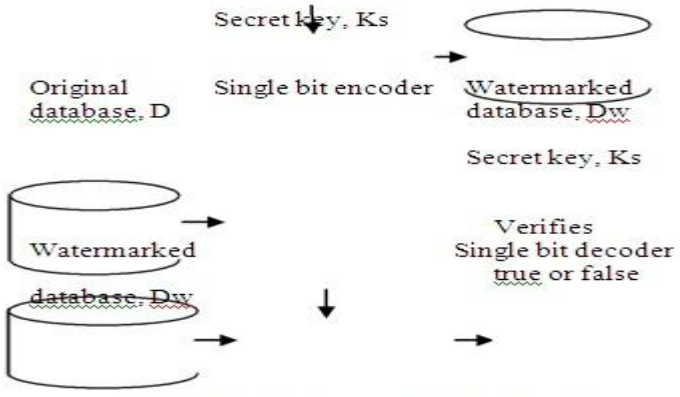

Fig.1.Basic watermarking technique

The organization of the paper is as follows: Section 2 describes the literature survey done by different researchers including watermarking basics, characteristics, different types of attacks on watermarking relational databases, Section 3 elaborates on comparative analysis of different watermarking techniques whereas section 4 summarizes the paper and suggests future directions.

II. LITERATURE SURVEY

The technique used to hide a small amount of digital data in a digital signal in such a way that it can't be detected by a standard playback device or viewer. In Digital Watermarking, an indelible and invisible 'message' is embedded into both the image and the audio track of the motion picture as it passes through the server. According to R. Agarwal, piracy of digital assets such as image, video, audio and text can be protected by inserting a digital watermark into the data thus providing a promising way to protect digital data from illicit copying and manipulation [4]. After embedding the watermark, the data and watermark are in-separable [11].

The security of relational databases has been a great concern since the expanded use of these data over the Internet. Because data allow unlimited number of copies of an "original" without any quality loss and can also be easily distributed and forged[11]. Hence, Digital watermarking for relational databases emerged as a candidate solution to provide copyright protection, tamper detection, traitor tracing and maintaining integrity of relational data [13].

The desirable characteristics for watermarking relational databases are [4]:- Detectability: This property says that Alice i.e. the owner of the database should be able to detect the watermark by examining the tuples from the suspicious database. -Robustness: *Robust* means that embedded watermarks should not be washed out by slight modifications of the watermarked content. We say that the watermark is robust when the attacker is not able to destroy it[10].-Capacity: It is the maximum amount of data that can be embedded and the optimal way to embed and extract this information.-Private key selection: According to Kirchhoff's, the method used for inserting the watermark is public. To protect the watermark from the intruder, the private key should be selected properly.

-Updatability: The watermark algorithm should be such that either the tuples of the relational database are inserted or deleted, the watermark value should not be changed.

-Security measures: Security of the watermarking depends on the choice of the private key.

Generally, in a robust watermarking scheme, the embedded watermark should be robust against various attacks which aim at removing or distorting the watermark. While in a fragile watermarking scheme, the embedded watermark should be fragile to modifications so as to detect and localize any modification in presence of different attacks [13]. The watermarked database may suffer from various types of intentional and unintentional attacks which may damage or erase the watermark, as described below [4]

1. Benign Updates: Let us suppose that Mallory has stolen Alice's data and subsequently, modifies the data as he uses it. Watermarking should be such that Alice doesn't lose her watermark in the stolen data in spite of Mallory's updates [4].
2. Malicious Attacks: Mallory has stolen the data which has a watermark, but he may try to erase the watermark or may claim false ownership. The watermarking system's malicious attacks: -Bit attacks: In this type of attack, malicious attack attempts to destroy the watermark by updating some of the bits. -Randomization attack: It assigns random values to some number of bit positions [4]. -Bit flipping attack: It inverts the values of the some number of bit positions [4]. -Subset attack: In this attack, Mallory takes a subset of the attributes of a watermarked relation and hope that the watermark is lost [4]. -Mix and match attack: In this attack, Mallory creates his own relation by taking disjoint tuples from multiple relations containing similar relations [4].
3. Deletion attack: Here, the Mallory deletes marked tuples from the relational database which leads to synchronization errors [10].
4. Alteration attack: In this attack, Mallory alters the data values of the tuples which leads to disturbance in the watermark. Altering the data values violates the usability constraints and makes the data useless [10].
5. Insertion attack: Mallory inserts tuples to the data set hoping to disturb the embedded watermark which results in synchronization errors [10].

