International Journal of Security and Its Applications
Vol. 7, No. 2, March, 2013

A Novel Relational Database Watermarking Algorithm Based on Clustering and Polar Angle Expansion

Zhiyong Li[1], Junmin Liu[1] and Weicheng Tao[1]

[1] *College of Information Science and Engineering,*
Hunan University, Changsha, China
zhiyong.li@hnu.edu.cn, 326860194@qq.com, weishens1985@hotmail.com

Abstract

Digital watermarking has been widely applied to relational database for ownership protection and information hiding. But robustness and reversibility are two key challenges due to the frequently database maintaining operators on those tuples. This paper proposes a novel relational database watermarking scheme based on a fast and stable clustering method on database tuples, which adopts Mahalanobis distance as the similarity measurement. Before the process of watermark embedding and detecting, the databases tuples are adaptively clustered into groups according to the length of binary watermark. Moreover the watermark segments are respectively embedded into or detected from those groups according to the numeric field's Lowest Significant Bit (LSB) and polar angle expansion. The majority decision strategy is used to determine the value of watermark bit in blind detection process. The experiment results indicate that the proposed watermarking scheme has higher robustness and reversibility under blind detection against the database maintaining operators.

Keywords: *Database watermarking, robustness, reversibility, blind detection, tuples clustering, polar angle expansion*

1. Introduction

Nowadays cryptography is the most important technique to protect data and digital works. However, cryptography cannot provide effective ways to protect digital content from illegal copying, spreading again and malicious tampering after being decrypted. Digital watermarking is developed in recent years as a potential information security key technology, which can determine the ownership or originality of digital content by embedding perceivable or unperceivable information in digital works [1]. It can make up for the deficiencies of cryptography and has better characteristics on security, invisibility and robustness [2].

Digital watermarking technique has been successively applied to protect the multimedia works and software products. Similarly, database watermarking has been proposed on large database security-control. However, there are some differences between relational database and multimedia data [3]. Firstly, a relational database table consists of many attributes and tuples, but there is no certain ordering between tuples or attributes of a relation table. Secondly, database maintaining operators could frequently change those tuples unlikely other type of multimedia object. Moreover, database tuples processing rely on logical set operational language such as SQL. So database watermarking should also have the ability of real-time update and blind detection and cannot directly adopt those multimedia watermarking method. It is more difficult to ensure the robustness and reversibility of database watermarking.

In recent years, scholars have carried out extensive research on database watermarking. The groundbreaking study in this area was conducted by R. Agrawal and R. Sion in 2002 [4, 5]. In 2003, X.M. Niu proposed that a meaningful string could be inserted into relational database as the watermark [6]. Y. J. Li raised a method of inserting watermark by changing the order of relational data index [7], and it does not change the physical location or data value to impair its use. Whereas the index is additional information outside of relational data content, watermark information could be completely lost if index of relational table is reestablished or deleted. Y. Zhang converted image information into watermark cloud droplets according to D.Y. Li's cloud model idea, and then embedded it into relational data [8]. When being extracted, the cloud droplet should be compared with original copyright image. Moreover, Y. Zhang put forward a reversible watermarking method for relational database [9] that took the differences at the end of relational data and expanded it using wavelet transformation, then embedded watermark information. G. Gupta utilized difference expansion and Lowest-Effective-Bit on integers to achieve embedding and blind detection of watermark, but the method is only used for integer data that make it not universal [10]. Many other watermark workers also make a lot of efforts to promote the development of database watermarking [11-22], yet there are still many shortcomings in current study, they could be included into two aspects: On one hand the watermark robustness is too weak to resist various conventional database operations and illegal watermark attacks, such as selection, addition, modification and so on, on the other hand the original relation cannot be restored from the watermarked relation. As a result, how to improve the robustness and reversibility of database watermarking is a very difficult and significant work.

To improve the robustness and reversibility of database watermarking, the paper puts forward an adaptive relational database watermarking scheme based on clustering and polar angle expansion. The rest of the paper is organized as follows: Section 2 introduces the basic framework and strategy of the proposed scheme for relational databases; Section 3 provides the details of the algorithm; Section 4 presents the simulation experiment and analysis, and Section 5 concludes the paper.

