ISSN 2348 – 9928
IJAICT Volume 2, Issue 1, May 2015 Doi:01.0401/ijaict.2015.01.02 Published on 05 (06) 2015

REVERSIBLE WATERMARKING TECHNIQUE BASED ON TIME-STAMPING IN RELATIONAL DATA

Ms. S. Panimalar
Assistant professor,
Computer Science and Engineering,
Panimalar Institute of Technology,
Chennai, Tamilnadu, India

Dr. D. Srinath
Associate professor,
Computer Science and Engineering,
Panimalar Institute of Technology,
Chennai, Tamilnadu, India

Abstract— Ownership protection and controlling the copies of digital data have become very important issues in Internet-based applications. To solve this reversible watermarking is applied on relational database which is hard to find and robust against malicious attacks. In our proposed approach robust and efficient watermarking scheme is used to provide proof of ownership for the owner of a relational database. For generating watermark bits we use UTC (Coordinated Universal Time). The owner of the Relational Database embeds the watermark data in each selected row (in a numeric attribute) with the objective of having maximum robustness even if an attacker is somehow able to successfully corrupt the watermark in some selected part of the data set, the distortions in the original data are kept within certain limits, to preserve the knowledge contained in the data.

Keywords— Digital Data, UTC, Watermarking.

I. INTRODUCTION

The advancement of digital communications and the internet has opened new horizons in the social and business domain and has re-defined traditional perceptions of fields such as trade, banking and social welfare. Easy modification and reproduction of digital data (software, images, video, audio, and text) without leaving any trace of manipulation makes it very easy victim of piracy. Number of watermarking based solutions proposed so far for copyright protection of relational data. Watermark is a secret code embedded in digital contents [1]. This watermark can be extracted/ detected from the watermarked contents and can be used to establish the ownership of data.

Watermarking fails to prevent illegal copying but it can be an effective tool for establishing original ownership of pirated data. This discourages piracy and enables owners to prosecute copyright violators. Growing use of outsourced relational data, especially availability of relational data over the internet, demanded an effective mechanism for copyright protection so that owner of the data can identify pirated copies of their data. Relational data in particular is shared extensively by the owners with research communities. This paper focuses on numerical relational database. In some applications, especially in the medical, military, and legal domains, even the imperceptible distortion introduced in the watermarking process is unacceptable [2]. This has led to an interest in reversible watermarking, where the embedding is done in such a way that the information content of the host is preserved. This enables the decoder to not only extract the watermark, but also perfectly reconstruct the original host signal from the watermarked work [3].

Two categories, depending on application, distinguish watermarks: robust watermarks for ownership verification and fragile watermarks for tamper detection.The purpose of a robust watermark is to resist a variety of attacks and legitimate Users' data modifications, and categorically determines intellectual property. The purpose of a fragile watermark is for it to be damaged or destroyed by even the slightest data manipulation, thus determining categorically (and possibly localizing) any attack directed at the integrity of a digital object [5][4]. In the context of databases, the copyright protection is essential where it concerns sensitive data or data to be sold from a collecting institution A to an institution B (outsourcing), for uses such as data mining. Independently of the sale, institution A retains the copyright, while institution B holds the right to use, but not to sell the data to another institution. Unanimously, a good database watermarking technique should meet the following challenges: (i) imperceptibility: the embedded watermark should be invisible and the watermark insertion process should not degrade the data usability, (ii) robustness: the watermark should be robust against attack with the aim to destroy the watermark, (iii) security: the watermarking embedding process must use a private key for the security purpose, (iv) blindness: the watermarking detection process should not require the knowledge of original data and the watermark information. Due to differences between multimedia and database we cannot directly use any of the technique as it is for database, which developed for multimedia data[8][10]. These differences include:

- A multimedia object consists of a large number of bits, with considerable redundancy. Therefore, the watermark has more space to hide where as a database relation consists of tuples, each of which represents a separate object. So the watermark needs to be spread over these separate objects.
- The relative spatial/temporal positioning of various pieces of a multimedia object typically does not change. But, tuples may change with updates in database.
- Portions of a multimedia object cannot be dropped or replaced arbitrarily without causing perceptual changes in the object. Whereas, tuples may simply be dropped by delete operation in database.

