DWT Based Blind Watermark in Relational Database

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

Juan Wang
Department of Computer Science
China West Normal University
NanChong, China
email:wjuan0712@126.com

***Abstract*—Aiming to balance the robustness and imperceptibility of database watermark, propose a wavelet transform (DWT) based blind watermarking algorithm. The algorithm screens candidate attributes that can be embedded watermark and conducts subset segmentation and rearrangement, and then performs DWT transformation to the data subsets and the scrambled watermark image respectively. Embed the compressed low-frequency part of the watermark into the High-frequency part of the data set to achieve data fusion. Theoretical analysis and experiments show that the algorithm enjoys strong robustness and good invisibility.**

Keywords- digital watermark, relation database, wavelet transformation, condensation, robustness

I. INTRODUCTION

Database watermark is currently facing two major challenges, one is the small redundant space and the limited watermark, the other is the high database watermark robustness requirement, which will result in watermark lost in normal update to the database otherwise. But these two are contradictory, low redundant is bound to bring about small watermark and poor robustness, a large redundancy is required to enables strong robustness. On the premise of guarantying the invisibility and integrity, there are a large number of studies have been done by scholars home and aboard on how to improve the watermark robustness in limited redundant space. By modifying some of the least significant bit(LSB) value for watermark embedding in literature [1] and it was easy to conduct but with poor robustness that would lead to abnormal state. R.sion et al [2-3] implemented the watermark embedding by changing the distribution of data in a continuous sequence and the robustness was good but only applicable for part data items which limited the capacity of watermark embedding. Francesc Sebe et al [4-5] spreaded watermark spectrum with a pseudo-random sequence, by adjusting the parameters to maintain the data average. It was robust to noise adding like attacks but was difficult to meet the requirements of database dynamical update with a limited watermark channel. Niu [6] extended the LSB method, which only embed the LSB of properties that meet the constraint and achieved a multiple meaningful bits embedding with a probable abnormal result. Zhang et al [7-8] studied the primary key and characteristics of content based database watermark but was still hard to embed significant information and the information was limited.

Propose the DWT-based robust watermarking blind algorithm aim to solving these issues using image watermark skills. The algorithm select those properties that with a more important degree and a high redundancy as candidate by using screening algorithm. Perform wavelet transform on the low-frequency part of copyright image to construct a watermark and compress it. And plunge the compressed watermark into the high-frequency part of the raw image in wavelet domain and achieve the watermark and data integration. Experimental results show that the algorithm greatly improve the robustness and invisibility.

II. ALGORITHM PRINCIPLE

The basic idea of wavelet transform is the detailed frequency and decomposition to signal, that is, multi-resolution decomposition. A such two-dimensional signal as an image by a wavelet transform, is decomposed into a four one level subgraph through one level decomposition, namely, one low-frequency approximate subgraph of *LL1* (approximation of the original image) and three high-frequency subgraphs of *HL1* (horizontal details), *LH1* (vertical details) and *HH1* (diagonal details). If performs wavelet decomposition again to one level low-frequency approximate subgraph *LL1*, four tow level subgraphs with the more lower resolution appear. The approximate subgraph *LL2*, the horizontal details subgraph *HL2*, the vertical details subgraph *LH2*, the diagonal details subgraph *HH2*and if this process is repeated the multi-layer wavelet decomposition subgraphs are available. After wavelet decomposition, the lower the frequency is, the greater the coefficient of sub-band is, the more information and energy the subgraphs contained. The low-frequency part of the image concentrates most of the image energy and depicts the main features of the image, which is the best approximation of the original image and the coefficients distribution and the statistical characteristics are similar to that of the original image. If embed watermark in the low-frequency coefficients, the watermark enjoys good stability and strong anti-attack ability, but the change to coefficients is likely to affects the host value. The high-frequency part of the image is the details in different scales and resolution and is less important relatively, if embed watermark in this part, the fusion effect is transparent and imperceptible to human eyes, but some

Published by Atlantis Press, Paris, France.

information is probable lost after encountered some image processing. Therefore, on the premise of do not influence the use-value of data, modifying the high-frequency coefficients to achieve information hiding and balance the robustness and imperceptibility.

