Use of Watermark in Educational Administration System

Yi Liu
Office of Academic Affairs
China West Normal University
NanChong, China
liu4fire@126.com

Siyu Lai
Department of Medical Image
North Sichuan Medical College
NanChong, China
lsy791211@126.com

***Abstract*—With the widespread use of watermark in video, audio and image, propose a relational database based watermark embedding and extraction method. The algorithm utilizes the single—multiple mapping approach to plug in watermark, which scatter the distribution of watermark information and enhance the anti-attack ability of the algorithm. The experiments show that the method enjoys strong robustness to such attacks as tuples resorting, subset selection and so on.**

Keywords- digital watermark, relation database, mapping, bit stream, multiple embedding

I. INTRODUCTION

Relationship database (RDB) has been seen as the most important part in high educational academic management department, which is confronting with an increasingly numbers of threatens as illegal login and malicious tampering. In recent years, there have been raised a new and efficient digital technology for copyright protection and data security maintenance in the world --- Digital Watermarking. The watermarking is an invisible label that hides in digital products and can be extracted or detected by digital embedding methods. It can be used to detect the ownership of the original author, as well as the evidence to identify, prosecute the illegal infringement.

Digital watermark must meet two rules, one is that the embedded watermark cannot damage the validity of the original data, that is to say, the original data is still applicable in the range of application. The other says that attackers are hard to remove or change the watermark without damaging the data. We can see that how to keep validity of the data are the key problems in watermark handling while the value and type of the data and where the data will be used are closely related. For example, the watermarked software must guarantee the conformity of the result as that of the original one and the textual watermark should ensure the consistency of the semantics before and after watermark embedment, and so on.

At present, the abundant researches have been made in the fields of software, text and multimedia(image, audio , video) and achieved systemic results, but the research on RDB are just begin. Generally speaking, watermark should be plunged into the least important area of the multimedia system, which allows the variation is insensitive to human visual and auditory system. Due to the insensitivity of the perception of people, there are many choices to select to embed watermark on condition that do not damage the validity of the data (refers to the range tolerable). And the data in RDB is more strict with validity, micro change is possible to destroy the validity, what's more, as take the semantics of the data into account (for instance, micro alteration may not damage the data but will breaking the semantics when score 59 has been changed to 60 in score processing unit), which make it more difficult to embed watermark in RDB than software and multimedia.

This article propose a new RDB watermarking algorithm based on the analysis of various watermarking algorithm which uses a group multiple embedding and multi-unit mapping method that enjoys robustness to a number of attacks and is more convenient to detect watermark than the usual unit embedding.

II. WATERMARK ALGORITHM

A. Research Foundation

A well watermarking algorithm should have the following attributes:

1. Robustness: Algorithm should robust enough to prevent attackers from easily erasing watermark.

2. Update ability: The database may need updating, insertion/deletion constantly or modify the value of a property in any tuple, which does not undermine the overall watermark. We need only to recalculate the watermark of inserted or modified tuple.

3. Imperceptibility: The changes caused by embedding of watermark should not undermine the validity of data. Besides, such frequently used statistics as averages et al. also should not has the larger change for the embedding of watermark.

4. Blind detection: Be capable of detecting watermark directly from the watermarked database neither requires the original database nor the embedded watermark in advance. By doing so, the watermark in illegal copies of the database can be detected without relying on the original database that may have been updated.

The different algorithms have been proposed by Rakesh Agrawal and Radusion, both are defective. Radusion embed

Published by Atlantis Press, Paris, France.

watermark utilizing the distribution of property in each data group in algorithm [1], which enjoyed well robustness. But as the data group is altered when update occurs, the scrambled data distribution that is contribute to damage the watermark and need to embed watermark again in this group after updating. So the algorithm does not have a good updatable ability. The literature [2] using the order of properties in the database to embed watermark. But it may lead to a detection failure when the order was changed or the properties were deleted and must rely on the original database. Hence, the algorithm does not have a good detectability and blind ability, the proposed algorithm put up good performance in all above mentioned aspects.

B. Watermark Embedding

Suppose the database is *R* with *n* tuples and the watermark is *M*. It is not true that each attribute in database can be embedded watermark, usually select those attributes with weak validities and semantic constraints. Given that the primary key in database is *P_key*, the properties used to embed watermark are $A_1,A_2,\ldots,A_k$ and the length of the *LIB* (least important bits) can be plug in watermark is *S*.

Figure.1 shows the flow chat of watermark embedding.