The major contributions of this paper are presented in the following: We propose a watermark decoding algorithm which ensures that its decoding accuracy is independent of the usability constraints (or available bandwidth). As a result, our approach facilitates to define usability constraints only once for a particular database for every possible type of intended application. Moreover, it also ensures that the watermark introduces the least possible distortions to the original data without compromising the robustness of the inserted watermark. In the proposed system we implement a new approach to generate the watermark bits from UTC (Coordinated Universal Time) date time which is the primary time standard used to synchronize the time all over the world. A robust watermark algorithm is used to embed watermark bits into the data set of Database Owner. The watermark embedding algorithm takes a secret key (Ks) and the watermark bits (W) as input and converts a data set D into watermarked data set DW. A cryptographic hash function MD5 is applied on the selected data set to select only those tuples which have an even hash value. The Watermarking process includes Encoding and Decoding Phase. The Encoding phase consist of Data partitioning, Selection of data set for watermarking, Watermark embedding process .Decoding phase consist also these process to extract the Watermarked content. Edge detection Authentication is proposed as an alternative solution to text based for user authentication.

II. RELATED WORK

Work on database watermarking started in 2002 when Agrawal and Kiernan presented a robust watermarking scheme for databases [1]. The scheme focused on watermarking relational data with numeric attributes. It is assumed that these numeric attributes can tolerates small amount of modification.

Bit encoding algorithm for encoding the data. It allows us to convert the original string into ASCII then to binary digits. Then by using data padding we are inserting the 0's and 1's to the original encoded binary value and then it is send to the receiver. In receiver side first it will check the user authentication [2]. If the receiver is administrator means first he will check whether the data is modified or not by comparing the data sent by the sender admin and with the original data stored in the database. If the values in the two tables are matched then the receiver gets the original data by decoding process. If the values in the two tables are not matched then an alert box intimates the receiver that the data is modified. But here there no selection of tuple mechanism carried out.

The technique used ensures that some bit positions of some of the attributes of some of the tuples contain specific values which are algorithmically determined under the control of a secret key known only to the owner of the data. This bit pattern constitutes the watermark. Only if one has access to the secret key can the watermark be detected with high probability [3]. Detecting the watermark requires access neither to the original data nor the watermark, and the watermark can be easily and efficiently maintained in the presence of insertions, updates, and deletions. This paper shows that the proposed technique is robust against various forms of malicious attacks as well as benign updates to the data.

In lossless and exact authentication of relational databases is achieved via expansion on data error histogram. This reversible watermarking scheme possesses the ability of perfect restoration of the original attribute data from the untampered watermarked relational databases, thus guaranteeing a "clear and exact" tampered-or-not authentication without worry about causing any permanent distortion to the database Histogram expansion technique is used to reversibly watermark the selected nonzero initial digits of errors[4]. This technique keeps track of overhead information to authenticate data quality. However, this technique is not robust against heavy attacks (attacks that may target large number of tuples).

Multipurpose because it can be used for both watermarking(i.e., the same bit string is embedded and detected in everydatabase copy) and fingerprinting (i.e., a different bit string isembedded and detected in each database copy).It is robust against a range of attacks. The algorithm sorts the bits of each tuple in a secret order and selects some of its data bits to route the tuple to a specific watermark bit and one data bit to be marked by the value of the assigned watermark bit[5]. But here there is no error

correction mechanism adopted. They do not provide correct watermark recovery decisions in view of other types of attack.

