International Journal of Pure and Applied Mathematics
Volume 117 No. 20 2017, 881-886
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue

ijpam.eu

A Robust Database Watermarking Algorithm using Bezier Cubic Curves

Manoj Kumar
Delhi Technological University, Delhi, 110042, INDIA
mkumarg@dce.ac.in

Om Prakash Verma
Delhi Technological University, Delhi, 110042, INDIA
opverma@dce.ac.in

Abstract—The piracy of digital assets including databases is usually based on insertion of digital watermarks. In this paper we present a new robust database watermarking scheme for copyright protection of relational databases. A cubic Bezier curve is used as watermark A secret key based watermark embedding scheme is used to embed the watermark in relational database. Cubic curve parameters along with many other secret parameters and secret key makes this scheme very secure in comparison to other existing schemes. This scheme is resilient against wide variety of attacks such as database modifications, insertion of new records, deletion of existing records because these attacks do not destroy the watermark completely. Though extracted watermark sometimes contains too much noise but still cubic curve is visible and identifiable. Proposed scheme can use single or multiple attributes for embedding watermark.

Index Terms—Bezier cubic curves, curve control points, message digest, Message Authentication Code (MAC)

I. Introduction

Digital watermarking technology has been widely used as an effective method to achieve copyright protection, for the image, video, audio and other areas. The digital watermarking technology on relational databases has gradually increased, several relational databases watermarking algorithms have appeared, which provide necessary copyright protection for relational databases. Nowadays, sharing information online is an important activity for business and research. It also involves buying/selling of databases. Sharing of data related to sales, surveys, weather, stock markets, power consumption, consumer behavior, medical tests, scientific experiments, etc. is frequently required. Consequently ,there is a great need for providing security of databases to discourage illegal copying and distribution in today's internet-based application environment [1]. In this context, proof of ownership and tamper-proof-transportation of the databases are the most challenging issues these days[2]. However, the existing works do not fully consider how to utilize the human vision and other physiological characteristics to enhance the robustness of relational data watermarks. In this paper, the watermark used is a curve, an identifiable shape, comparing logos or small images which are usually not identifiable if distorted badly.

The remainder of this paper is organized as follows: section II describes some related work in database watermarking. Section III elaborated proposed work describing embedding and extraction process. Section IV analyzes experimental results. and finally, we conclude with a summary and discussions in section V.

II. Related Work

The groundbreaking study in this area was conducted by R. Agrawal and R. Sion in 2002 [3,4]. In 2003, X.M. Niu proposed that a meaningful string could be inserted into the relational database as the watermark [5]. Y. J. Li [6] raised a method of inserting watermark by changing the order of relational data index and keeping the physical location or data values unchanged to impair its use. While the index is additional information outside of relational data content, watermark information could be completely lost if the 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 [7]. When being extracted, the cloud droplet should be compared with original copyright image. G. Gupta utilized difference expansion and Lowest-Effective-Bit on integers to achieve embedding and blind detection of the watermark. however, this method is restrictively used for integer data [8]. Many other watermark workers also make a lot of efforts to promote the development of database watermarking, [9,10,11,12] yet there are still many shortcomings in the said study. A novel approach for tamper detection, analysis and recovering original data back by adding a set of bits to the original data that acts as a watermark was proposed in [13]. The proposed system provides a strong validation and recovery scheme to maintain data security and integrity. Another scheme for embedding watermark information Reversible Watermarking uses the most irrelevant feature [14]. There is scope to apply this scheme to preserve data ownership of dataset which are used for supervised learning. Features importance depends on the relevance between the feature and the class feature in such datasets. Various attacks like insertion, deletion and modification are then performed to check the robustness of the watermark. The watermark robustness is too weak to resist various conventional database operations and

illegal watermark attacks, such as selection, addition, modification and so on. As a result, improving the robustness of database watermarking is a challenging and yet significant work.

