IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 3, Ver. 1 (May – Jun. 2015), PP 54-65
www.iosrjournals.org

Image Based Relational Database Watermarking: A Survey

Sapna Prajapati[1,] Namita Tiwari[2]
([1]Department Of Computer Science Engineering NIT-Bhopal, India)
([2]Asst. Professor, Department Of Computer Science Engineering NIT-Bhopal, India)

Abstract: *In past few years relational databases watermarking has emerged a great topic for research because of increase in use of relational databases. The basic need for relational database watermarking is to prevent data from illegal access by providing copyright protection, tamper detection and maintaining integrity. To serve this purpose many database watermarking techniques have been addressed with different algorithm and cover type. With the use of watermarking unauthorized duplication and distribution can be detected. The watermarking scheme should be able to meet some important challenges: 1). The scheme should be capable enough to bear attacks so that the watermark doesn't get corrupted 2). The technique should be data preserving i.e. the error introduced into database as watermark information should not affect the value of data. The basic purpose of this paper is to review and analyse some existing image based database watermarking techniques.*
Keywords: *Relational database, database watermarking, copyright protection.*

I. Introduction

The rapid growth of internet access and exchanging digital data online has made duplication and distribution of the data easier than ever before. To overcome the challenges of forgery, false copyright claim, illegal redistribution of data, etc. watermarking has emerged as great technology. A watermark is information that is embedded into original data for ownership proof, copyright detection, traitor tracing etc. Since use of database has increased enormously, database watermarking has gain much of the attention in last few years. Just like encryption, typical watermarking will modify the ordinal data and will cause permanent distortion to the original ones and this is an issue if integrity in the database is utmost important requirement. Basic idea behind database watermarking is to change some of attributes value to another value, to such an extent so that the distortion is minimal and tolerable. [2]

The watermark approach basically involves two phases: firstly, inserting the watermark into the database, in which a private key K (known to the owner only) is used to embed the watermark information W into the original database. This watermarked database is then made publicly available for use. Secondly, verifying the watermark from the suspected database. The suspicious database is taken as input and by using the private key K (same as that used for watermark information embedding) the embedded watermark is extracted if present and then compared with the original watermark. Relational database watermarking approaches are application-specific rather than being generalized which means that there is no general algorithm that can be applied to all databases. Normally the characteristics of a watermarking algorithm are tied to the application it was actually designed for.

II. Literature Survey

"Watermarking" is the process of hiding information into another data; the hidden information does not need to have a relation with the data which is containing the information. [3] A **digital watermark** is a kind of mark or information inserted into any data such as audio or image. It is basically used to identify ownership of the data. Watermarks are generally used to verify the authenticity or integrity of the data or to show the identity of its owners.

2.1. Applications Of Watermarking [3]:

2.1.1 Copyright protection: Watermarking is initially born to provide the copyright protection for the media contents. The copyright information embedded into the data should survive all kinds of attacks either intentional or unintentional.

2.1.2 Transaction tracking or fingerprinting: It also requires the embedded watermark should be robust enough against malicious attacks.

2.1.3 Content Annotation: for this purpose the digital watermark is embedded to identify the producers and provide his contact address.

2.1.4 Content authentication: It prevents the attackers from tampering the digital contents or the data. This application is also known as fragile watermarking, which detects any form of changes even if one bit is converted.

2.2 Classification Of Watermarking Techniques [3]: The watermarking techniques proposed so far can be classified along various dimensions as follows:

2.2.1 Watermark Information: Different watermarking schemes embed different types of watermark information into the underlying data of the database for e.g. image, text etc.

2.2.2 Distortion: Watermarking schemes may be distortion-based or distortion free depending on whether the marking introduces any distortion to the data into which it is inserted.

2.2.3 Cover Type: Watermarking schemes can be classified based on the type of the cover (e.g. type of attributes) into which mark information is embedded.

2.2.4 Granularity Level: The watermarking can be performed by modifying or inserting information at bit level or higher level (e.g. character level or attribute level or tuple level).

2.2.5 Verifiability/ Detectability: The detection/verification process may be deterministic or probabilistic in nature, it can be performed blindly or non-blindly, it can be performed publicly (by anyone) or privately (by the owner only).

2.2.6 Intent of Marking: According to the purpose to be served watermarking schemes can be, namely, integrity and tamper detection, localization, ownership proof, traitor detection etc.