It is possible to handle the numerical data in database as pixels in image, map it to two-dimensional space signal by way of line scanning and conduct DWT transformation to the signal to achieve watermark embedding by changing the high-frequency coefficients in DWT domain. In fact, embedding watermark in this part may decrease the robustness, but it could balance the modification to the original image, that is, reduce the visibility of the watermark. On the other hand, embedding watermark in this region enable choosing a higher watermark intensity to compensate the reduction of robustness. Then the data processing is transferred from space to frequency domain, and construct watermark utilizing the low-frequency information that contains main features of the copyright image and then embed the watermark. This approach not only makes the uniform distribution of watermark energy, but also reduces the amount of watermark to achieve the purpose of repeating embedding, and totally will greatly enhance the watermark robustness and invisibility.

III. THE DWT BASED BLIND ALGORITHM

In order to improve the invisibility and randomness of the watermark, the algorithm scrambles the watermark image at first with Arnold approach, and then screens those attributes that can be used to embed watermark and executes subset segmentation and rearrangement respectively. And performs DWT transformation to subset data and scrambled watermark image and finally compress the low-frequency part that contains most of the watermark energy and embed it to the high-frequency coefficients of the subset, which is essential to realize the fusion of watermark and data. Take the combination of Hamming code and majority selection method as the extraction approach to improve the watermark detection rate.

A. *Watermark Embedding Algorithm*

1) *Copyright Image Processing*

The watermark in this algorithm is a binary image, in order to enhance the security of the algorithm and eliminate the correlation between pixels in watermark image, we employ Arnold scrambling to watermark image at first. The scrambled digital watermark is very large, if embed watermark into database directly would require a large redundant space of the database, otherwise it can not be embedded. Even if they are embedded, as a result, this will cause a lot of modification among the database and greatly impact its use-value. So, for the sake of not losing watermark information, we reduce the quantity of watermark that to be embed by compressing the low-frequency part of copyright image after wavelet transformation.

2) *Data Screening Algorithm*

Definition 1: Given that the relationship of database is $R(P,A_1,A_2,\dots,A_v)$, where P is primary key and A_j are numerical attributes of $R(0\leqq j\leqq M)$. $r_i(1\leq i\leq n)$ are tuples of R and each tuple has a primary key $r_i{\cdot}p$ and v numerical attributes $r_i{\cdot}A_1$, $r_i{\cdot}A_2,\dots$, $r_i{\cdot}A_v$..

Definition 2: The conditions of numerical attribute $r_i{\cdot}A_j$, $(1\leq i\leq n,1\leqq j\leq v)$ that can be embedded watermark under the constraint of $b\%$ is $Floor(lb(r_i{\cdot}A_j*b\%))>0$ and call those attributes that meet the clause as candidate attributes, where *Floor()* is the rounding down function

Definition 3: Suppose $I_j(1\leqq j\leq v)$ is the weight of candidate attribute $A_j(1\leqq j\leq v)$, the value of I_j is determined by the significance and redundancy of attributes.

Definition 4: If and only if the weight of candidate attribute $A_j(1\leqq j\leq v)I_j$ is greater than I_0, we select A_j as the object to embed watermark and I_0 is determined by users in the light of the embedding intensity of watermark and preserved as the key.