Fragile zero watermarking for the authentication of numeric relational data. It does not embed any watermark in the original database. There are two main phases: 1) the watermark generation and certification phase and 2) the watermark verification phase. The watermark generation and certification phase focuses on the characteristics of the content of the subsets of numeric database values which are data set partitioning, watermark generation, watermark encryption, watermark certification and registration. The watermarking verification phase is the process of comparing the data set watermark registered at the certification authority and the data set watermark from suspicious data set[6]. Cost is more since there is no selection of individual square matrix.

In watermark embedding using the embedded secret key *K*, the tuples of relational database *R*are selected from the relational database *R* using one way hash function. Watermark *W*is embedded into selected tuples based on histogram shifting to generate the watermarked relational database $R'_{w.}$ After the watermark data are extracted completely, several copies of each watermark bit can be obtained. Then, the majority voting mechanism is applied to determine the final watermark bit[7]. Once the watermark data *W''* have been reconstructed successfully, they are used for verification. But one way hash function alone is used for tuple selection so that number of tuples watermarked is more.

In the paper [8] they are inserting same watermark at different places so, there is a less chances of it to get attacked and if so, it is comparatively easy to extract the original watermark. They propose a method, based on image as watermark and this watermark is embedded over the database at two different attribute of tuple, one in the numeric attribute of tuple and another in the date attribute's time (seconds) field. A very important assumption regarding database watermarking is that small changes in LSB of a numeric attribute are tolerable within certain precision range. They have removed a keyed hash function from the algorithm [3] and made it simple to insert and detect though we compromise with the security.

III. PROPOSED WORK

Data Partitioning

The dataset D is a database relation with scheme D = (PK, A0 ..., An−1), where PK is the primary key attribute and A0 ..., An−1 are n other attributes. The partition algorithm divides the dataset D Into m non- overlapping partitions namely $\{P_0 \ldots, P_{m-1}\}$ such that for any two partitions there won't be same tuples. Moreover, the partition sets must be non-empty and collectively exhaustive to D such that $P_0 \cup P_1 \cup ... \cup Pm-1=D$. The data partitioning algorithm partitions the dataset into logical groups by using data partitioning algorithm. Partitioning is based on a secret key Ks and a cryptographic hash function Message Digest (MD5).

[Hash function.] A hash function H maps a variable-size input Y to a fixed-size string h, called the hash value h, as:

$H: Y \rightarrow h$ (1)

For each tuple r€D, the data partitioning algorithm computes Message Authentication Code (MAC) in order to assign tuples to the partitions using a hash function H as

$par(r) \leftarrow H(Ks||H(r.PK||Ks)) \bmod m$ (2)

Where r.PK is the primary key of the tupler,

H () is a secure hash function and|| is the concatenation operator. Algorithm 1 lists the steps of data partitioning process.

Input: Dataset D, Secret Key Ks, Number of partitions m

Output: Data partitions $P_0 \ldots, P_{m-1}$

1: for each Tuple r€D do

2: par(r)← H(Ks||H(r.PK||Ks)) mod m

3: insert r into Spar(r)