2. Framework and Strategy

Allowing for the disorderliness of tuples and attributes, insufficient redundant space of database, along with weak robustness of the general database watermarking algorithm, it is practicable to realize the database watermarking embedding and robust detection with the stable, high-efficiency and large-capacity database tuples clustering method, which is regarded as the basis of database watermarking algorithm in this paper. Thereinto the similarity among databases tuples is measured by Mahalanobis distance since it can effectively eliminate the influence of dimension and correlation interference. Meanwhile there are frequently database maintaining operators on tuples and attributes which would affect the robustness of database watermarking, and we use majority decision method to solve the problem when extracting watermark. Based on the above tuples clustering and majority decision strategy, we present a robust database watermarking framework shown in Figure 1(a) and (b).

The database data are mostly processed by the associated application, and for a highly available database, the original data should be restored exactly after the embedded watermark is extracted, this means that the watermark should have not only robustness but also reversibility. We already studied a reversible and blind database watermark method based on polar angle expansion before [23], which uses key as a seed to produce pseudo-random number to select the watermark embedding position, then maps these attributes to polar coordinates one by one, and embeds watermark into those points extending polar angle. It applies logistic chaotic sequence encrypt watermark to improve watermark security before watermark embedding, and adopts LSB method to extract the watermark to achieve blind detection. In view of the aforementioned database watermarking framework and our preliminary related study, the paper proposes a robust and reversible database watermark method based on clustering and polar angle expansion for numerical data.

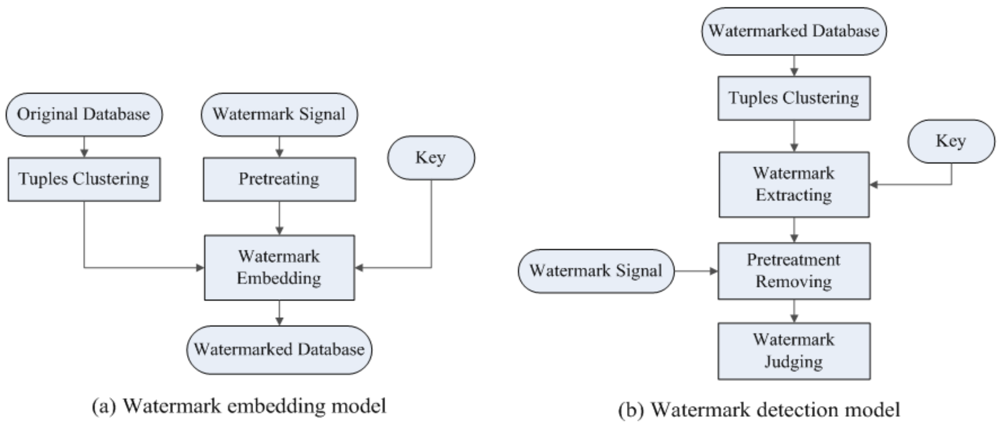

Figure 1. Robust Database Watermarking Framework based on Tuples Clustering and Majority Decision Strategy

3. Method

According to the above-mentioned framework and strategy, this section describes the reversible and robust relational database watermarking algorithm in detail, and the main idea is as follows: First we classify the tuples by the given number, and each category represents a particular meaning; Next use a key as a pseudo-random number seed to produces pseudo-random numbers to select the watermark embedding position in each category, then map these attributes to polar coordinates one by one, and embed watermark into those points extending polar angle; Finally take the LSB method to extract the watermark. Some notations used in the watermarking algorithms are given in Table 1.

Table 1. Notations used in the Watermarking Algorithms

Notation	Explanation
$\lvert W\rvert$	Binary bit length
R	Original database
R'	Watermarked database
$Y=(y_1,y_2,\cdots,y_n)$	Database tuple attribute where the watermark will be embedded
$Y'=(y_1',y_2',\cdots,y_n')$	Database tuple attribute where the watermark have been embedded
$G=(g_1,g_2,\cdots,g_k)$	Tuples clustering
$L=\{L_1,L_2,\cdots,L_k\}$	Accumulation point
$f(x)=hash(x)$	Hash function that meet $hash(Y)=hash(Y')$
$\alpha=(\alpha_1,\alpha_2,\cdots,\alpha_n)$	Polar angle corresponding to Y that meet $Y=hash(Y)\tan(\alpha)$
$\beta=(\beta_1,\beta_2,\cdots,\beta_n)$	Expending polar angle corresponding to Y' that meet $Y'=hash(Y)\tan(\beta)$
$p(D)$	Watermark detection rate
$W=w_1,w_2,\cdots,w_n$	Binary representation of the watermark
μ	The up limit of data change

3.1 Tuples Clustering

Here we apply the fast clustering method to the classification of database tuples, which begins with classifying samples roughly, then uses certain regulations to adjust the categories gradually based on the distance between samples. It is suitable for clustering analysis of large data sets. The similarity of samples is measured by distance. Due to the disunity of various attributes units in database, in order to eliminate the influence of dimension, this paper adopts Mahalanobis distance to cluster the tuples.