III. COMPARITIVE ANALYSIS OF WATERMARKING RELATIONAL DATABASES

a) Algorithm proposed by R.Agarwal and J Kieman for watermark insertion and detection:

Agarwal et al. [4] proposed a watermarking technique that marks the only numeric attributes and tolerate some changes in some of the values. This algorithm embeds the watermark bits in the least significant bits(LSB) of selected attributes of a selected subset of tuples The one who has access to the primary key, can detect the

watermark with high probability[4]. Fig.2 (a) shows the watermark insertion algorithm proposed by Agarwal et al.[4].

//Primary key K, known only to the owner of the database, g is a gap parameter,
//The parameters r, i, j relation, tuple of a relation, attribute index and bit index respectively private to the owner

1. For each tuple r ϵ P do if (F(r, P) % g) ==0) then //mark the attribute Ai //mark the jth bit
2. mark the primary key k & bit index j
3. Calculate the hash of secret key and Primary key K of the database using Message Authenticated Code
4. If(hash_val is even) then set LSB bit of the attribute of the database relation to 0 Else set it to 1
5. Return i

Fig.2 (a) Watermark insertion algorithm

Watermark detection : Alice is the owner of the database relation R and Mallory makes changes to the database but, will not be able to make changes into the primary key as it contains valuable information[4]. Changes to it will make the database less useful from the user's point of view. Fig.2 (b) shows the watermark detection algorithm.

//Parameter considerations are same as followed by the watermark insertion algorithm.
//α is the test significance level that the detector preselects.

1. Total_c=match_c=0
2. For each tuple s ϵ S do
   if (F(r,P) % g) ==0) then //mark the attribute Ai //mark the jth bit //increment the Total_c by 1 Match_c=match_c+(number returned by matching the private key,attribute index, bit index of the relation S)
3. Calculate the threshold
4. if(match_c≥threshold) then privacy is suspected
5. match the private key, attribute index and bit index of the relation R and return int
6. Calculate the hash of secret key and Primary key K of the database using Message Authenticated Code
7. If(hash_val is even) then return 1 if set LSB bit of the attribute of the database relation is 0 Else return 0
8. Else return 1 if LSB bit of the attribute of the database relation is 1 Else return 0

Fig.2 (b) Watermark detection algorithm

This technique can't be used for multi-bit watermarks.LSB bits in any tuple can be altered without checking data constraints. It is also not resilient to insertion, deletion and alteration attacks.

b) Watermark encoding and decoding algorithm developed by R.Sion:

R.Sion [6] proposed a watermarking algorithm that embeds the watermark into the relational database using data portioning technique. In data partitioning technique, marker tuples are used which makes it vulnerable to synchronization errors. Hence, it is not resilient to deletion and alteration attacks. Further, M.Shehab et al. proposed a technique which enabled the decoder to reconstruct the underlying partitions [10]. Algorithm proposed by R.Sion [6] works in 2 stages i.e. encoding and decoding the bits. The algorithm starts by checking the fit tuples determined by the attribute and primary key of a relation. For checking the tuples, hash function is applied which uses SHA or MD5 algorithm for hashing. Fig.3 (a) depicts encoding algorithm [6].

Input: A relation with attribute A which is to be watermarked, a watermark w, a set of secret keys (ks1, ks2) and other parameters (e i.e. adjustable encoding parameter) which is used in the embedding process.

w_data=ECC.encode (w, w.len)
1. for (j=1 to n)
2. if(hash(primary key, secret key) % e)==0) then //Check the fit tuples
//set the bit index to t
//set the attribute index to at
3. Embedding_map [Tj(k)]□ id
Id□ id+1Return embedding_map

Fig.3 (a) Encoding algorithm

For the decoding phase, we assume the following inputs: Watermarked data, the secret key k1, K2 and e i.e. embedding bandwidth. Again, fit tuples are checked using Hash function. The aim of decoding algorithm is to discover the embedded w_data bit string. Fig.3 (b) depicts the decoding algorithm.

Inputs: Primary key K, secret key k1, k2, ECC

1. for(i=1 to n)
2. if(hash(primary key, secret key) % e)==0) then //Check the fit tuples find t (i.e. index of attribute A)
3. w_data[MSB(Hash(Ti(k),k2),b(n/e))]=t & 1
4. w ECC.decode(w_data,w,len)
5. return w

Fig.3 (b) Decoding algorithm

Hence, algorithm proposed by Sion [6] is resilient to alteration and data loss attack.

c) Watermark embedding and detection technique proposed by Zhi-Hao Zhang:

Zhi-Hao Zhang [7] proposed a watermarking algorithm in which an identification image is embedded into the relational data for representing the copyright information. This algorithm takes input as the relation R with the attributes as R(K.A0,A1,…..An) where K is the primary key of the database which is never marked[7]. The pixel values are marked as I(v0,vi,….v). The relation R is divided into group of uniform size equal to the size of the image. The algorithm compares a pixel value with an attribute value of a tuple in a relation. Pixel value (0 to 255) is divided into 3 parts. Three types of watermarks are inserted into the relational data. Else if 0 or 255 are repeatedly present, lots of attribute values will he marked as the same numbers in their decimal. Fig.4 (a) describes Watermarking embedding algorithm.