II. Proposed Work

The proposed system makes use of a Cubic Bezier Curve according to the parameters provided and then hides the cubic curve in the database by using an embedding algorithm. Only floating point attributes are used to hide watermark information. Watermark information is hidden in the fractional part of the floating point attribute. Here we assume that the attribute used for hiding the watermark are such that small changes in some of their values are acceptable [3]. All floating point attributes are not used for watermarking. Proposed algorithm can be implemented using one, two or three attributes. Embedding algorithm utilizes a parameter γ and a secret key K which provides control for the hiding and recovery processes restricting extraction by those who do not possess the key or do not have access to it.γcontrols the embedding density by choosing one tuple out of α tuples in the database for hiding one point of the cubic curve. Bezier cubic curve is defined using four points P_0, P_1, P_2, and P_3. P_0 is the starting point for curve, and P_3 is the ending point of the curve. Line joining P_0 to P_1 defined the tangent on curve at point P_0, whereas line joining P_2 to P_3 defines tangent on curve at point P_3. Thus four points control the location and shape of the curve. Each point P_i is defined as (P_{ix}, P_{iy}). The algorithm makes use of security function Message Authentication Code (MAC). This is key base hashing algorithm H=MAC(P,K), which generates a MAC value for some input P based on a secret key *K*.

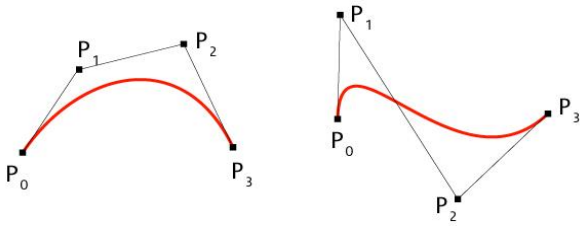


Fig. 1. Bezier Cubic Curve with four control points controlling shape of curve.

A. Embedding Process

Suppose that we are watermarking a database relation R whose scheme is R(P,A_0, . . . , A_{v-1}), where P is the primary key attribute.

Following are some parameters used in proposed algorithm:

- Secret key *K*, parameters γ, *i*, *j*, *k*, points P_0, P_1, P_2, and P_3 are known only to the owner of the database.
- All points P_0, P_1, P_2, and *P3* are normalized to have values between *0* and *1*.
- $0 \leq t \leq 1$and all computed *x(t)* and *y(t)* have values between *0* and *1*.

Table.1List of Parameters and symbols

η	Number of tuples in relation
ν	Number of attributes in relation
γ	Density control parameter
i, j	Attribute A_i for embedding *x(t)* value and A_j for embedding *y(t)* value.
K	A secret key
ω	Actual number of tuples marked
P_0, P_1, P_2, P_3	Four points controlling Cubic curve shape.
k	k-th byte of hash is used to derive parameter value t for the curve

```
1)  ω = 0;
2) for each tuple r ∈ R do
3)     compute hash H = MAC( r.P || K )
4)     if (H mod γ ) = 0) then // mark this tuple
5)          derive  t  from k-th byte of  H
6)          compute x(t) and y(t) from curve equation
7)          replace fractional part of r.Ai with x(t)
8)          replace fractional part of r.Aj with y(t)
9)          ω=ω + 1
10) return R
```

Fig.2. Watermark Insertion Algorithm: Embedding Process

- The curve which is to be hidden in the database is selected according to four parameters known as controlling points: $P_0(P_{0x},P_{0y})$, $P_1(P_{1x},P_{1y})$, $P_2(P_{2x},P_{2y})$, *and* $P_3(P_{3x},P_{3y})$.
- The embedding process includes hiding the coordinates of the points lying on the cubic curve in the selected tuples in the database.
- The tuple is selected based on the secret key and value of γ. For each tuple, the primary key of that tuple and secret key are concatenated and its hash is calculated. If the modulus γ of that hash value is zero then that tuple is selected to hide the curve points.
- *k-th* byte of the calculated hash value is normalized to a value between *0* and *1*. This normalized value is denoted by *'t'* and is used to calculate the coordinates of a point lying on the curve. The value of *'t'* is put into the equation of the cubic curve which gives the x and y co-ordinates of a point lying on the curve:

$$x(t) = (1-t)^3P_{0x} + 3t(1-t)^2P_{1x} + 3t^2(1-t)^3P_{2x} + t^3P_{3x}$$

$$y(t) = (1-t)^3P_{0y} + 3t(1-t)^2P_{1y} + 3t^2(1-t)^3P_{2y} + t^3P_{3y}$$

Two attributes of the selected tuple- A_i and A_j are used to hide the x and y coordinates calculated above by replacing the decimal part of A_i and A_j with the x and y coordinates respectively.

B. Extraction Process

- Four curve parameters: P_0, P_1, P_2 and P_3, which have been used to select the curve during the embedding process, are required to extract the hidden curve from the database.
- The tuple is selected based on the secret key and value of γ. For each tuple, the primary key of that tuple and secret key are concatenated and its hash is

calculated. If the modulus γ of that hash value is zero then that tuple has the hidden curve points.

- $i^{th}$and j^{th} attribute of the selected tuple- A_i and A_j are used to extract the hidden *x* and *y* coordinates from the decimal part of A_i and A_j respectively.
- The eleventh byte of the calculated hash value is normalized to a value between *0* and *1*. This normalized value is denoted by *'t'*and is used to calculate the coordinates of a point lying on the curve. The value of *'t'* is used in the equation of the cubic curve to compute *x* and *y* coordinates of a point lying on the curve:

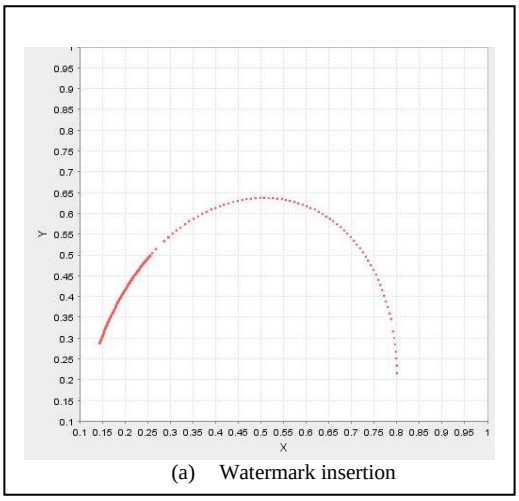
$$x(t) = (1-t)^3 P_{0x} + 3t(1-t)^2 P_{1x} + 3t^2(1-t)^3 P_{2x} + t^3 P_{3x}$$

$$y(t) = (1-t)^3 P_{0y} + 3t(1-t)^2 P_{1y} + 3t^2(1-t)^3 P_{2y} + t^3 P_{3y}$$

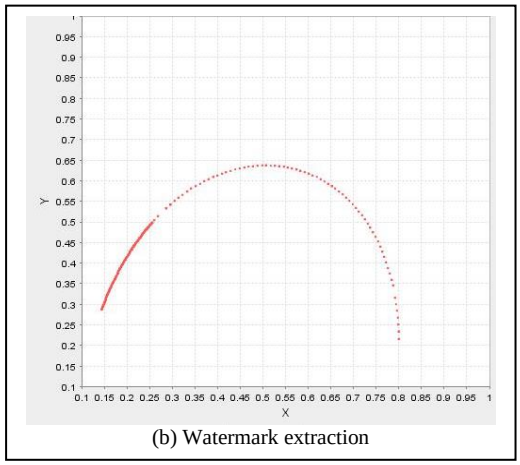
- Then the calculated curve points are compared with the extracted curve points. If they are same then this curve point is a match.