– **Definition 1 (Mahalanobis Distance):** $x_i=(x_{i1},x_{i2},\cdots,x_{iq})^T$ for $i=1,2,\cdots,n$ represents n samples. Mahalanobis distance is marked as $d(x_i,x_j)=\sqrt{(x_i-x_j)^T s^{-1}(x_i-x_j)}$ where s is the covariance matrix of samples.

– **Definition 2 (Clustering):** For a data set $A=(a_1,a_2,\cdots,a_n)$, clustering algorithm is to classify A into k categories marked as $G=(g_1,g_2,\cdots,g_k)$ according to the given rule. Each category has high similarity but differ greatly from other category, and it meets the condition $\bigcup_{i=1}^{k} g_i = A$ where $g_i \cap g_j = \varnothing, i \neq j$.

– **Definition 3:** q and n respectively indicate the number of attributes and tuples in database R, so n tuples can be taken as n samples in q dimensional space.

Procedure of Fast Clustering

Step1. Suppose the set $L_0=\{x_1^0,x_2^0,\cdots,x_k^0\}$ includes k initial cluster points;

Step2. Achieve initial classification according to the following rule:

$$G_i^0 = d\left(x, x_i^0\right) \le \left\{x : d\left(x, x_j^0\right), j = 1, 2, \cdots, k,\ j \ne i\right\},\ i = 1, 2, \cdots, k$$

Thus n samples are divided into k non-intersect categories $G_0 = \left\{G_1^0, G_2^0, \cdots, G_k^0\right\}$ by their respective closest initial cluster point.

Step3. Calculate new cluster points set $L_1 = \left\{x_1^1, x_2^1, \cdots, x_k^1\right\}$ based on G_0, where $x_i^1 = \frac{1}{n_i} \sum_{x_l \in G_i^0} x_l,\ i = 1, 2, \cdots, k$ is the barycenter of G_i^0 and n_i is the number of samples. Next classify samples again by L_1 to get a new classification $G_1 = \left\{G_1^1, G_2^1, \cdots, G_k^1\right\}$. Then calculate in turn as above. Assuming we get a classification $G_t = \left\{G_1^t, G_2^t, \cdots, G_k^t\right\}$ in step t, where x_i^t is the barycenter of G_{t-1} and neither sample nor the barycenter of G_{t-1}. As the increase of t, the classification tends to be stable when x_i^t approximate to the barycenter of G_t and $x_i^{t+1} \approx x_i^t, G_{t+1} \approx G_t$, and the calculation can be stopped now. Sometimes classification $G_{t+1} = \left\{G_1^{t+1}, G_2^{t+1}, \cdots, G_k^{t+1}\right\}$ and $G_t = \left\{G_1^t, G_2^t, \cdots, G_k^t\right\}$ are just the same from step t in practical calculation, and at this point the calculation can be over.

As a result, we can use the fast clustering method measured by Mahalanobis distance to classify original database tuples into desired categories. The clustering algorithm is described in detail as below:

```
Algorithm 1: G = fastClustering ($k$,$\varepsilon$, $R$)
Input: $k$ – The number of clustering
           $\varepsilon$ -- Convergence threshold
           $R$ -- Database
Output: G -- Clustering results
1.  begin:
2.      $L_0$ = randSelect($R$); // Randomly selected initial cluster point
3.      $d^0$ = minDistance($L_0$);// The minimum distance between $L_0$
4.      $G^0$ = clustering($R$,$L_0$);// Initial clustering
5.      $d^m = \infty$;
6.      while ($d^m > \varepsilon d^0$ ) then
7.          $L_i$ = clusterPoint($G_i$);// Adjust the accumulation point
8.          $G_i$ = clustering($R$, $L_i$);// clustering
9.          $d^m$= maxChangedDistance($L_i$, $L_{i-1}$);
10.     end while
11. end
```

The convergence condition set in algorithm 1 is as below: when the changed maximum distance d_m of cluster points is less than or equal to a specified value multiplied by the minimum distance d_0 of original cluster points, the algorithm will be terminated.