Inputs: Relation R(K.A0,A1,…..An) where K is the primary key

1. for each tuple r ϵ R do
2. if vi=255 then
   mark (r.Ai %3)=1
3. else if vi =O then mark (r. Ai % ***3)=2***
4. elseif(vi!=O and vi!=255) then mark (r. Ai mod ***3)***
5. r. Ai =int(r. Ai)+ unitary(\);

Fig.4 (a) Watermark embedding algorithm

Alice i.e. data owner suspects that the Mallory i.e. intruder pirated from his relation R. Then he **uses** watermark detection to protect his ownership. The function of imshow (vi). displays the image pixels seriatim in the detection algorithm. Fig.4 (b) shows the watermark detection algorithm.

Input: Relation R(K.A0,A1,…..An) where K is the primary key

1. for each tuple r ϵ R do
2. if **(r.** Ai % 3)=1 then
3. elseif (r. **Ai** % 3)=2 then vi=0
4. else vi =(I. Ai -int(r. Ai))*255;
5. imshow(vi);

Fig.4 (b) Watermark detection algorithm

After performing some experiments on images, it has been verified that the watermarking algorithm designed by Zhi-Hao Zhang [7] is resilient to subset selection attack but not resilient to subset alteration, deletion and insertion attacks.

d) Watermark embedding and detection algorithm by Shehab:

M. Shehab et al [10] proposed a watermarking technique of relational databases which solves the optimization problem based on genetic algorithm and pattern search techniques. It is divided into two parts: Watermark encoding and decoding. Further, watermark encoding is done in three stages: Data set partitioning, watermark embedding and optimal threshold evaluation. While, watermark decoding is done in three stages such as data set partitioning, threshold based decoding and majority voting.

For each tuple r ϵ data set S, data partitioning algorithm (Fig.5 (a)) computes a message authenticated code

(MAC) which is secure and is given by Hash (r.p||k) %m.
Input: Data set D, secret key k, Number of partitions m
Output: Data partitions S0, S1, S2.., Sm-1

1. for each tuple r ϵ D,
2. partition(r)□ Hash(k||Hash(r.p||k))%m
3. insert r into Spartition(r)
4. return S0,S1,S2,…..Sm-1

Fig.5 (a) Data partitioning algorithm

M.Shehab et al. [10] proposed the watermark embedding algorithm by processing the bit encoding as an optimization problem. A genetic algorithm and a pattern search technique is used to solve the optimization problem. According to the application time and processing requirements, which optimization algorithm to use is decided. In single bit encoding, optimization problem is solved by maximizing or minimizing the hiding function which is based on single bit bi. If the bit bi is equal to 0, then the problem is considered as a minimization problem otherwise, it is considered to be a maximization problem.

Watermark embedding algorithm embeds a bit bi in the partition. Multiple embedding of watermark is supported for that number of bits l and number of partitions m are checked (i., e.l<<m). Watermark embedding algorithm is explained in fig.5 (b)

Input: Data set D, Watermark W= {b0, b1, b2, b3…..bl-1}, secret key ks, number of partitions m Output: Watermarked data set Dw, Optimal decoding threshold T*

1. for each partition Sk
2. i□ k mod l
3. Sk□ ecode_single_bit(bi,Sk,Xmax,Xmin)
4. Insert Sk into Dw
5. T*□ get_optimal_threshold(Xmax,Xmin)
6. return Dw,T*

Fig.5 (b) Watermark embedding algorithm

The bit decoding algorithm is based on decoding optimal threshold T* which minimizes the decoding error probability. Watermark detection algorithm extracts the watermark using the parameters secret key Ks, Number of partitions m, Threshold T, etc. Majority voting technique is used to detect the
watermark. The algorithm for watermark detection is described in fig.5(c).
Input: Watermarked data Dw, m, Ks, watermark length l