2.3 Characteristics Of Watermarked Database [2]: The desired characteristics for watermarking relational databases are:

2.3.1 Detectability: The watermark should be such it can be easily detected by the owner by examining thetuples of the suspicious database.

2.3.2 Robustness: The capability of the watermarking scheme should be capable to survive intentional (modifying, adding, deleting the tuples of database) as well as unintentional attacks (digital reproduction and photocopying). Even if the database is modified the watermark should be detectable.

2.3.3 Capacity: The watermarking scheme should be such that maximum watermark information can be embedded into the database.

2.3.4 Updatability: The watermark scheme should be such that the tuples of the relational database either inserted or deleted; the watermark can withstand the change.

2.4 Different Types Of Attacks On Database [3]: The different types of attacks which can be done to destroy the database are:

2.4.1 Benign Update: Modifying the original data without prior permission of owner. The watermarking should be such that even after the modification the watermark isn't lost.

2.4.2 Value Modification Attack: In this type of attack the watermark is destroyed by altering one or more bits in the watermarked database. For example an attempt of destroying watermark can be made by rounding all the values of numeric attributes.

2.4.3 Subset Attack: By deleting or updating some of the tuples (subset) of the database attacker may try to destroy the watermark

2.4.4 Superset Attack: Some new attributes or tuples are added into the database.

2.4.5 Subset Reverse Order Attack: By changing the order or position of the tuples the attacker try to erase or disturb the watermark

2.4.6 Brute Force Attack: It is hit and trail method to destroy the watermark from the database.

III. Analysis Of Different Image Based Relational Database Watermarking Schemes

The first method for relational database watermarking was proposed by Agrawal and Kiernan, in which the technique used, was to marks only numeric attributes with one-bit watermark scheme. [1] In image based watermarking, image is used as the watermark information. There are various image based watermarking schemes been suggested till now few of them are discussed here.

(i). **Algorithm proposed by Zhang Yong et al., NIU Xia-mu et al., WU Di et al., Zhao Liang et al., LI Jun-cao et al., XU Wei-jun et al.[11]**

In this algorithm, Zang used patchwork algorithm to choose random pairs of points(ax. by) of any image in spectral domain. They then increased the brightness at ax by 1 unit and then decreasing by's brightness and inserted the processed image information to the attributes which can tolerate some errors.

Watermark insertion algorithm:

1) Read the size of image(m×n)
2) From the left to the right and from the top to the bottom, read the value of each pixel's RGB from Image and obtain an ordered dataset, noted RGBValue(i,j) as embedding marks, x = 0…m×n-1, y = 0, 1,2(the value of y denoted R, G and B of each pixel correspondingly)
3) Watermarking attributes Ak(k = 1…t);
4) RDB.FIRST;
5) While not RDB.EOF do
6) Begin
7) For k:=1 to t do
8) if HASH(Key°PK°Ak)mod X = 0; //PKdenotes the primary key of the current tuple
9) Begin
10) s = HASH(Key°PK°Ak)mod (m×n);
11) r = HASH(Key°PK°Ak)mod 3;
12) Embedding the value of RGBValue(s,r) into the attribute Ak of the current tuple;
13) End
14) Else no operation;
15) RDB.next;
16) End;

Watermark extraction algorithm:

1) Select the embedded mark attributes from the suspicious relational database, noted Ak (k = 1…t);
2) Initialize a (x,y), aCount(x,y), the initialized value is 0, x =0…m×n-1, y =0, 1,2;
3) RDB.first;
4) While not RDB.EOF do
5) Begin
6) for k:=1 to t do
7) if HASH(Key°PK°Ak) mod X = 0 //PK denotes the primary key of the current tuple
8) begin
9) s = HASH(Key°PK°Ak) mod (m×n);
10) r = HASH(Key°PK°Ak)mod 3;
11) Extract the mark from the current attribute Ak of the current tuple, added into x(s,r);
12) aCount(s,r) = aCount(s,r)+1;
13) end;
14) else no operation;
15) RDB.next;
16) End;
17) for x:=0 to m×n-1 do
18) for y:=0 to 2 do
19) begin
20) if aCount(x,y)<>0 then
21) a(x,y)= x(x,y)/ aCount(x,y);
22) According to the values of a(x,y), generating an image, the size of the image is m×n and the R, G and B value of each pixel equals the values of a(x,y) corresponding;
23) end;

In the algorithm, among three attack modes, for the subset deleted attack, the method is the strongest robustness, and for the subset modified attack, the robustness of the method is common, and for the added attack, the robustness of the method is weaker.