III. Results

The proposed watermarking scheme was applied on a database containing 10,000 records. A two-dimensional cubic curve was used for watermarking. For two-dimensional cubic curves, only two attributes were selected for inserting watermark in each tuple. Control points for cubic curve was used are $P_0(0.1,0.1)$, $P_1(0.2,0.8)$, $P_2(0.8,0.8)$, and $P_3(0.8,0.2)$.



(a) Watermark insertion



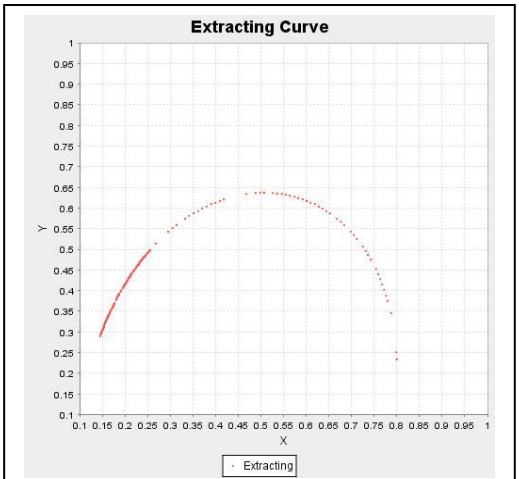
(b) Watermark extraction

Density control parameter γused here is *10* means only 10% tuples are selected for watermarking. Watermark extracted was exactly similar to watermark inserted.

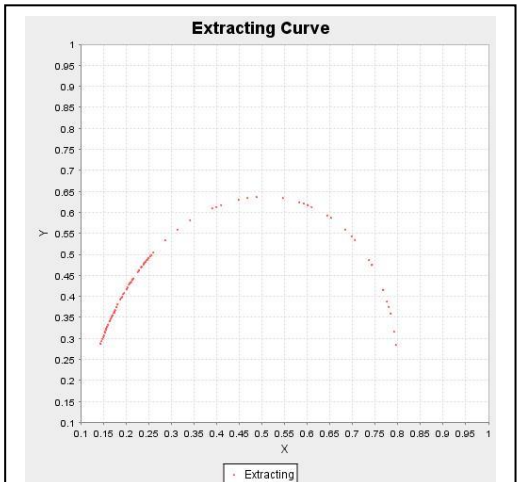
Attacks on watermarked database: Experimental setup is used to observe the effect of various types of database modification operations on watermark identification. Various type of operations which modify database contents are DELETE, UPDATE, and INSERT.

A. Deletion Attacks:

Effect of deleting some of the tuples from a database will not affect identification of extracted watermark. As we have inserted watermarks in only 10% of total tuples, deleting some of the tuples from database will only delete a very small number of those tuples having watermarking information. Deleting those tuples containing watermarking information will only eliminate some points on extracted cubic curve. Even if 90 percent of the points on the cubic curve are removed, the shape of the curve is still recognizable.