3) *Data Identification Algorithm*

The arranging order of tuples and columns are changeable in the database and the data is changing frequently in the course of database manipulation. How to ensure the tuple is exactly the one that was embedded watermark when conduct watermark extraction, the key step is identify the tuple, that is, assign an identity card like tag *ID* number for each tuple. An effective marking algorithm must be able to resists various attacks and enjoys strong robustness. The unidirectional Hash function is usually used to mark the candidate attribute that can be embedded watermark. For a unidirectional Hash function H, enter information *M* in a certain length and will always output a value in solid length. And the forward calculation is easy and the reverse calculation is extremely difficult, it can also resists the birthday attack and it is too hard to find *M* and *M'* to make *H(M)=H(M')*. Work out the value of *ID* according to the attribute name *A*, the primary tuple key *P* and the user key *K*, namely, *ID = hash (P, A, K).*

4) *Watermark Embedding*

The watermarking embedding are mainly includes three aspects: one is to determine the position in which the watermark is embedded, that is, find a possible place to embed the watermark in original data; specify watermark embedding density, that is, increase the embedding intensity to improve watermark robustness and ensure the invisibility simultaneously on the circumstance of not affecting the database usage; select the appropriate watermark embedding method, the embedding model, to embed the watermark. In order to balance the invisibility and robustness, we embed low-frequency wavelet coefficients of the watermark image to the wavelet transformed high-frequency part of data. The specific embedding process is shown as follows:

a) Performs *K* times of Arnold scrambling to a $M\times N$ binary image *W* and *W* becomes *W'*. $W' = \{W'(i, j)|0\leq i\leq M, 0\leqq j\leq N\}$, 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 low-frequency subblock LL_3. Calculate the mean of the coefficient matrix and label it *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 LL_3'.

b) Utilizing the algorithm mentioned in section 3.1.2 screen the candidate attributes that can be embedded watermark and using label algorithm mark the A_j, $ID_{ri.Aj} = hash(Key, P, A_j)$.

c) Grouping the data into λ packets according to the values that come from the labeled *ID* mod λ(*ID* % λ). $Group\ (k) =ID_{ri.Aj}\ Mod\ \lambda\{0\leqslant k\leqslant\lambda\text{-}1\}$, 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.

d) Sort the data in each packet according to the *ID* value, there are total $M \times N$ bits in each packet (the watermark length) and fill them with 0 if void bits appeared.

e) Perform three level wavelet transformation to each *Group(k)(0* ≤ *k* ≤ *λ-1)* respectively and get the third level high-frequency subblock $HH_3^k(0 \le k \le \lambda - 1)$.

f) By way of adding embed the compressed watermark low-frequency coefficients LL_3' into the high-frequency subblocks $HH_3'(1 \le k \le \lambda-1)$ of the host data. The specific embedding mode as follows: $HH_3^k = HH_3^k(1+\alpha LL_3')$, $(0 \le k \le \lambda-1)$ where α represents the intensity factor of high-frequency subband watermark embedding.

g) Conduct the inverse transformation to the watermarked coefficients and get the watermark contained data.

B. *Watermark Detection 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. The threshold T can be set based on the actual situation and considering the normal database update and the malicious attacks, that is, balance the robustness and accuracy of the algorithm.

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

Specific watermark detection process is shown in Figure 1.