(ii). Algorithm proposed by Zhi-Hao Zhang et al., Xiao-Ming Jin et al., Jian-Min Wan et al., De-Yi Li et al. [12]

In this algorithm, they consider a pixel of image as a little bit error in database. All pixels express an integrated copyright image. They assumed that the database contains float attribute, the number of tuples must he greater than the numbers of image pixels. They then watermarked a database of relation R whose schema is **R (K. Ao, A_1 ...A n),** and K is the primary key of database, and it is never marked. The pixel values of an image are **I(vo, vi..** . . . **.v,). "m"** is great less than **n. I: Ai** denotes the value of attribute **Ai** in tuple **r6R.** The relation R is divided into chunks of uniform size as same as the size of the image. Each chunk is regarded as a watermarking cell. Having lowered the planar image dimension, the pixel values of the image could he embed into the corresponding location of attributes. Pixel value and an attribute value of a tuple in relation are compared. According to the pixel values and the embedding rule, the corresponding attribute values can he marked.

Watermark Insertion Algorithm:

```
for each tuple re R do
if vi =255 then
mark (r. Ai mod 3)=1;
elseif vi =O then
mark (r. Ai mod 3)=2;
elseif(vi!=O and vi!=255) then
mark (r. Ai mod 3)+,
r. Ai =int(r. Ai) + unitary(\);
end;
end;
```

Watermark Detection Algorithm:

```
for each tuple re R do
if (r. Ai mod 3)=1 then
elseif (r. Ai mod 3)=2 then
else
vi 255;
VI 3;
vi =(I. Ai -int(r. Ai))*255;
imshow(vi);
end:
end;
```

The above algorithm suggested is easy and very effective. The algorithm is tested against various attacks such as: subset addition attack, sunset out off order attack, subset selection attack and subset alteration attack. It is found that the suggested algorithm is robust enough to different attacks.

(iii). Algorithm proposed by Jianhua Sun et al.,Zaihui Cao et al, Zhongyan Hu et al.[7]

This algorithm suggests a novel multiple watermarking scheme, which embeds two image as watermark information into relational database.

Watermark insertion algorithm:

```
// Insert a plain watermark W into relation R in form of 0,1 sequences, return marked R
// The parameters k$, L, , and x are all private to the owner.
1) calculate L-bit EMC E[L]=H(k$ concatenate W)
// L is the length of watermark
2) foreach tupler R do
3) t= H(k$ concatenate r.P_keyi)
4) if ( t mod x equals 0) then // mark this tuple
5) attribute_ index i = t mod x // mark attribute Ai
6) bit_index j = t mod x // mark j th bit
```

7) watermark index k= t mod L //use the k-th bit of 0,1 sequences to mark
8) mark_bit m=E k XOR (k mod 2) // get the value of marked bit
9) set the j-th least significant bit of r.Ai to m
10) if (not within_usability(new_data)) // check the availability
11) rollback
12) else commit
13) return R

Watermark Extraction Algorithm
// Algorithm to return a watermark M[ ] from relation R
// parameters k, L, p, q and u are also private to the owner.
1 for s=0 to L-1 do
2 DM[s]=' ' // initialize the detected mark code
3 count[s][0]=0, count[s][1]=0 // initialize counter
4 for each tuple r_R do
5 t= H1(k concatenate r.P)
6 if (t mod x equals 0) then // select this tuple
7 i = select_attribute() // mark i-th attribute
8 j = t mod q // select j-th bit
9 k= t mod L // mark the k-th bit of EMC
10 m= (j-th LSB of r.Ai) XOR (k mod 2)
11 count[k][m]=count[k][m]+1 // add the counter
12 for s=0 to L-1 // get the watermark
13 if (count[s][0]>=count[s][1]) // majority voting
14 then M[s]=0 else M[s]=1 //the final bit value

The algorithm is "blind" in that it requires neither the original data nor the watermark in order to detect a watermark in an object. The proposed method of watermarking relational databases using character images is correct, feasible, and robust. The approach used here is more intuitive, and it support easy watermark identification.