3.2 Database Watermarking Algorithm

Adaptive Factor. We should analyze the influence of embedded watermark to data before giving specific watermarking algorithm. Suppose the clustering result of data set A is $G=(g_1,g_2,\cdots,g_k)$ before embedding watermark and $G'=(g'_1,g'_2,\cdots,g'_k)$ after embedding watermark. Interleaved class is defined as follows.

– **Definition 4 (Interleaved Class):** For $\forall x\in A$, if x belongs to a classification before embedding watermark but not belongs to it after embedding watermark, x is called interleaved class, that is, $(x\in g_i)\&\&(x'\in g'_j)$, where $i\neq j$.

Assuming the minimum distance between any two adjacent classifications is $d_{ab}=\left\{\min\ d(x_i,x_j)\,|\,x_i\in g_a,x_j\in g_b,a\neq b\right\}$. In order to avoid arising interleaved class, the change of data should meet the following situation:

$$\begin{cases} \eta\leq\mu \\ \eta\subseteq\left\{g_a+\frac{1}{2}d_{ab}\,|\,a,b=1,2,\cdots,k,\ a\neq b\right\} \end{cases} \qquad (1)$$

The neighborhood η of x is shown in Figure 2, where x is any sample of classification g_a, x' is the sample after embedding watermark, d_{ab} is the nearest distance between g_a and g_b. Thus, the change of data just needs to meet $|x'-x|\subseteq\left\{g_a+\frac{1}{2}d_{ab}\,|\,a,b=1,2,\cdots,k,\ a\neq b\right\}$.

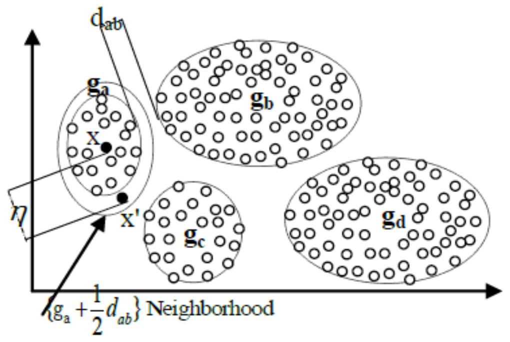

Figure 1. Situation that the Change of Data Needs to Meet

Watermarking Embedding Algorithm. The basic idea of watermarking embedding algorithm is as below: firstly generate binary watermark and use fast clustering method to classify database tuples into $|W|$ categories and list its sequence, next select the watermark embedding attribute $Y=(y_1,y_2,\cdots,y_n)$ and calculate the corresponding polar angle α based on literature [23], then get the expanding polar angle β by combining the polar angle related to each category with one watermark bit successively, finally calculate the watermarked attribute and write it back to the database. The number of embedded multiplicity is m, and the method of embedding watermark is to change the least significant bit. The watermarking embedding algorithm is described in detail as below:

```
Algorithm 2: watermarkEmbed ($key$, $R$, $w$, $m$, $\mu$)
Input: $key$, $R$ -- db, $w$ -- watermark, $m$ -- embedded repeats
            $\mu = (\mu_1, \mu_2, \dots, \mu_q)$ -- Allows you to change the scope of properties
Output: Embedded watermark $R'$
1.  begin:
2.  $W$ = binarization (Original watermark);
3.  $G$ = fastClustering (|$W$|,db);
4.  sort ($G$, $key$);
5.  $d_{ab}$=Computer ($G$); // The minimum distance between two adjacent categories
6.  for ( $i$=0; $i$<|$W$|; $i$++)
7.    for ( $j$=0; $j$<|$g_i$|; $j$++)
8       $k$=0;
      // Use hash map to select the location of embedded watermark
9.      if (hash ( $key$, $g_i[j].P$ ) && $k<m$) then
10.       $e$ = hashSelect ( key, $g_i[j].P$ );
   // Use function polarangleCal to calculate the polar angle corresponding to the selected attribute
11.       $\alpha_e$ = polarangleCal ( $g_i[j].A_e$ );
12.       if ( $\alpha_e\%2 == 0$ || $(\alpha_e\%2-1) == 0$ && (polarangleExp ( $\alpha_e, w_i$ )) meet(1) )
13.         Then
14.         $\beta_e$ = polarangleExp ( $\alpha_e, w_i$ );
15.       $g_i[j].A_e$ = attributeCal ( $\beta_e$ );// Use function attributeCal to calculate the related attribute
16.       end if
17.       $k$ = $k$+1;
18.     end if
19.   end for
20.  end for
21. end
```

The secret key and tuple primary key are taken as parameters to choose the location of embedded watermark. The function *hashSelect* is used to choose attribute subscript e of embedded watermark. The database owner holds secret key, number of embedded multiplicity and length $|W|$ of watermark.