(a) 70% tuples deleted

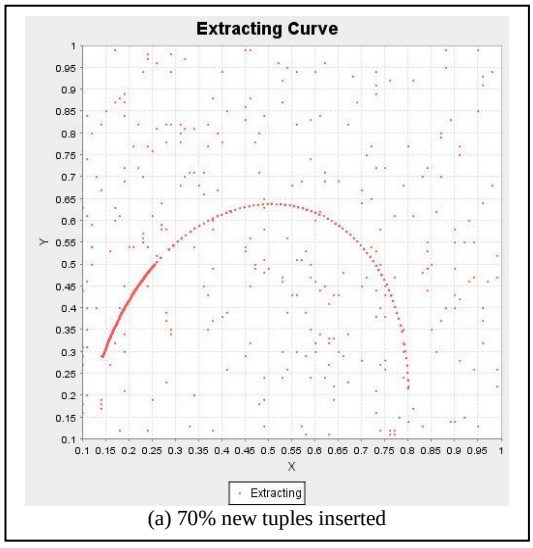


(b) 90% tuples deleted

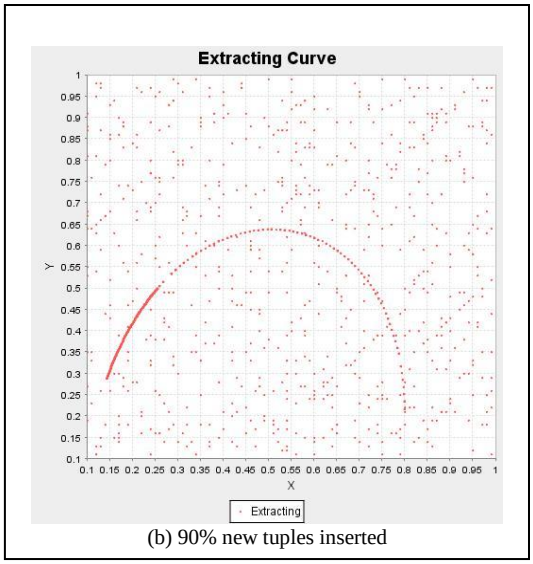
Fig.4. Watermark extraction results after deletion attacks

A. Insertion Attacks

Watermarking scheme is robust against insertion attack according to the algorithm. New records inserted actually do not have any watermark hidden in attributes. Watermark extraction process to extract watermark information from these tuples will extract invalid watermark which looks like noise in extracted pattern. The level of noise will increase as we increase the percentage of newly inserted tuples. But original tuples which contain watermark are not disturbed and all watermark information we have embedded earlier are extracted. Therefore, extracted watermark will contain all the watermark information along with some noise representing invalid watermarks detected in newly inserted tuples.



(a) 70% new tuples inserted

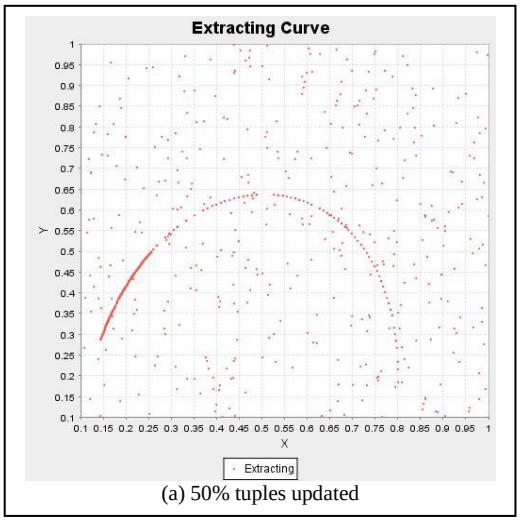


(b) 90% new tuples inserted

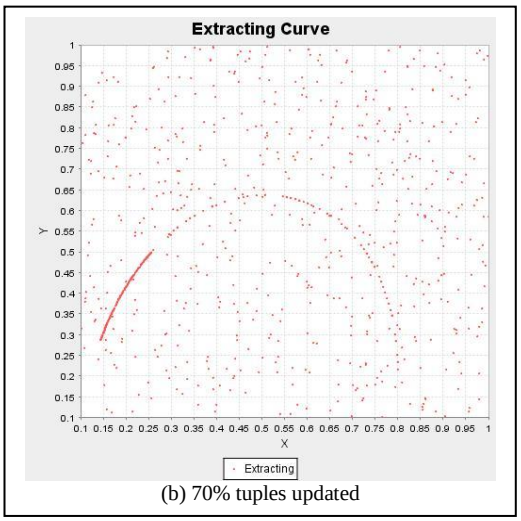
Fig. 5.Watermark extraction results after insertion attacks

C. Update Attacks:

Updating a tuple where a watermark is embedded will create a noise during the extraction process. As we have used small percentage of tuples for watermarking, chances of updating an attribute of a tuple which is watermarked are very low. So only a few tuples having watermarks will be updated and their watermark will not match during the extraction process. Thus, very few noise marks will be created when a fraction of tuples are updated. In extracted watermark, the shape of the curve can be recognized easily.