4: end for

5: return $P_0 \ldots, P_{m-1}$

Algorithm 1: Data Group Partitioning

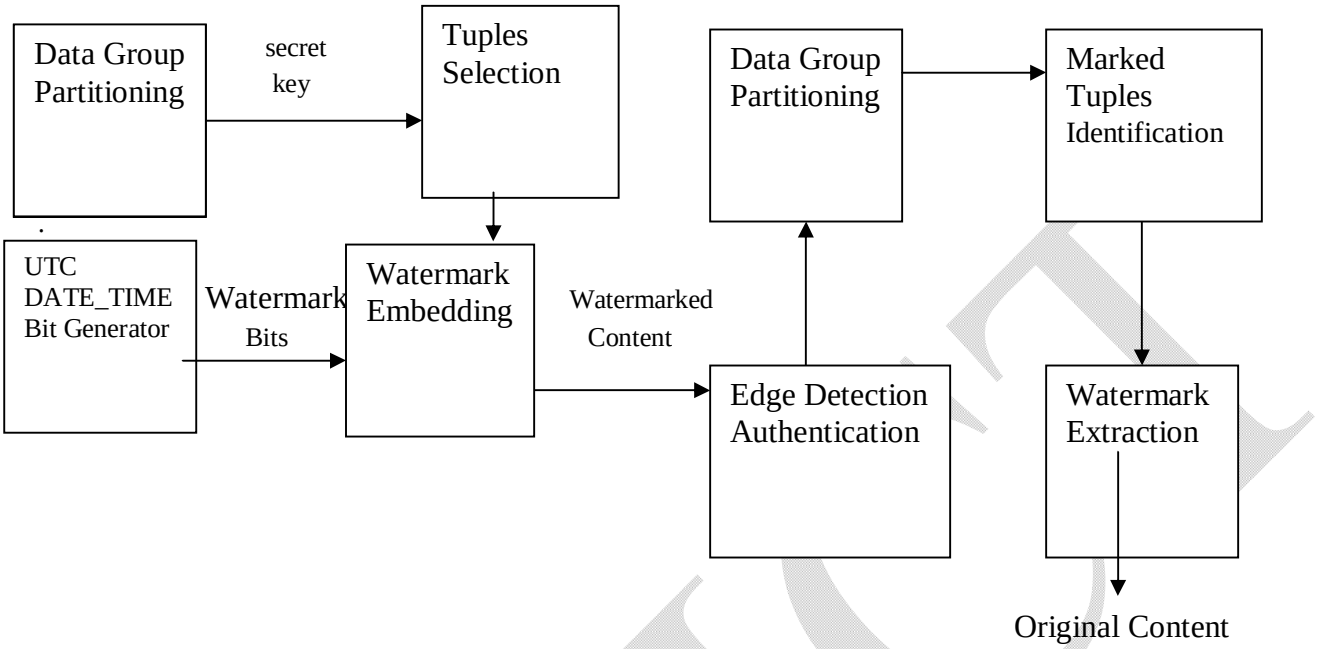
Tuple Selection

The following steps are done to select the tuples for watermarking.

ThresholdComputation

Inthisstep,a thresholdiscomputedforeach

Fig 1 : System Design



attribute. If the value of all attribute of a tuple is above its respective computed threshold ,it is selected for water marking.

[Data selection threshold] Given a dataset D, a function fis used to calculate data selection threshold for constructing DT from D.

$f:D \rightarrow DT$(3)

The data selection threshold for an attribute is calculating using equation (4).

$T = c * \mu + \sigma$(4)

Whereμ is the mean, σ is the standard deviation of the values of an attribute Ain D, and c is the confidencefactor with a value between 0 and 1. The confidencefactor c is kept secret to make it very difficult foran attacker to guess the selected tuples in which thewatermark is inserted. In The database, all same columns of the tuples R present in the same partition group P areadded to get the mean μ and the square root

is taken for that result to get the standard deviation σ.In this way, we reduce the number of tuples to bewatermarked. As a result, data distortions during watermarkembedding are minimized. Algorithm 2 depictsdifferent steps of this phase.In order to ensure that the tuples, for which if any of the attribute value is above T, are included in the watermarked tuples set, a union of tuples in this phase is taken.

Input: Data partitions P_0 …, $P_{m-1}$and c
Output: Dataset DT
1: **for** i = 0 to m − 1 **do**
2: **for** each Attribute A€Si**do**
3: Compute μ and σ on A
4: Calculate T using equation (4)
5: **end for**
6: **end for**
7: **return** DT← □R>T

Algorithm 2: Data selection using threshold

Hash Value Computation

In this step, a cryptographic hash function MD5 is applied on the selected dataset to select only those tuples which have an even hash value. And it further reduces the number of to-be-watermarkedtuples to limit distortions in the dataset. The dataset DTis used to select tuples with even hash values and put them in

the datasetDT. The steps involved in this phase are illustrated in Algorithm 3.