The main steps of the embedding are listed below:

1. Transform the watermark *M* to bit stream and the length is *w*.

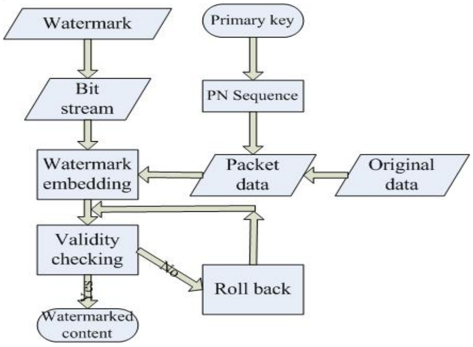

Figure 1. Watermark Embedding Diagram

2. Calculate the value of primary key in each tuple according to Hash function [5].

$id=Hash(k_s,P\_key,k_s)$

Where k_s is the private key to the database owner.

As the primary key of each tuple are different, so the corresponding *id* also differ from each other in the light of the Hash function property.

3. Assort all tuples into $w\times r$ groups according to various remainders produced by $w\times r$ divided by all *id* and *r* is the number of times that embed the watermark.

4. Embed the i_{th} watermark bit *M[i]* into attributes $A_1,A_2,\ldots,A_k$ of groups *i*, $w\times 1+i, \ldots, w\times(r-1)+i$.

5. Perform validity checking to watermarked information and recover data if the result exceeds the limit.

The Watermark embedding process adopt a single---multiple bits mapping method, that is, map one watermark bit to *m* bits of database and the concrete value of *m* is determined by the owners. We let *m* is 3 and map 0, 1 to 011, 110 respectively.

The detailed algorithm is listed below:

w represents the number of bits of watermark *M*, *k* is the key to Hash function and *r* stands for the number of times of watermark embedding repetition. The single bit maps to *m* bits, 0 corresponds to M_0 and 1 corresponds to M_1. And the number of watermarked tuples accounts for $1/\lambda$ of the total tuples. *s* is the order number of *LIB*, $w, k_s, r, m, M_0, M_1, \lambda, s$ are privately possessed by the owners.

```
watermark(attribute $A$, $P\_key$, $k_s$, $M$, $\lambda$, $s$)
// Embed watermark in attribute A
(a) $M[w]=chartobit(M)$; //transform watermark to bit stream
(b) for each tuple
id=Hash($k_s$, P_key, $k_s$); //Hash the primary key of each tuple
(c) $subset_i$←classify all tuples to $w\times r$ subsets according the different remainders when id is divided by $w\times r$;
(d) for($i=0$; $i<w\times r$; $i$++){
      encode($subset_i$, $M[i\%w]$);
         If check_validity($subset_i$) // check the validity
         commit; // submit if success
       else rollback; }
encode($subset_i$, bit $b$)
for each tuple in subset
   if $[id/(w\times r)]$ mod λ==0{ // embed watermark in this tuple
      bit_index $j$=$[id/(w\times r)]$ mod $s$;
   // embed watermark from the jth bit in LIB
if b==0
   set $j_{th}$, $[(j+1) \bmod \lambda]_{th}$ …$[(j+m) \bmod \lambda]$th to $M_0$;
else
set $j_{th}$, $[(j+1) \bmod \lambda]_{th}$ …$[(j+m) \bmod \lambda]_{th}$ to $M_1$;}
```

C. Watermark Extraction

Figure.2 shows the flow chat of watermark extraction.

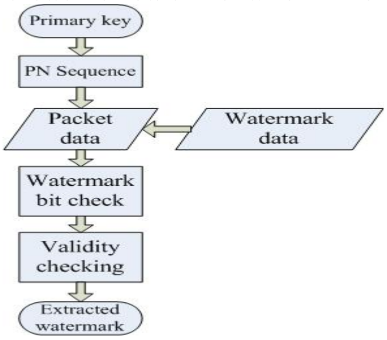

Figure 2. Watermark Extraction Diagram

1. Calculate the *id* of each tuple based on the Hash function;

2. Categorize the tuples into $w\times r$ groups on the basis of the varied remainders generated from id divide by $w\times r$;

3. Detect the watermark in each group;

4. Apply multiple selection method to watermarks in groups i, $w\times 1+i$, $w\times 2+i$ ...$w\times(r-1)+i$ and get the i_{th} bit $M[i]$,i=0,1,2…$w-1$.

Published by Atlantis Press, Paris, France.