(a) 50% tuples updated



(b) 70% tuples updated

Fig. 6. Watermark extraction results after updation attack

Comparing the results with image-based algorithm for watermarking relational databases [9], it is observed that in image based algorithm 70-80 percent updation attack destroyes the embedded image beyond recognition. Where as in proposed method curve can be recognized even after updation of 90 percent of the tuples. We can easily distinguish between valid curve points and noise in extracted curve image. Similar results were observed in

case deletion attack. When 80 percent tuples are deleted from image based watermarked database, extracted watermark image is too much distorted. In proposed method, curve is still recognizable as deletion of tupples only reduces density of points on curve.

IV Conclusion

We presented a new approach using Bezier curves and discussed the insertion and extraction watermarking algorithms in details. Usually, other database watermarking algorithms alter properties of tuples. Later during the extraction phase, we count a number of tuples having matching property. Exact matching of the watermark is achieved if 100 percent watermarked information is extracted. It means that we are able to extract the correct watermark from all tuples where watermarks have been inserted earlier. If due to some alteration in the database, we are unable to extract complete watermark information, then a threshold is usually set on the percentage of matching watermarked tuples. Due to this, if the database is altered beyond the acceptable limits, we are unable to prove the existence of the watermark in the database. In the proposed scheme, extracted watermark is identifiable even if a small fraction of the watermark is extracted. Our goal has been to ensure that our watermarking approach is robust for Watermarking Relational Databases. Our approach has been discussed analytically in terms of robustness, however, the quantitative evaluation is to be done. In fact, evaluating watermarks for Relational Database is a challenge and requires further consideration. However, the persistency of the watermark after both malicious and benign updates, as a sub-problem, might be evaluated by acquiring access to a log of user queries on a particular database over a reasonably long period of time. Then run the log on the watermarked database and observed whether the watermark detection algorithm confirms the presence of the watermark in the database.

References

[1] R. Chamlawi, A. Khan, I. Usman, Authentication, and recovery of images using multiple watermarks, Computers & Electrical Engineering 36 (2010), 578–584.

[2] 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.

[3] R. Agrawal and J. Kiernan, Watermarking Relational Databases, Proc. VLDB'02, (2002), pp. 155-166.

[4] R. Sion, M. Atallah, and S. Prabhakar, On Watermarking Numeric Sets, Proc. IWDW, (2002), pp. 12-15.

[5] X. Niu, et al., Watermarking Relational Databases for Ownership Protection, Chinese Journal of Electronics (in Chinese), vol. 31, no. 12A, (2003), pp. 2050-2053.

[6] 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.

[7] 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.

[8] 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.

[9] Zhongyan Hu,Zaihui Cao, Jianhua Sun, An image-based algorithm for watermarking relational databases, In Proc of IEEE International Conference on Measuring Technology and Mechatronics Automation, 2009.

[10] Franco-Contreras J, Coatrieux G, Cuppens-Boulahia N, Cuppens F, Roux C. Robust lossless watermarking of relational databases based on circular histogram modulation. IEEE T. Inf. Foren. Sec. 2014; 9(3): 397–410.

[11] Deshpande A. and Gadge J., New Watermarking Technique for Relational Databases, IEEE International Conference on Emerging Trends in Engineering and Technology, Nagpur, 2009, PP 664-669.

[12] Shehab M., Bretino E., and Ghofoor A., Watermarking Relational Databases using Optimization Based Techniques, IEEE transactions on Knowledge and Data Engineering, 2008, vol 20, no 1, PP 116-129.

[13] Unnikrishnan K., Pramod K.V., Dynamic Prediction Based Watermarking for Temporal Relational Databases, International Conference on Data Science and Engineering (ICDSE)-2016.

[14] Madhuri V. Gaikwad, Roma A. Kudale, Robust Reversible Watermarking For Relational Database, IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE) 19-21 December 2016, AISSMS, Pune, India.