(iv). Algorithm proposed by Ashraf Odeh et al. and Ali Al-Haj et al. [5]

This algorithm suggests an efficient database watermarking algorithm based on inserting a binary image watermark in the 'time' attribute of database tuples. The 'Time' attributes exit by default, but in most applications they're not used. To be specific, the 'Date' attribute in databases is made of two fields: 'Date' and 'Time'. 'Time' field which is made of three fields: hours, minutes and seconds (HH:MM:SS). Hiding the binary information of the watermark in the seconds field (SS) should have the least effect on the usability of the database. Basic reason behind using time attribute is the large bit-capacity available for hiding the watermark information, and thus large watermarks can be easily hidden.

Watermark insertion algorithm:
1: Transfer the image into a flow of bits.
2: Group every 5 bits as a binary string,
3: Find the decimal equivalent of the string
4: Embed the decimal number in tuples selected
by the pre-defined key '**K**' as follows:
for each selected tuple do
for each selected 'Time' attribute do
if the 'SS' field of the 'time' mode **K = 0**
embed the decimal number
else Next attribute
end if
end loop
end loop

Watermark Extraction Algorithm
1. Extract the decimal number in tuples selected by the pre-defined key '**K**' as follows:
for each selected tuple do
for each selected 'Time' attribute do

if the 'SS' field of the 'time' mode **K = 0**
extract the decimal number
else Next attribute
end if
end loop
end loop
2: Find the binary equivalent of the extracte decimal number.
3: Group every 5 bits as a binary string.
4: Reconstruct the binary image watermark from the binary strings.

A major advantage of using the time-attribute in database watermarking is the large bit-capacity available to hide watermarks in the database. This is opposite to the more common bit-level database watermarking algorithms where watermark bits have limited potential bit-locations that can be used to hide them without being subjected to removal or destruction. The robustness of the proposed algorithm was verified against a number of database attacks such subset deletion, subset addition, subset alteration and subset selection attacks.

(v). Algorithm proposed by Zhongyan Hu et al., Zaihui Cao et al., Jianhua Sun et al. [13]

In this approach, an identification image is embedded into the relational data for representing the copyright information. It is assumed that some minor changes of some attributes values can be tolerated. Copyright information is embedded into these attributes. It is considered that the character image (copyright information) is a sequence of 0 and 1, the marks of 0 and 1 are small errors in the relational data. All the marks of 0 and 1 represent integrated copyright information. A character image which will convert as a sequence of 0 and 1 is to be embedded into relation R for the purpose of copyright protection.

Watermark insertion algorithm:
// Insert a plain watermark W into relation R in form of 0,1 sequences, return marked R
// The parameters k$, L, γ, and ѵ are all private to the owner.
1) calculate L-bit EMC E[L]=H(k$ concatenate W)
// L is the length of watermark
2) foreach tupler R do
3) t= H(k$ concatenate r.P_keyi)
4) if (t mod γ equals 0) then // mark this tuple
5) attribute_ index x = t mod ѵ // mark attribute Ax
6) bit_index y = t mod ξ // mark y th bit
7) watermark index k= t mod L //use the k-th bit of 0,1 sequences to mark
8) mark_bit m=E k XOR (k mod 2) // get the value of marked bit
9) set the y-th least significant bit of r.Ax to m
10) if (not within_usability(new_data)) // check the availability
11) rollback
12) else commit
13) return R

Watermark Extraction Algorithm:
// Algorithm to return a watermark M[ ] from relation R
// parameters k, L, α, ξ and ѵ are also private to the owner.
1 for s=0 to L-1 do
2 DM[s] ='' // initialize detected mark code
3 count[s][0]=0, count[s][1]=0 // initialize counter
4 for each tuple r_R do
5 t= H1 (k concatenate r.P)
6 if (t mod γ equals 0) then // select this tuple
7 x = select_attribute() // mark x-th attribute
8 y = t mod ξ // select y-th bit
9 k= t mod L // mark the k-th bit of EMC
10 m= (y-th LSB of r.Ai) XOR (k mod 2)
11 count[k][m]=count[k][m]+1 // add the counter
12 for s=0 to L-1 // get the watermark
13 if (count[s][0]>=count[s][1]) // majority voting
14 then M[s] =0 else M[s]=1 //the final bit value

The suggested algorithm is tested against 3 major attacks namely, subset addition, deletion and alteration attack. By the different attack models and the corresponding analysis, it is shown that the proposed method of watermarking relational databases using character image is correct, feasible, and robust.