Watermarking Detection and Data Recovery Algorithm. Watermarking detection and data recovery is the inverse of the embedding process. The main idea is that classifying test database into $|W|$ categories, next using secret key to find the position of embedded watermark and calculating the corresponding polar angle β, then extracting watermark from β by the means of LSB and majority decision method and getting polar angle α, finally restoring the original attribute and writing it back to the database. The detailed process is described as below:

```
Algorithm 3: watermarkDet ($key$, $R'$ , $length$, $m$, $\mu$)
Input: key; $R'$ -- Candidate database;
       $length$ -- Binary watermark bit length;
          $m$ -- Embedded repeats;
          $\mu = (\mu_1, \mu_2, \dots, \mu_q)$ -- Allows you to change the scope of properties
Output: $W$ -- watermark
1.  begin:
2.  G = fastClustering (length, wdb);
3.  sort (G, key);
4.  for ( $i$=0; $i$< length; $i$++)
5.    for( $j$=0; $j<|g_i|$; $j$++)
6.      $k$=0;
7.      if ( hash(key,$g_i[j].P$ ) && $k<m$)   then
8.        e =hashSelect ( key, $g_i[j].P$ );
9.       $\beta_e$ = polarangleCal ( $g_i[j].A_e$ );
10.       if ( ( $\beta_e$ - lsb ( $\beta_e$ ) )meet(1) )
11.         then  $w_i[k'] = lsb(\beta_e)$;
   // Use function polarangleRes to restore the original polar angle
12.              $\alpha_e$ = polarangleRes ( $\beta_e, w_i[k']$);
13.              $g_i[j].A_e$ = attributeCal ( $\alpha_e$ );
14.       end if
15.     end if
16.     $k = k+1$;
17.   end for
18.  end for
19.  for( $i$=0; $i$< length; $i$++) // Majority decision
20.     if( zeroCount( $w_i[k']$ ) > ($m$/2) )
          then $w_i'$=0;
21.       else $w_i'$=1;
22.     end if
23.   end for
24.  end
```

Function *sort* is used to achieve synchronization of detecting watermarking, which takes secret key as parameters and lists its sequence based on tuple primary key. Due to repeated embedding, the watermark bit is decided by majority decision method.

4. Simulation Experiment and Analysis

We use open-source database *MySQL* to make research and simulation of database watermark and take visual studio as fore-end. There is 100000 tuples, each of which has 21 attributes (attributes value is generated randomly by computer). Selecting 10 numeric data as candidate attributes to embed watermark, and inserting "HNU" into database for 100 times. Moreover experimental result is compared with literature [10] algorithm under the same data set.

4.1 Invisibility

The holistic influence of embedded watermark to each attribute column of data in database (Rounded to 3 decimal places) is shown in Table 2. It can be seen that the error caused by embedding watermark is very small and not far removed from the result of literature [10] algorithm.

Table 1. Invisibility after Embedding Watermark

Attributes	The changed ratio of Mean (%)		The changed ratio of Variance (%)	
	Our algorithm	Literature [10] algorithm	Our algorithm	Literature [10] algorithm
a1	0	0	0.002	0.001
a2	0	0	0.006	0.000
a3	0	0	0.002	0.004
a4	0	0.013	0.001	0.000
a5	0	0	0.004	0.024
a6	0	0	0.005	0.009
a7	0	0	0.004	0.003
a8	0	0	0.004	0.006
a9	0	0	0.001	0.000
a10	0.043	0.052	0.014	0.011

4.2 Test of Database Reversibility

Due to space limitations, here we only talk about a group of 24 watermarked attributes, as shown in Table 3. We can find that the restoration is satisfactory.