The main steps of the extraction are listed below:

w, k_s, r, m, M_0, M_1, λ, s are similar with those of embedding algorithm. *mark1[w×r]* save watermarks extracted from *w×r* groups and *mark2[]* saves watermarks extracted from tuple of single group.

```
detect($P\_key, k_s, \lambda, s$)
(a) for each tuple
    id=Hash(ks, P_key, ks); // Hash the primary key of tuples
(b) $subset_i$←classify all tuples to $w\times r$ subsets according the different remainders when id is divided by $w\times r$;
(c) for($i=0; i<w\times r; i++$)
    $mark1[i]=decode(subset_i)$;
// Extracted watermark from groups and store to mark1[]
(d) for($i=0; i<w; i++$)
  $M[i]$=majority_vote($mark1[i],mark1[i+w],mark1[i+w\times r]$)
…$mark1[i+w\times(r-1)]$]) // Using the multiple selection method
decode($subset_i$)
    for each tuple in subset
        if $[id/(w\times r)] \bmod \lambda==0$ {
            bit_index $j=[id/(w\times r)] \bmod s$
// embed watermark from the $j_{th}$ bit in LIB
if $j_{th}$, $[(j+1) \bmod \lambda]_{th}$ …$[(j+m) \bmod \lambda]_{th}$ equals $M_0$
    mark2[ ]←0;  // if 0 then store it in mark2
if $j_{th}$, $[(j+1) \bmod \lambda]_{th}$ …$[(j+m) \bmod \lambda]_{th}$ equals $M_1$
mark2[ ]←1;  // store 1 to mark2
else break; }// No watermark in the tuple, go next one
        return majority_(each element in mark2[]);
```

III. Algorithm Analysis

To ensure the watermark distribution is average in the algorithm, we select tuple that should be embedded watermark depend on whether the *id/(w×r)* divided exactly by λ. Rakesh Agrawal selected tuple in document [3] based on whether *id* is divided exactly by λ but in this paper it does not work. Because we grouping the units on the basis of the remainder produced by *id/(w×r)*, so the varied *id* in one group have the same relation with *w×r*. If *w×r* and λ meet certain rules, then it will be possible that all the tuples in this group would be embedded watermark or not. All tuples in Group 0 divided exactly by λ, they all should be embedded watermark and if the remainders of *id/(w×r)* are 1, the above mentioned process should not happen when *(w×r)* is integral multiple of *r* . But the quotients of *id/(w×r)* have little relation with *(w×r)* and remainders of *id/(w×r)* are well-distributed, which make all groups are embedded watermark components.

The algorithm adapts multiple--group and single-multiple mapping embedding method that has the following advantages:

1. The multiple group method enables the watermark distribute broadly and embed the meaningful watermark to every group, so that the extracted watermark is more stringent. This approach enhanced the robustness and would not be influenced by the disappearance of the watermark in a tuple, improved the likelihood of success on watermark extraction significantly. Being the use of multiple—embedding and majority—selection extraction, which could correct bit extract errors caused by attacks and guarantees the accuracy of the resulting watermark.

There are several ways could be used to group tuples, one is that fix the number of tuples in each group after sorting the *id* of all tuples. Method used in this paper perform grouping strategy according to the remainder of *id* divided by a certain number. Each bit of the watermark information is corresponding to a fixed group which identify the fact that watermark from a certain group and the bits in overall watermark *M* are correspondent. And ensures the combination of watermarks extracted from groups is meaningful information, the algorithm is blind.

2. The single--multiple bits mapping method change the number of bits in the range of data validation and enhance the accuracy of the watermark extraction.

The single bit embedding method tests a bit on the tuple, it can only be 0 or 1 and the test man would think that the watermark is 0 or 1 while the tuple may simply not be added the watermark. It is the one that attacker trying to disrupt the watermark, he get a 50% more chance to make the bit on the position not equal to the embedded one and thus affecting the extraction.

The main idea of the method is mapping one bit to *m* bits. *m* bits compose 2^m different values, only two of which(M_0,M_1) are corresponding to embedded 0 or 1 and all other values are invalid. Only when *bit(m)* is equal to M_0 or M_1, we consider it is the true watermark and it is added or modified by attackers should be discard, which reduce the possibility of watermark breaking. He only have a $1/2^{m-1}$ chance to make test man believe watermark exist in data and $1/2^m$ opportunity to make the *bit(m)* identical to original one. Take the later experiment as an example, *m*=3, M_0 =011 and M_1=110, thus the probability that test man fell there are watermarks in data is $2/2^3$ and the probability of these 3 bits are identical to original ones is 1/ 8.

IV. Experiment And Realization

The experiment is developed under the environment of windows sever 2003 with 2.0 GHz CPU, 256MB RAM, Sun JDK 1.4.2. Connect SQL Server 2005 database using JDBC, the watermark is "water" and the repeat times are 5. The number of tuples in database is 10000 and set the LIB 5, map 1 bit to three bits and figure out the value of *id* in each tuple by using SHAI[4] approach.

The following list the major attacks and analyzes the ability of the algorithm against the these attacks

A. Subset Appending[6]

The single—multiple bit method excludes tuples that the corresponding *m* bits are not equal to M_0 and M_1 which reduces the interference posed by the joining of attacks and enjoys good robustness of subset appending.

Experiments show that it is still possible to detect about 83% of the watermark bits when λ=100 and the subset increases 100% and about 87% when λ=10 and subset increases 10%.