Input: Dataset DT, Secret Key Ks
Output: DT'
1: **for** each r€DT**do**
2: Even Value(r) = H(Ks||r.PK) mod 2
3: **if** Even Value(r) == 0 **then**
4: insert r into DT'
5: **else**
6: don't consider this tuple for watermarking
7: **end if**
8: **end for**
9: **return** DT'

Algorithm 3: Even Hash Value

The dataset DT, consisting of tuples, is the subpartof the dataset D and is not physically separated fromthe rest of the parts of D.

Generation of watermark bit

In this step, date time stamp UTC is used for generating the watermark bit W.A watermark generation function "g"transforms an alpha-numeric string λ to an l-bits long binary bit string {b0b1b2...bl−1}.

g: λ → {b0b1b2...bl−1} (5)

The watermark generating function" g" takes date-timestamp as an input and then generates watermark bits{b0b1b2...bl−1} from this date-time stamp. The date-timestamp "might" also help to identify additive attacks inwhich an attacker wants to re-watermark the dataset.To construct a watermarked dataset, these watermarkbitsW are embedded in the original dataset by using the following watermark embedding algorithm.

Watermark embedding algorithm

The watermarking algorithm uses multi-bit watermarking property and is scalable to any number of attributes. For a better Where i is the column in each row, MSB is Most Significant Bit.

Edge detection Authentication and Watermark Decoding

Edge detection Authentication is proposed as an alternative solution to text based. It is mainly depends on images rather than alphanumerical. The main argument here is that pass-images from the challenge set and then he/she will be authenticated users are better at recognizing and memorizing pictures. During Registration phase Admin has to provide some images to the user. In the registration phase the user is supposed to choose the pass-images for the verification phase. That image has to be Stored in Server For that Specific User. During Login phase Admin has to converting the raw image to a gray scale followed by Edge detection image.

The idea here is the user will have a challenge set which contains decoy and pass-images. The decoy images are randomly generated by the scheme during the verification process.On the other hand, pass-image will be the users selected

understanding, we assume that the partition set Si in dataset DT' contains a single attributeAi€S_i. The encoding function generates bits b0, b1, b2…, bl−1, where l is the length of the watermark. Since our technique embeds watermark bits b0, b1, b2…, bl−1, in each partition of Si; therefore, the watermark bits can be recovered from the remaining partitions if the watermark is removed from a particular partition Si.

Input: Datasets D, DT', Watermark bits W, Secret Key Ks
Output: Watermarked Dataset Dw
1: Dw=D
2: for each row r in DT' do
3: for each i in r
4: if b==1 then
5: s=r.Ai%1.5
6: if(s==0.5 || s==-0.5) then
7: append 1 in MSB of r.Ai
8: else
9: append s in MSB of r.Ai
10: end if
11: else
12: s=r.Ai%0.5
13: if(s==0.5 || s==-0.5) then
14: append 1 in MSB of r.Ai
15: else
16: append s in MSB of r.Ai
17: end if
18: end for
19: insert r into Dw
20: end for
21: return Dw

Algorithm4: Watermark Embedding

images. Basically authentication is simple; a legitimate user needs to correctly identify pass-images from the challenge set and then he/she will be authenticated.

Watermark Extraction process in the Decoding phase. The Watermarked Content has to be extracted only by legitimate user to give the proper ownership. If the User ownership content is matched by the Admin generated content Decoding process has to done. Otherwise it's not done.

IV. OBSERVATION AND RESULT

The major motivation of designing experiments is to prove the objective which guarantees that the decoding accuracy is obtained. We have selected a subset of 400 tuples from a real-life dataset that shows the employee information of an organization. The watermark length is 29 bits (as the conversion of UTC data-time to binary string yields a bit string consisting of 29 bits).

The number of partitions m=4 is used. The experiment has been performed on a server that has Pentium(R) Dual-Core CPU 2.10GHz with 4GB of RAM.Fig.2 plots the proportion of marked tuples that must have the correct watermark value for successful detection. The X-axis varies the percentage of tuples marked. The Y-axis varies the ratio of correct marks/total marks. It shows that in the paper "Watermarking Relational Database" the required proportion of correctly marked tuples decreases as the percentage of marked tuples increases.