Table 2. Situation Before and After Data Restoration

Before restoration	After restoration	Before restoration	After restoration
132.00	132.34	90.00	89.99
128.25	128.05	184.5	184.50
504.00	504.20	270.00	270.30
135.00	135.00	180.00	180.10
210.00	210.25	210.00	210.25
72.00	73.00	391.50	392.00
420.00	422.10	326.25	326.25
56.25	56.20	114.00	114.00
356.25	356.25	40.50	41.00
54.00	54.20	477.75	478.00
20.25	20.10	67.50	67.90
322.50	322.47	276.00	276.00

4.3 Test of Watermarking Robustness

The simulated attacks include subset selection, subset addition and subset modification. These attack tests take the current system time as random seed and select tuples and attributes randomly (taking the average of 20 tests). The result of simulation experiment is shown in Figure 3, 4 and 5.

Figure 3 shows that the detection effect on subset selection attack is better than the algorithm from literature [10] and increased by nearly 5%. Figure 4 shows that the robustness on subset addition attack is preferably and relatively stable. Figure 5 shows that the robustness on subset modification attack is the same as literature [10] algorithm on the whole.

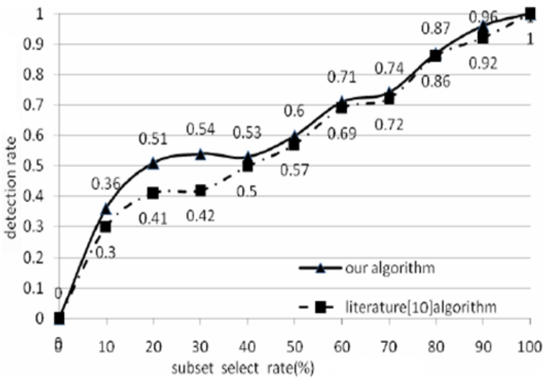

Figure 2. Detection Situation of Subset Selection Attack

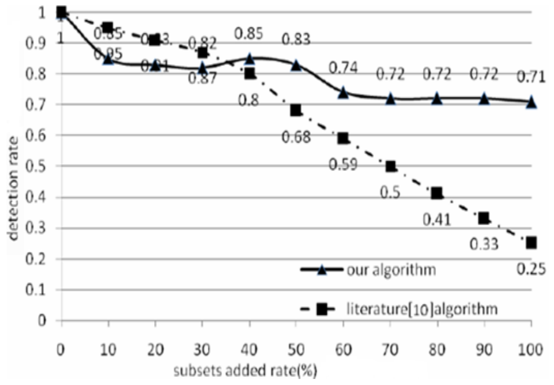

Figure 3. Detection Situation of Subset Addition Attack

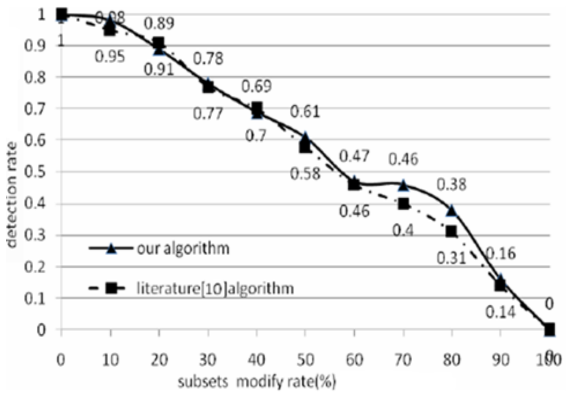

Figure 4. Detection Situation of Subset Modification Attack

4.4 Analysis of Algorithm Time Complexity

Time complexity of Algorithm 1. The original operations of fast clustering which classify n samples into k categories include calculating the distance between two samples, comparing the size, and calculating cluster points. Suppose $f(n)=O(n+k)+O(nq(k-1))$ represents the time complexity at the iterations, where the first item is the time complexity of computing cluster points and the second item is the asymptotic time complexity at one clustering. The algorithm will be stopped after n iterations, so the whole asymptotic time complexity is:

$$T_1(n)=O(tkqn) \tag{2}$$

Where t is the number of iterations, k is the class number of clustering, q is the number of attributes (dimensionality) and n is the samples number.