(vi). Algorithm proposed by Hossein Moradian Sardroudi et al., Subariah Ibrahim et al. [4]

In this approach an image as watermark information is embedded into numerical attributes of relational database. It refers to image as 2-dimension matrix and tries to embed medium size image in small scale relation. By applying this method upper Correction Factor for image can be obtained. Furthermore at the phase of embedding watermark, this algorithm assures the minimum modification to original database without decreasing imperceptibility by minimizing data variation. In extracting process majority voting method is used to retrieve the correct watermark. Recovering step is added to the extracting process to improve this process. This step tries to recover extracted watermark and makes a guess for missing elements value, so that watermark can be detected even in a small subset of a database. Detecting the watermark does not require the original database and the watermark. Therefore the approach is blind.

Watermark insertion algorithm:

Inputs: M, KEY, R, #LSB, A, #MSB, F
BEGIN
1. WM = Get_Watermark(M); //transfer image pixel into a matrix
2. FOR all tuples in relation LOOP
3. vpk = Hash_Primary_Key(pk, KEY); //calculate virtual primary key
4. IF MOD(vpk, F) = 0 THEN
5. a# = MOD(vpk, #A) + 1; //select one attribute for embedding
6. b# = MOD(vpk, #LSB) + 1; //select one of LSB bits for embedding
7. IF Binary Length(a) > (2 * #LSB) THEN //check for attribute value tolerable for embedding
8. h= HASH(vpk, ImageHeight, 1); //compute mark height position
9. w= HASH(vpk, ImageWidth), 2); //compute mark width position
10. MSB_BIT=Select_MSB_BIT(vpk,a,#MSB);//select one of MSB bits
11. a'(b#) = WM(h)(w) XOR MSB_BIT; //XOR corresponding mark bit with selected MSB bit
12. a' = MinimizeVariation(a, a' , b#, #LSB); //minimize attribute value variation
13. Update Table(R, pk, a#, a');
14. END IF;
15. END IF;
16. END LOOP
17. IF NO ERROR THEN
18. COMMIT;
19. ELSE
20. ROLLBACK;
21. END IF;
END;

Watermark Extraction Algorithm:

Inputs: #MSB, Image_Height, KEY, R, A, #LSB, F, Image_Width.
BEGIN
1. FOR all tuples in relation LOOP
2. vpk = Hash_Primary_Key(pk, KEY); //calculate virtual primary key
3. IF MOD (vpk, F) = 0 THEN
4. a# = MOD(vpk, #A) + 1; //select one attribute for embedding
5. b#= MOD(vpk, #LSB) + 1;//select one of LSB bits for embedding
6. IF Binary Length(a) > (2*#LSB) THEN //check for attribute value tolerable for embedding
7. h = HASH(vpk, ImageHeight, 1);//compute mark height position
8. w = HASH(vpk, ImageWidth), 2);//compute mark width position
9. MSB_BIT=Select_MSB_BIT(vpk,a,#MSB);//select one of MSB bits
10. WM (h)(w) = a(b#) XOR MSB_BIT; //XOR corresponding extracted mark bit with selected MSB bit
11. END IF;
12. END IF;
13. END LOOP
14. IF NO ERROR THEN

15. M=Generate_Watermark(WM);//transfer matrix elements into image
16. Recover_Image(M);
17. ELSE
18. Message (Error_Code);
19. END IF;
END;

This paper illustrated the new approach for watermarking relational database. It presented a resilient watermarking technique for relational database that embeds image bits in the small size database as the watermark. It is robust to the important attacks

(vii). Algorithm proposed by Udai Pratap Rao et al., Dhiren R. Patel et al., Punitkumar M. Vikani et al. [6]

This paper suggests a technique for relational database watermarking which uses binary image as watermark information. The image bits are inserted into database, it represents the copyright information. The overall variation in the watermarked database is also minimized so that distortion is less.

Watermark insertion algorithm:
// the algorithm inserts a watermark information into the original database D, and return marked D.
// the parameters M, F,
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 D do
3. t = HASH(Ks concat r.P)
4. if (t mod F == 0) then // this tuple is available for marking
5. attribute_index x = t mod v // mark attribute Ax
6. x th bit
7. select row of an image a = (x * v) mod m
8. watermark_index k = t mod length(a) // it gives some bit position in ath row of watermark(image)
9. h = (HASH(t concat k(row value))) mod m // h is the position for selected mark bit from M
10. w = (HASH(t concat k(col value))) mod n // w is the position for selected mark bit from M
11. Replace the jth LSB of r.Ax with W[h][w] bit
12. Now, apply the minimize variation
13. Update D;
14. End loop;