Output: Detected watermark Wd
1. //Initialize 2 arrays cnt [o, 1… l-1] and cnt1 [0, 1… l-1] to 0
2. // Get the data partitioned according to number of partitions, watermarked data, Secret key Ks
3. for i=0 to m-1 //for each partition
4. get partition Si
j = k % 1 //get ith bit for selected attribute
if i = 1 //check the bit value
cnt=cnt+1 //count the number of ones
else cnt1=cnt1+1 //count the number of zeroes
5. end partition Si
6. if cnt>cnt1 Wd[j]=1 //watermark bit is set to
else if cnt<cnt1 Wd[j]=0
else Wd[j] =X
7. l=l+1 //increment the watermark length
8. next partition
9. majority_voting(Wd)
10. return Wd

Fig.5(c) Watermark detection algorithm

This algorithm is resilient to various attacks such as deletion, insertion, alteration and not vulnerable to synchronization errors. It also minimizes the probability of decoding error because of optimal threshold [10]. Table 1 shows the Comparison between the above described techniques with respect to the different types of attacks and characteristics.

IV. CONCLUSION

In this paper, we reviewed four papers proposed by different authors on watermarking relational databases that embeds the watermark bits in the database set by partitioning it. Every author worked for the robustness of the technique so as to protect the system from various types of attacks. As technique proposed by M.Shehab [10] as compared to other techniques is more resilient to deletion, insertion and alteration attacks. It is not vulnerable to synchronization errors and minimizes the probability of decoding errors because of optimal threshold usage. Furthermore, the algorithm developed by M.Shehab can be made more secure by applying SHA algorithm for hashing as it is more secure though slower as compared to MD5 algorithm.

REFERENCES

[1] F. Petitcolas, R. Anderson, and M. Kuhn. Attacks on Copyright Marking Systems. Lecture Notes in Computer Science, 1525:218– 238, April 1998.

[2] M. Swanson, M. Kobayashi, and A. Tewfik. Multimedia Data-Embedding and Watermarking Technologies. Proceedings of the IEEE, 86:1064–1087, June 1998.

[3] I. Cox, J.Bloom, and M.Miller ,Digital Watermarking, Morgan Kaufmann, 2001.

[4] R. Agarwal and J Kieman, Watermarking relational databases In Proceedings of 28th International In Proceedings of 28th International Conference on very large databases, Hong Kong, China, 2002.

[5] D. Gross-Amblard .Query Preserving Watermarking of relational databases and XML documents, In PODS '03: Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pages 191–201. ACM Press, 2003.

[6] R. Sion, M. Atallah, and S. Prabhakar. Rights Protection for Relational Data. IEEE Transactions on Knowledge and Data Engineering, 16(6), June 2004.

[7] Zhi-Hao Zhang , Xiao-Ming jin, jain-Min wan , "Watermarking Relational Database Using Image" , 0-7803-8403-2/04/$20.00 @004 IEEE, Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, 26-29 August 2004.

[8] Y. Li, H. Guo, and S. Jajodia. Tamper Detection and Localization for Categorical Data Using Fragile Watermarks. In DRM '04: Proceedings of the 4th ACM Workshop on Digital Rights Management, pages 73–82. ACM Press, 2004.

[9] Y. Li, V. Swarup, and S. Jajodia. Fingerprinting Relational Databases: Schemes and Specialties. IEEE Transactions on Dependable and Secure Computing, 02(1):34–45, Jan-Mar 2005.

[10] Mohamed Shehab, Elisa Bertino and Arif Ghafoor, "Watermarking Relational Databases Using Optimization-Based Techniques", IEEE Transaction on Knowledge and Data engineering, VOL. 20, NO. 1, JANUARY 2008.

[11] Y. Li, H. Guo, and S. Jajodia. Tamper Detection and Localization for Categorical Data Using Fragile Watermarks. In *DRM '04: Proceedings of the 4th ACM Workshop on Digital Rights Management*, pages 73–82. ACM Press, 2004.

[12] B. Schneier. Applied Cryptography. John Wiley, 1996.

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

Characteristics	Proposed schemes			
	R. Agrawal and J. Kiernan(2002)	R. Sion, M. Atallah, and S. Prabhakar(2004)	Zhi-Hao Zhang , Xiao-Ming jin, jain-Min wan(2004)	Mohamed Shehab, Elisa Bertino and Arif Ghafoor(2008)
Watermark information	Bit pattern	Binary string	Database content	Database content
Cover type	Numeric	Categorial	Numeric	Numeric
Granularity type	Bit level	Bit level	Attribute value	Attribute value
Intent	Ownership proof	Ownership proof	Ownership proof, tamper detection	Ownership proof, tamper detection
Resilient to Insertion attack	No	No	No	Yes
Resilient to Deletion attack	No	Yes	No	Yes
Resilient to Alteration attack	No	Yes	No	Yes
Resilient to Synchronization errors	No	No	No	Yes

Table 1. Comparitive analysis of different watermarking techniques