Time complexity of Algorithm 2. Watermarking embedding includes binarizing, clustering, sorting and embedding, thus the asymptotic time complexity of Algorithm 2 is $T_2(n)=O(k)+O(tkqn)+O(n\log^n)+O(k\,|\,g_i\,|)$. Where $|\,g_i\,|$ is the samples of category i and whose extremum is $|\,g_i\,|=n$. Since k and q are far less than n, $T_2(n)=O(tkqn)+O(n\log^n)$. Seriously, due to the local convergence of fast clustering [24], the number of iteration t is uncertain. Convergence criterion defined in Algorithm 1 indicates that t is less than n, therefore the time complexity of the algorithm in the worst case is:

$$T_2(n)=O(n^2) \tag{3}$$

Time complexity of algorithm 3. Similarly, the time complexity of Algorithm 3 in the worst case is $T_2(n)=O(n^2)$.

4.5 Capacity

The length of watermarking sequences is $|W|$, tuples number is n and the number of embedded multiplicity is m, thus the capacity $c=n/(|W|\times m)$. It can be seen that once the tuples number n in database and watermark are made, only can we adjust the embedded multiplicity m to reduce capacity so as to make small data modifications. Owing to the higher watermarking robustness requirement of copyright protection is allowed.

4.6 Robustness Analysis

Suppose the attackers select each tuple with an equal probability $p(t)=1/n$ and choose each attribute of tuple with an equal probability $p(A)=1/q$. At the following, we will analyze the watermarking detection rate under attacks such as random bit flipping, subset selection, sorting, subset substitution and subset addition.

Random Bit Flipping. We assume that the attackers know the number of database classification, namely: the length of watermarking sequences is $|W|$. The most extreme case of destroying watermarking detection is to make random bit flipping on the category with least data records. Suppose the category tuple records is ν, the attackers randomly choose ξ tuples and flip the LSB bits of all attributes without impacting data. Thus the watermark can be detected by probability:

$$p(D) = 1 - \frac{\binom{\nu - m}{\xi - m}}{\binom{\nu}{\xi}} \tag{4}$$

where $\xi \geq m$.

Subset Selection. Similarly, suppose the attackers know the number of categories. If they can pitch on the tuples without watermark on the category with least data records, they are able to destroy watermark detection successfully as random bit flipping. The probability of detecting watermark successfully is:

$$p(D) = \sum_{i=1}^{m} \frac{\binom{m}{i}\binom{\nu - m}{\xi - i}}{\binom{\nu}{\xi}} \tag{5}$$

Sorting. It makes no difference to watermark detection if the attackers randomly resort the database tuples. We just need to make database fast clustering and recover the original order by secret key rearrangement of each category, then extract watermark.

Subset Substitution. Subset substitution is similar to subset selection.

Subset Addition. Subset addition will only increase the tuple records of each category. Since the embedded location is determined by the secret key and hash mapping of tuple primary key in the process of watermark embedding and detecting, subset addition will not produce huge impact.

Secondary Watermark Addition. Suppose A inserts watermark w_a into R to get R_a, while B pirates the database of A and makes some operations as above to obtain attacked database R'_a, then adds its own watermark w_b to get database R''. So A can detect watermark w_a from R'' in probability $p(D)$ and B only can detect watermark w_b from R_a, in probability $\rho \approx 0$. As a result, it is effective to resist secondary watermark addition attack with the probability of $p(D)$.

5. Conclusion

This paper provides a novel adaptive watermarking scheme based on clustering and polar angle expansion for relational database, which first takes advantage of the disorder character among database tuples to cluster them by Mahalanobis distance, and then combines with the polar angle expansion strategy to embed and extract watermark. The scheme shows a high robustness under blind detection for subset selection, addition and modification attack, and also can restore the original data more truly. Due to the local convergence of fast clustering and the error of restoration data, it can not satisfy the application requirement of high-accuracy data. The next step is to adopt new update strategy to speed up convergence rate and global convergence, then design a completely reversible database watermarking algorithm and prove it in theory.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61173107), Natural Science Foundation of Hunan Province, China (10JJ5068) and the Science and Technology Plan of Changsha, Hunan Province, China (K1109099-11).