Watermark extraction algorithm:
//two integer parameters are used, 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 D do
3. t = HASH(Ks concat r.P)
4. if(t mod F == 0) then // select this tuple
5. attribute_index x = t mod v // mark attribute Ax
6. bit_inde th bit
7. select row of an image a = (x * v) mod m
8. watermark_index k = t mod length(a) // it gives some bit position in ath 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 jth LSB
13. match_count++;
14. End if;
15. End loop;
16. if (match_count / total_count>= a)
17. Has watermark

The proposed technique minimizes the variation by inverting some bits of the watermarked attribute. That proposed technique is robust irrespective to the tuples order

(viii). Algorithm proposed by Ying Wang et al., Geng-Ming Zhu et al., Shao-Bo Zhang et al. [10]

This paper presents a watermarking algorithm which is based on numerical attribute in the relational databases. The watermark information is embedded by using the Arnold transformation and scrambling technology, the parity of the low decimal number of numeric attribute is modified, connected with some attribute in the physical storage space in the relational databases.

Watermark insertion algorithm:

1. Identify a number of secret information, the image watermark w, the user's watermarking key user _ key, the controlling factor ω, the embedded factor γ

$W_t = \{W_t(x) | W(x) \in \{0,1\}, 0 \le i \le m \times m\}$ //to get one dimensional vector, m×m is size of watermark image

3: identify the number of numeric attributes of the embedded watermarking as v, and arrange the primary key and numeric attributes by order in the relational database;

4. Set the searching function as find()//If the function value returns to be true , then the candidate attribute values $r_x.A_y$ can be embedded watermarking

5. id =hash(user _ key,P)

6. To determine the embedding watermarking tuples r_i according to the embedding watermarking factorγ, using the mod as the taking over function that satisfy if (id mod 1 /γ == 0);

7: To determine the watermarking attributes of the tuples $r_i.A_j$ that is candidate attribute value $r_i.A_j$ according to the remainder of (id mod v)

8: To determine the low position of the decimal j d of the attribute value $r_x.A_y$ according to the control factor ω , and seek the candidate parity bits of $r_x.d_u$, that is par (x) = $r_x.d_y$ mod 2, par(x)= {0,1};

9. sta(x) Statisticw(r_x .P)mod 2 and sta(x) = {0,1};

10: To obtain x = id mod l according to id of the tuples r_x, and determine the corresponding position x of the watermarking according to the remainder;

11: wvalue = addmark (w (x), sta (x), par (x));

12: Then to modify the $r_x.d_y$ value as wvalue;

13: Return to Setp6, until all the one-dimensional watermarking vectors w (x) can be embedded in the candidate attributes;

Watermark extraction algorithm:

1. Identify a number of secret information: the image watermark w, the users' watermarking key, user _ key, control factorω , embedded factor γ ;
2. Arrange the primary key in the relational database and numeric attributes by order;
3. Search the relational database in detection, and find out the candidate tuples attributes $r_i.A_j$ ($1 \le x \le n$, $1 \le y \le v$) x y that can be embedded watermarking within the permissible data error δ_y. Use find () in embedding operation to complete the process.
4. To restore the marks id of the numeric tuples, that is to recalculate the hash value.
5. To find out the tuples r_x that is embedded watermarking according to the watermarking embedding factor γ. That is to select the candidate tuple attribute values $r_x.A_y$ and identify the tuple watermarking embedding position as long as if (id mod 1 == 0), and the watermark embedding attributes Ay is determined;
6. Determine the low position in decimal of the embedding watermarking attributes $r_x.A_y$ according to the control factor ω ;
7. w(x)=holdmark(par(x) , sta(y))
8. Determine the watermarking position x corresponding to the watermarking signal w (x) according to x = id mod l;
9. Back to Step5 to find the next relational databasetuple that is embedded watermarking, until all the embedded watermark tuples are found out.
10. To do the majority of the election on the watermarking signal w (x) in the same extracted watermarking position x;
11. After get a one-dimensional watermarking vector – W'_t, $W'_t = \{ W'_t(x) \mid W'(x) \in \{0,1\}, 0 \le x \le m \times m \}$ to map W'_t into a two dimensional Matrix W' ,$W' = \{w'(x, y) | w'(x, y) \in \{0,1\}, 1 \le x \le m, 1 \le y \le m\}$according to the line scan sequence, which is the restored watermarking signal;
12. To express the black and white pixel values of

W' with the actual value and to restore the binary image D' from the database, and then get the original image from

T − C times Arnold transforming and restoring.