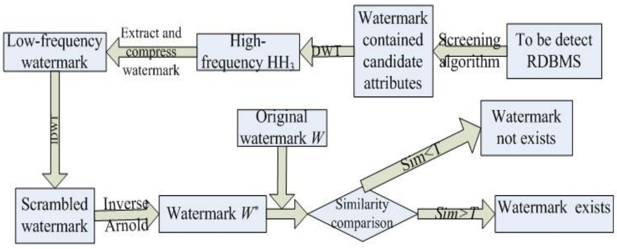

Figure 1. Watermark Detection Process

IV. ALGORITHM ANALYSIS

From the point of robustness: 1. Scramble watermark image with Arnold, eliminate the correlation of each pixel and make the distribution of pixels disorderly and unsystematic, which not only improve the robustness, but also enhance anti-interference ability. Even if the database is damaged in the course of regular use, the extracted damaged watermark bits are distributed the whole image after performing the inverse Arnold transformation, which is not obvious to human visual system. At the same time, the extracted watermark is an scrambled image, attackers do not know how to recover the original image at all. 2. The database owner define the scope of the data that can be modified and limit the weights of the attribute according to the actual requirement. And only embed watermark in attributes that is essential and with large redundant space, which protect the usefulness of the data in a maximum extent. Hackers are generally not modify or delete the important data on a large scale, otherwise the database will lost the due value to them, which improve the resistance to a attacks of subset modification and the subset deletion effectively. 3. What is embedded is the low-frequency part of the watermark, which contains most of the image energy and can still contain recognizable features when damaged or interfered. 4. Compress the low-frequency part of the watermark to construct the embedded watermark, which get the watermark capacity reduced greatly, decrease the changes to the original database. Combine Hamming code and majority election to detect watermark and conduct the watermark embedding repetitive, which improves the robustness of the watermark further. 5. The watermark embedding and detection are carried out under the control of the key, whether the watermark is embedded or not depends on the constraints and weights of the attribute, the embedded position and the value are related to primary key. Attackers must be firmly believed that a tuple is embedded watermark and which attributes are embedded and how embedded watermark if they want to erase the watermark successfully. So, the probability to erase the watermark is extremely low. 6. Identify each tuple with Hash function and stabilize the structure between tuples relatively, which solving the reordering problem. 7. Conduct watermark algorithm to those only tuples that meet the rules when doing database update to achieve the dynamic watermark embedding for adapting to the frequent updates of the database.

From the point of invisibility: 1. Modify the data in frequency domain, embed watermarks in high-frequency part of wavelet transformed data, which make the distribution of the watermark is more uniform, dispersed and the embedded watermark does not significantly response to the data modification. And the modified high-frequency (detailed part) change data little after its inverse transformation, which will not change the overall distribution of the data and enjoys good concealment. 2. Embed different watermarks into the same attribute column and the same watermark into the varied columns, then generally balance the modification, greatly reduce the impact to database and enhance its transparency.

V. SIMULATED EXPERIMENT

To verify the effectiveness of the algorithm, we carry out relevant tests, the adapted experimental data is a table in a student relation database and the table has a total of 200000 records. Choose four of numeric attributes and a 32×16 watermark image to embed watermark. The experiment is developed with Matlab7.0.1 and VC6.0 through connecting to SQL2005 database by ODBC. We let b = 0.1, I0 = 0.6, λ = 10, α =0.0004, there are 58740 measured attributes can be embedded watermark. Then embed the equivalent watermark into relationship of the database with this watermark algorithm and conduct a comparative analysis to document [9].

1. Watermark embedding error. Figure 2 shows the variation of mean and variance caused by the watermark embedding. As we can see from the figure, the error is very small. The algorithm has better error control ability and invisibility compare to the experimental [9].

2. Anti-interference. Figure 3 shows the subset selection, subset adding and subset modification attacks compare to experiment [9].

The experimental results show that the algorithm improves high in anti subset modification and adding compare to document [9] due to only select attributes that is important and with large redundancy to embed watermark by using screening method. The illegal users are intend to profit from the database, he must make a trade-off between destroy the digital watermark and the availability of the database, so it is generally not possible to modify these important data. In addition we embed the compressed watermark into the wavelet domain, the watermark distribution is not only uniform but also reduce the amount of the watermark, which achieves embedding repetition.

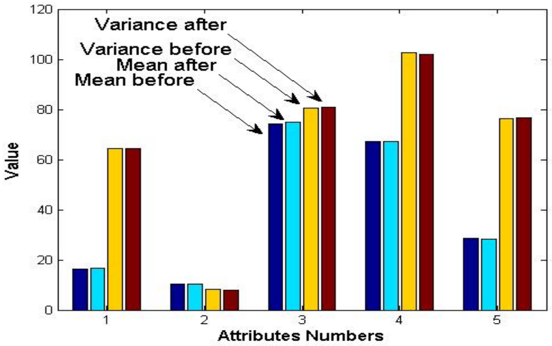

Figure 2. The Comparison of Mean and Variance before and after Watermark Embedding