As we can see from Figure 3, λ is smaller the effect of watermark extraction is better when increasing the same number of subsets. And there is distinguished difference between the two actions when increasing 60% of the subset.

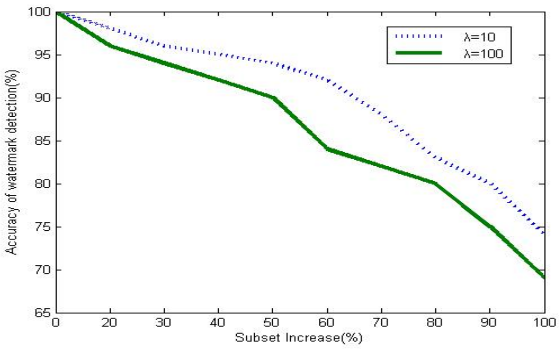

Figure 3. The Subset Appending based Watermark Detection

B. *Subset Selection*[7]

The grouping method scatters the effect of subset selection to varied groups. There must be another 50% remaining when selecting 50% of the total number of tuples, which prevent the loss of the watermark effectively.

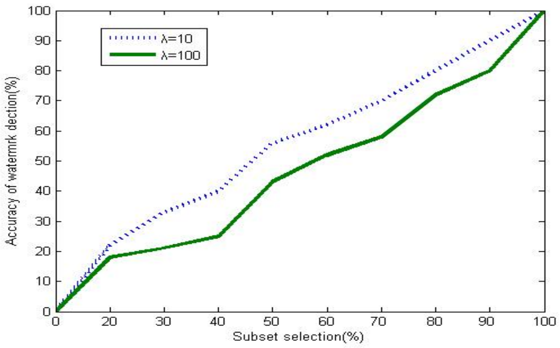

Figure 4. The Subset Selection based Watermark Detection

As we can see from Figure 4:

The difference of watermark extraction is small when λ=10 or λ=100, that is, selecting the minority or majority of the subsets. And subset selection affects watermark extraction clearly when λ lies in the range of 20% to 80%. The more subsets the attackers selected the more information extracted in watermark detection process.

C. *Subset Alteration*[8]

The single—multiple bits method is also defensive against the subset alteration attacks. The approach discards those tuples that was watermarked and altered by attackers which result m bits are not equal to M_0 or M_1 and decrease the disturbance of attacks.

Just like the contrasts in Figure 5, λ=100 can detect 60% of the watermark and λ=10 can detect 67% of the watermark when the subset alteration rate reach to 50%.

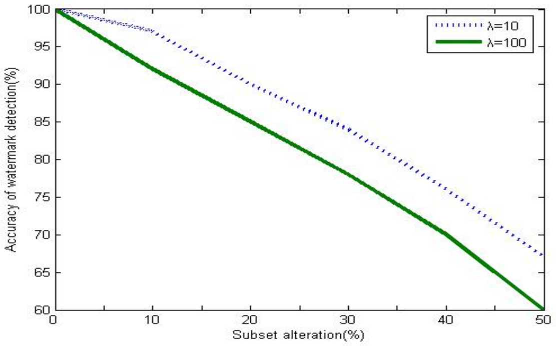

Figure 5. The Subset Alteration based Watermark Detection

V. Conclusion

Propose a new algorithm that adapts multiple grouping and single—multiple bits mapping method on the basis of the full study of the varied database watermarking method, which improves the possibility of watermark successful extraction. The experiments show that the algorithm enjoys good robustness to most of the attacks. Our future work is embedding watermark in non numeric attributes of database and design the attack model in this new field.

References

[1] Radu Sion, Mikhail Atallah, Sunil Prabhakar. Rights Protection for relational data. In Proceedings of the ACM Special Interest Group on Management of Data Conference SIGMOD, 2003.

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

[3] Y.G. Li, Vipin Swarup and Sushil Jajodia. A robust watermarking scheme for relational data. In Proceedings of the Workshop on Information Technology and Systems (WITS), 195-200, 2003.

[4] D.Y. Li, K.C. Di, D.R. Li et al. Mining Association Rules with Linguistic Cloud Models. Journal of Software, 2000, 11(2): 143-158.

[5] D Eastlake, P Jones. US Secure Hash Algorithm, Network Working Group. Request for Comments: 3147, Category: Informational, 2001, 9.

[6] Radu Sion, Mikhail Atallah. On Watermarking Semi-Structures

[7] D.Y. Li, H.J. Meng and X.M. Shi. Membership Clouds and Membership Cloud Generators. Journal of Computer Research and Development, 1995, 32(6): 15-20.

[8] Y. Wang, G.M. Zhu and Q.F. Nian. Study and Analysis on Digital Watermark for Relational Database. Journal of Hunan City University (Natural Science Edition), 2008 2.