References

[1] R. G. van Schyndel, A. Z. Tirkel and C. F. Osborne, "A Digital Watermark", Proc. ICIP'94, vol. 2, **(1994)**, pp. 86-90.
[2] I. Cox, M. Miller, J. Bloom and C. Honsinger, "Digital Watermarking", Academic Press, USA **(2002)**.
[3] R. Sion, M. Atallah and S.Prabhakar, "Rights Protection for Relational Data", IEEE Transactions on Knowledge and Data Engineering, vol. 16, no.12, **(2004)**, pp. 1509-1525.
[4] R. Agrawal and J. Kiernan, "Watermarking Relational Databases", Proc. VLDB'02, **(2002)**, pp. 155-166.
[5] R. Sion, M. Atallah and S. Prabhakar, "On Watermarking Numeric Sets", Proc. IWDW, **(2002)**, pp. 12-15.
[6] X. Niu, *et al.*, "Watermarking Relational Databases for Ownership Protection", Chinese Journal of Electronics (in Chinese), vol. 31, no. 12A, **(2003)**, pp. 2050-2053.
[7] Y. J. Li, V. Swarup and S. Jajodia, "Fingerprinting Relational Databases: Schemes and Specialties", IEEE Transactions on Dependable Secure Computing, vol. 2, no. 1, **(2005)**, pp. 34-45.
[8] Y. Zhang, X. M. Niu and D. N. Zhao, "A Method of Protecting Relational Databases Copyright with Cloud Watermark", Proc. World Academy of Science, Engineering and Technology, vol. 3, **(2005)**, pp. 68-72.
[9] Y. Zhang, B. Yang and X. M. Niu, "Reversible Watermarking for Relational Database Authentication", Journal of Computers, vol. 17, no. 2, **(2006)**, pp. 59-65.
[10] G. Gupta and J. Pieprzyk, "Reversible and Blind Database Watermarking Using Difference Expansion", International Journal of Digital Crime and Forensics, vol. 1, no. 2, **(2009)**, pp. 42-54.
[11] Z. H. Zhang, *et al.*, "Watermarking Relational Database Using Image", Proc. ICMLC, **(2004)**, pp. 1739-1744.
[12] X. M. Jin, *et al.*, "Watermarking spatial trajectory database", Proc. DASFAA, **(2005)**, pp. 56-67.
[13] X. C. Cui, *et al.*, "A Robust Algorithm for Watermark Numeric Relational Databases", Intelligent Control and Automation, vol. 344, **(2006)**, pp. 810-815.
[14] H. P. Guo, *et al.*, "A Fragile Watermarking Scheme for Detecting Malicious Modifications of Database Relations", Information Sciences, vol. 176, no. 10, **(2006)**, pp. 1350-1378.
[15] M. L. Meng, X. C. Cui and H. Cui, "The Approach for Optimization in Watermark Signal of Relational Databases by using Genetic Algorithms", Proc. ICCSIT, **(2008)**, pp. 448-452.
[16] M. Shehab, E. Bertino and A. Ghafoor, "Watermarking Relational Databases Using Optimization-based Techniques", IEEE Transactions on Knowledge and Data Engineering, vol. 20, no. 1, **(2008)**, pp. 116-129.
[17] G. Gupta and J. Pieprzyk, "Database Relation Watermarking Resilient against Secondary Watermarking Attacks", Proc. 5th International Conference on Information Systems Security, **(2009)**, pp. 222-236.
[18] S. Bhattacharya and A. Cortesi, "A Generic Distortion Free Watermarking Technique for Relational Databases", Proc. 5th International Conference on Information Systems Security, **(2009)**, pp. 252-264.
[19] I. Kamel, "A Schema for Protecting the Integrity of Databases", Computers & Security, vol. 28, no. 7, **(2009)**, pp. 698-709.
[20] A. H. Ali, *et al.*, "Copyright Protection of Relational Database Systems", Proc. 2nd International Conference on Networked Digital Technologies, vol. 87, **(2010)**, pp. 143-150.
[21] G. A. David, "Query-Preserving Watermarking of Relational Databases and XML Documents", ACM Transactions on Database System, vol. 36, no. 1, **(2011)**, pp. 301-324.
[22] M. E. Farfour, *et al.*, "A blind reversible method for watermarking relational databases based on a time-stamping protocol", vol. 39, no. 3, **(2012)**, pp. 3185-3196.
[23] W. C. Tao, Z. Y. Li and H. F. Li, "Reversible and Blind Database Watermark Algorithm Based on Polar Angle Expansion", Computer Engineering (in Chinese), vol. 36, no. 22, **(2010)**, pp. 155-157.
[24] Z. Q. Wen and Z. X. Cai, "Convergence Analysis of Mean Shift Algorithm", Journal of Software (in Chinese), vol. 18, no. 2, **(2007)**, pp. 205-212.