This proportion also decreases as the number of tuples in the relation increases whereas in our paper the proportion of correctly watermarked tuples is always 1 i.e. every selected tuples for watermarking are being correctly watermarked.The Fig.3 plots the probability of successful attack. The x axis varies % of tuple changed and y axis varies the probability of successful attack. In this paper "Watermarking Relational Database" the tuples changed due to successful attack is being higher than in our proposed work.

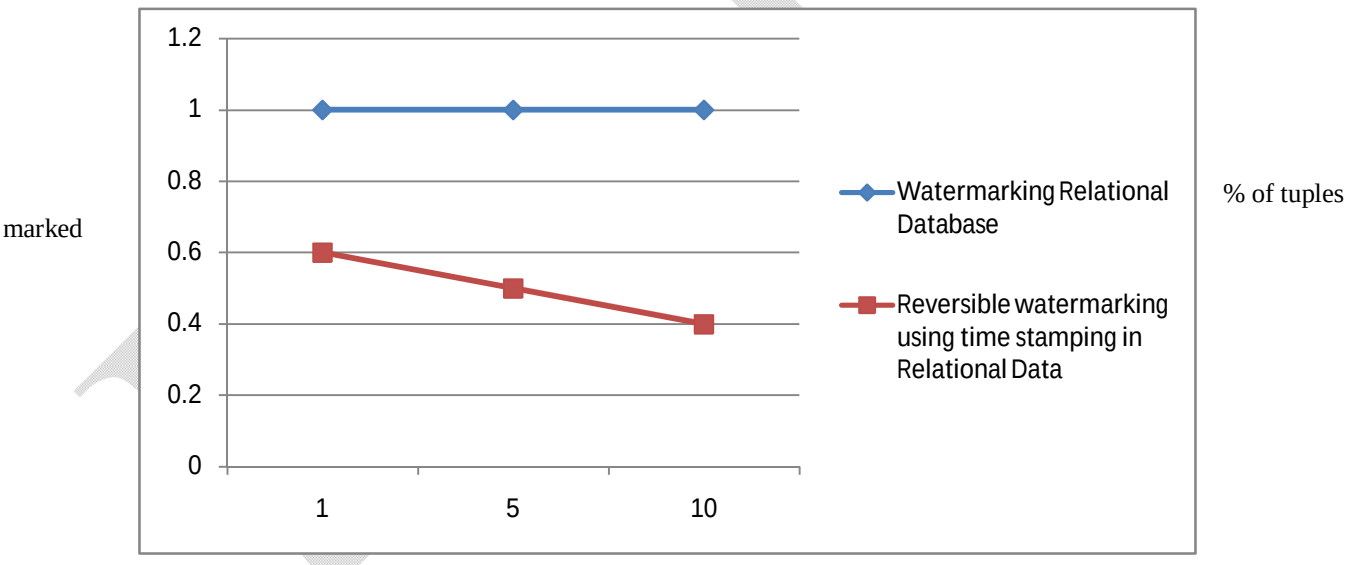


Fig 2 : Proportion of correctly marked tuples needed for detectability

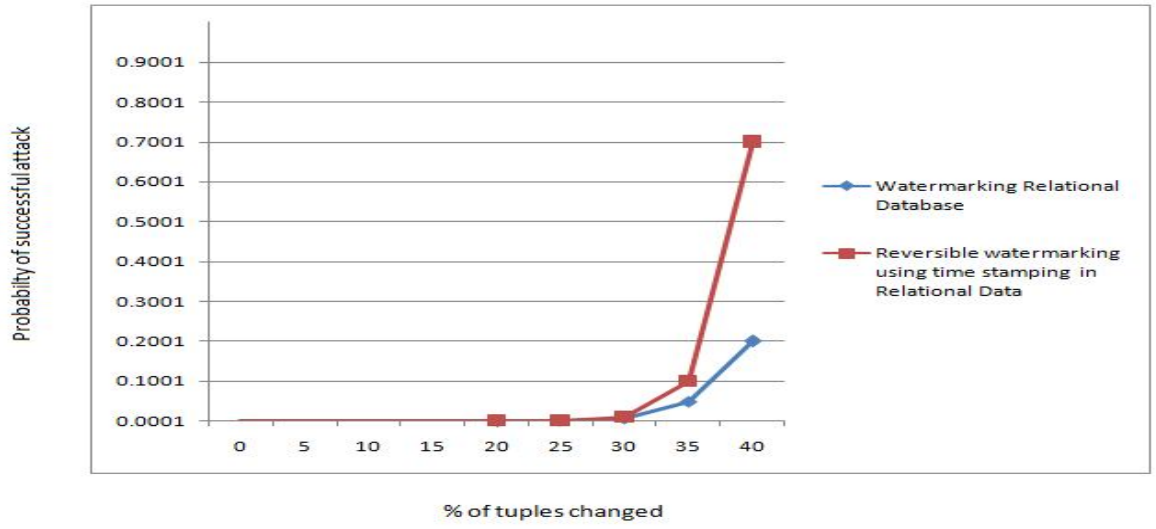


Fig 3 : Probability of a successful attack

V. CONCLUSION

Thus the Reversible watermarking Technique based on Time-Stamping in Relational data allows recovery of original data and users are authenticated through Edge Detection Authentication which is a safe method. The distortions are minimized in the watermarked content and the ownership rights are protected for the database owner. In our future enhancement we extend our approach to non-numeric domains.

References

[1] RakeshAgrawal and Jerry Kiernan, "Watermarking Relational Databases", Proceedings of the 28th international conference on Very Large Data Bases. VLDB Endowment, 2002, pp. 155–166.

[2] Loganayaki A., "Robust Watermarking For Relational Database", International Journal of Communication and Computer Technologies Volume 01 – No.41, Issue: 05 May 2013.