The above algorithm, image has the stronger robustness. By taking double filtering mechanism to determine the location of the watermarking embedded can make the watermark embedding much safer and more subtle. By modifying the parity of the numerical attribute of the low value of to embed watermarking, rather

than directly operating against the physical storage of data-bit, the range of data modification and amount of computing is smaller, and the operation is much easier. The algorithm meets the synchronization requirements of the database dynamically update it is blind watermarking algorithm.

(ix) Algorithm proposed by Yi Liu et al., Juan Wang et al. [9]

The algorithm scans candidate attributes where watermark can be embedded and then conduct subset segmentation and rearrangement, and then DWT transformation is performed to the data subsets and the scrambled watermark image respectively. The compressed low-frequency part of the watermark is embedded into the High-frequency part of the data set to achieve data fusion.

Watermark insertion algorithm:

1) Perform K times of Arnold scrambling to a M×N binary image W and W becomes W'. W' = {W'(x,yj)|0≤x≤M, 0≤y≤N}, and preserve scrambling times k as the key. Execute the three level wavelet decomposition to the scrambled image and get the wavelet coefficients matrix of the third-level lowfrequency subblock LL3. The mean of the coefficient matrix is calculated and labelled as Avg and save it as a key, and then each coefficient of the matrix minus Avg to come in for the compressed low-frequency subblock LL3'.
2) Utilizing the algorithm mentioned in section 3.1.2 screen the candidate attributes that can be embedded watermark and using label algorithm mark the Aj, IDrx.Ajy= hash(Key, P, Ay).
3) Grouping the data into λ packets according to the values that come from the labeled ID mod λ(ID % λ). Group (k) =IDrx.Ay Mod λ{0≤k≤λ-1}, where λ is the repeating times of watermark embedding and its value can be set in accordance with the specific relationship and the watermark information.
4) Sort the data in each packet according to the ID value, there are total M × N bits in each packet (the watermark length) and fill them with 0 if void bits appeared.
5) Perform three level wavelet transformation to each Group (k)($0 \le k \le \lambda$-1) respectively and get the third level high-frequency subblock 3 HHk ($0 \le k \le \lambda - 1$) .
6) By way of adding embed the compressed watermark low-frequency coefficients LL3' into the high-frequency subblocks HH3'(1≤k≤λ-1) of the host data. The specific embedding mode as follows: HH3 k =HH3 k(1+αLL3'), (0≤k≤λ-1) where α represents the intensity factor of high-frequency subband watermark embedding.
7) Conduct the inverse transformation to the watermarked coefficients and get the watermark contained data.

Watermark extraction algorithm:

Watermark detection is the reverse process of embedding, but needs to calculate the similarity between the extracted watermark signal W* with the original watermark W. If the correlation coefficient Sim is greater than the threshold T, then the watermark exists and not exists otherwise.

$$Sim = \frac{W^T}{\sqrt{(W)T\ W}} \quad \frac{*\quad W^*}{\sqrt{(W*)T\ W*}}, \quad 0 < Sim > 1$$

Algorithm using wavelet transformation skill embeds the compressed "small" watermark to a relative "large" host database. Not only has little influence on the database but also greatly reduces the probability of watermark damage, which effectively overcomes the defect that spatial algorithms are usually produce morbid results. The experimental results reveal that the complexity of algorithm is simple, has perfect invisibility and strong resistance to varied attacks, and especially enjoys sturdy immunity for subset modification and subset deletion.

(x) Algorithm proposed by Ramani Sagar V. [8]

In the algorithm, ownership verification of a database by inserting an imperceptive watermark in such a way to provide robustness and security against attempts to remove the watermark. To prove ownership of the database watermarking is done using image. Image is converted into the row bits and row bits is encrypted using MD5 security algorithm .This row bits will be embed into the database attribute in terms of watermark.