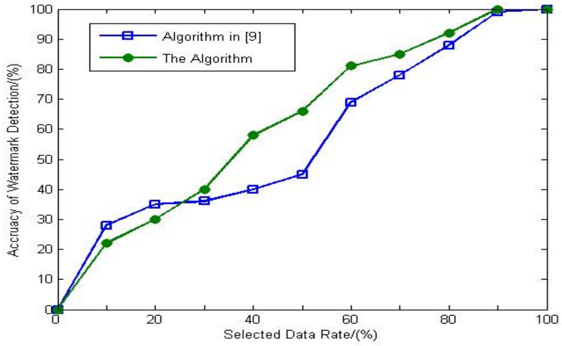

Figure 3(a). Comparison of Data Selection

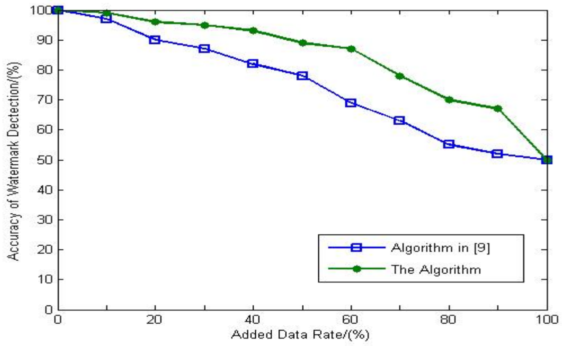

Figure 3(b). Comparison of Data Addition

Even if the watermark is destroyed, the modified watermark distribute to the whole image after the inverse Arnold transformation, so the damage is not obvious. Therefore, this algorithm enjoys strong anti-attack capability.

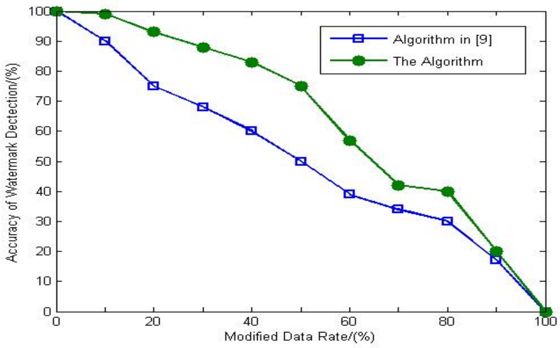

Figure 3(c). Comparison of Data Modification

VI. CONCLUSION

Propose a DWT-based watermarking algorithm in database field. 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, especially enjoys sturdy immunity for subset modification and subset deletion.

REFERENCES

[1] Agrawal R, Kiernan J. Watermark relational databases. The 28th VLDB Conference, Hong Kong, China, 2002:155-156.

[2] Sion R, Atallah M, Prabhakar S. Watermarking relational databases. Indiana: the Center for Education and Research in Information Assurance and Security of Purdue University, 2002:36-45.

[3] Sion R, Atallah M, Prabhakar S. Ownership proofs for categorical data. Proc of the IEEE International Conference on Data Engineering, Boston, 2004: 584-596.

[4] Sebe F, Domingo Ferrer J, Solanas A. Noise-robust watermarking for numerical datasets. LNAI 3558: Proceeding of the MDAI, 2005: 134-143.

[5] Sebe F, Domingo Ferrer J, Castella Roca J. Watermarking numerical data in the presence of noise. International Journal of Uncertainty, Fuzziness and Knowledge Based Systems, 2006, 14(8): 495-508.

[6] X.M. Niu, L. Zhao, W.J. Huang et al. Watermark Relational Databases for Ownership Protection. Acta Electronica Sinica, 2003, 1(31): 2050-2053.

[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.

[9] R. Wu, J.H. Cao, M. Huang et al. A New Digital Watermark Technology of Relational Database. Wuhan University Journal (Natural Science Edition), 2005 51(5): 590-593.

Published by Atlantis Press, Paris, France.