[3] RakeshAgarwal, Peter J. Haas, Jerry Kiernan, "Watermarking relational data: framework, algorithm and analysis", The VLDB Journal (2003)/ Digital Object Identifier (DOI) 10.1007/s00778-003-0097-x.

[4] Y. Zhang, B. Yang, and X.-M. Niu, "Reversible watermarking for relational database authentication",Pro.of the 1st international conference on Forensic applications and techniques in telecommunication, information, and multimedia and workshop. ICST 2008, p. 24.

[5] TheodorosTzouramanis, "A Robust Watermarking Scheme for Relational Databases", 6th International conference on Internet Technology & Secured Transactions,11-14 Dec2011.

[6] LancineCamara, Junyi Li, Renfa Li and WenyongXie, "Distortion-Free Watermarking Approach for Relational Database Integrity Checking", Hindawi Publishing Corporation Mathematical Problems in Engineering Volume 2014, Article ID 697165, 10 pageshttp://dx.doi.org/10.1155/2014/697165 .

[7] Chin-Chen Chang, Thai-Son Nguyen and Chia-Chen Lin, "A Blind Reversible Robust Watermarking Scheme for Relational Databases", Hindawi Publishing Corporation The Scientific World Journal Volume 2013, Article ID 717165.

[8] BrijeshB. Mehta, UdaiPratapRao, "A Novel approach as Multi-place Watermarking forSecurity in Database", Int'l Conf. Security and Management |SAM'11|,2011.

[9] R. Caldelli, F. Filippini, R. Becarelli Reversible watermarking techniques: an overview and a classification EURASIP J. Inform. Security, 2010 (2010), pp. 1–19.

[10] [10]A.M. Alattar, Reversible watermark using the difference expansion of a generalized integer transform, IEEE Trans. Image Process., 13 (8) (2004), pp. 1147–1156.