Watermark insertion algorithm:

1) Take a path for a given database to be watermarked
2) Based on database count no of tuples & no of columns to be watermarked
3) Obtain secret key Sk for MD5 hashing
4) For MD5 hashing obtain primary key P and concatenate with secret key Sk

5) Using MD5 algorithm obtain hash value using combination of P+Sk
6) Mark the tuple based on the hash value If(H_hashvalue mod
F= 0) then mark the tuple Where F= database partitioning value which will be private to the database owner
7) Obtain **attribute indexA** based on (H_hashvalue mod V =0)
Where V =No of columns of the database
8) Obtain **bit indexB** which will be marking a particular bit of attribute based on(H_hashvalue mod ξ= 0) where ξ=private to the owner
9) Select the image bit procedure for the given image bases on hash value generated by the MD5

- Select row of image based on (A*V) mod H where H = image height
- Select column of image based on H_hashvalue mod W where W = image width
- Convert row value into decimal value
- Convert column value into decimal value
- Obtain height & width for marked bit from the image based on hashing algo MD5 concatenated value of row
- Select bit of image for marking based on marked bits

10) Select field from database based on hash function
11) Covert attribute field into the binary field
12) Replace in LSB of binary value of attribute with selected image bit
13) Replace new attribute value in the database

Watermark extraction algorithm:
1) Take a path for a given database
2) Initially totalcount=matchcount=0
3) Obtain MD5 hashing obtain hash value using P+Sk
4) Detect the tuple based on the following condition

- IF (H_hashvalues mod F=0)then mark the tuple
  Where F=data partitioning value which will be private to the owner
- Obtain attribute index i = (H_hashvalue mod V=0) where V =No of columns of the database
- Obtain bit index B which will be marking a particular bit of the attribute based on (H_hashvalue mod ξ=0) where ξ=private to owner
- If(bit index B<=length of selected field) then

totalcount++
matchcount=matchcount+match(selected field ,bit index, selected bit)

In the given algorithm, each tuple in a table is independently processed; therefore, the scheme is particularly efficient for tuple oriented database operations. This watermarking scheme is robust for two attacks which are attribute addition attacks and subset reverse order attack.

IV. Conclusion

In this paper we survey the current state-of-the-art of different image based watermarking techniques for relational databases. We can insert an image as watermark in our database in various ways. It should be noted that all the techniques of watermarking database can only be applied for real time copyright protection of the database. These methods cannot prevent piracy or illegal copying of database. Methods of Watermarking of relational databases are basically divided into two types, Distortion Based watermarking and Distortion Free watermarking. Watermarking of Relational Database implemented by Image Based Watermarking is actually Distortion Based Watermarking. After studying and analysing various research papers and approaches it can assured that image based watermarking can be efficiently used for security and authentication of database. Image based database watermarking is also useful for ownership protection. This technique will be useful for fraud and temper detection also.

References

[1]. Agrawal, R., & Kiernan, J., 2002. Watermarking relational databases. In Proceedings of the 28th very large data bases VLDB conference, Hong Kong, China (Vol. 28, pp. 155–166).
[2]. Dwivedi, A. K., Sharma, B. K., & Vyas, A. K. (2014). Watermarking Techniques for Ownership Protection of Relational Databases, 4(1), 368–375.

[3]. Halder, R., Pal, S., & Cortesi, A. (2010). Watermarking Techniques for Relational Databases : Survey , Classification and Comparison, 16(21), 3164–3190
[4]. Hossein Moradian Sardroudi, Subariah Ibrahim, 2010. A New Approach for Relational Database Watermarking Using Image, Universiti Teknologi Malaysia (UTM)".
[5]. Odeh, A., & Al-Haj, A. (2008). Watermarking relational database systems. First International Conference on the Applications of Digital Information and Web Technologies (ICADIWT), 270–274. doi:10.1109/ICADIWT.2008.4664357
[6]. Rao, U. P., Patel, D. R., & Vikani, P. M. (2012). Relational Database Watermarking for Ownership Protection. Procedia Technology, 6, 988–995. doi:10.1016/j.protcy.2012.10.120
[7]. Sun, J. S. J., Cao, Z. C. Z., & Hu, Z. H. Z. (2008). Multiple Watermarking Relational Databases Using Image. 2008 International Conference on MultiMedia and Information Technology, 373–376. doi:10.1109/MMIT.2008.211
[8]. V Ramani (2013). Watermark based Copyright Protection for Relational Database, International Journal of Computer Applications (0975 – 8887) Volume 78 – No.2, pp. 22–28.
[9]. Wang, J. , 2013. DWT Based Blind Watermark in Relational Database, Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013), 2816–2819.
[10]. Wang, Y., Zhu, G.-M., & Zhang, S.-B. (2012). Research on the Watermarking Algorithm Based on Numerical Attribute in the Relational Database. International Conference on Computer Science and Electronics Engineering, 363–367. doi:10.1109/ICCSEE.2012.363
[11]. 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.
[12]. 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.
[13]. 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.