Defence Science Journal, Vol. 65, No. 3, May 2015, pp. 226-232, DOI : 10.14429/dsj.65.8444

Watermarking Categorical Data : Algorithm and Robustness Analysis

Vidhi Khanduja[!,*], Shampa Chakraverty[!], and Om Prakash Verma[#]

[!]Deptt. of Computer Engineering, Netaji Subhas Institute of Technology, Delhi-110 078, India
[#]Deptt. of Computer Science and Engineering, Delhi Technological University, Delhi-110 042, India
[*]E-mail: vidhikhanduja9@gmail.com

ABSTRACT

The importance of watermarking digital databases has increased by leaps and bounds due to the high vulnerability of digital assets to piracy attempts when they traverse through the internet. To deter piracy, we propose a robust watermarking scheme for relational databases containing categorical data that resolves ownership issues. We propose a three-level security strategy. Firstly, the watermark is itself made secure using playfair cryptographic algorithm. Secondly, the database is securely partitioned using a primary key independent hash partitioning technique. This step virtually reorders the tuples before embedding. Thirdly, we entail a secret key based embedding process to ensure security. Linear feedback shift registers are implemented to generate pseudorandom numbers which selects different watermark bit index for each partition. The process of embedding does not produce any distortion in the database. Hence it is suitable for databases with categorical attributes containing sensitive information that cannot tolerate perturbations. Each watermark bit is embedded multiple times into different partitions. This makes the scheme highly robust against various attacks. The technique is proved by experimentally, and by theoretical analysis to be extremely robust. Experimental results show that it is 400 per cent resilient to subset addition attack, 100 per cent resilient to subset alteration attack, and 96 per cent resilient to tuple deletion attack. We prove analytically the resilience of the proposed technique against invertibility and additive attacks.

Keywords: Information security, database protection, digital watermarking, categorical data, copyright issues

1. INTRODUCTION

In this new era of digital innovations, data plays a pivotal role. Owing to the increased use of databases, it has become increasingly vulnerable to copyright and piracy threats. With this, the need to protect them arises. Technological protection methods[1] (TPMs) backed by legal anti-circumvention measures offer a cost-effective solution to database protection. TPMs include the technologies that are used to control access to copyright content or to prevent users from copying protected content. Watermarking of digital databases is one such TPM that has emerged as an effective means to protect shared and outsourced data from unauthorized access.

Robust digital watermarking techniques aim at ownership protection. Normally, robust watermarking algorithms introduce small changes in the data throughout the database, *albeit* in manner that it does not render the data completely unusable. However, there are databases that contain sensitive information such as those used in medical and military applications where even the slightest distortions cannot be tolerated. In particular, categorical data assume a fixed and limited number of values and even a small change can lead to a change in category itself. It is therefore a challenge to implement robust watermarking in a database that contains predominantly categorical data. In this paper, authers proposed a distortion-free, robust watermarking technique for watermarking categorical attributes to resolve ownership verification.

2. RELATED WORK

Watermarking databases mainly resolves two important issues: proof of ownership and tamper detection. A robust technique provides ownership proof whereas a fragile watermark detects any tamperness in the database. Agrawal[2], *et al.* proposed robust watermarking technique to insert marks in the numeric attributes. They embedded the watermark by resetting the algorithmically determined least significant bit of a specific attribute. This technique is not resilient to bit-based attacks. Farfoura[3], *et al.* proposed time-stamping protocol to resolve additive attacks. The embedding process was reversible so that original values can be regained.

Shehab[4], *et al.* and Khanduja[5], *et al.* utilized statistical properties of the data-set to embed watermark into the numerical attribute. They framed watermarking problem as a constrained optimization problem and discuss the efficient techniques to solve it. Several other notorious works is proposed in literature to watermark numeric attributes[6-9].

Watermarking in non-numeric attributes requires different approach as bit-flipping does not work here[10,11]. Watermark is embedded using non-repetitive nature of occurrence of a vowel in a non-numeric attribute value[10]. A vowel is appended to the selected attribute depending upon the key value generated for a tuple.

In robust database watermarking schema, there is an underlying assumption that all watermarked attributes can

Received 08 February 2015, revised 15 May 2015, online published 29 May 2015

tolerate small perturbations. However, same techniques are not applicable in case of categorical attributes. Categorical attributes take a finite set of distinct values. Altering their values can make the data meaningless.

Sion[12], *et al.* proposed a technique to watermark categorical data by modifying their values within allowable limits based on watermark. Each watermark bit is embedded into selected tuples. Value of the selected attribute is secretly changed to value which lies in discrete domain of that attribute. For example, the colour of an item may be changed from blue to red. Finally, majority voting is used to extract the watermark bits correctly.

Apart from robust watermarking techniques, Li[13], *et al.* devised a distortion-free fragile watermarking technique in which all the tuples in a database relation are divided into groups and the watermark is embedded and verified in each group independently depending upon group hash values. However, this technique is designed for tamper detection and does not resolve ownership verification issues. Other distortion free techniques proposed in literature include[14-16]; Guo[14], *et al.* proposed the technique to embed and verify watermark group by group independently according to some secure parameters. In this scheme two sets of watermarks are embedded into LSB's of the attributes of a tuple within a group to localize modifications made to the database. The core idea of Khan[15], *et al.* is to generate watermark based on local characteristics of database relation like frequency distribution of digit, length and range.

Camara[16], *et al.* partitions the database into different set of square matrices. Watermark is then generated using determinant and the minor of the generated square matrix. This watermark is not embedded but registered with trusted third party for integrity verification.

The techniques proposed by Guo[14], *et al.,* Khan[15], *et al.,* and Camara[16], *et al.* can be used to identify if the database had been tampered with but cannot be used to claim ownership as the watermark can be easily spificated with the slightest attack.

In this paper, authors proposed a robust watermarking technique for categorical attributes to resolve ownership verification. A distortion-free technique is proposed that embeds watermarks by changing the positions of tuples within the database in a secure manner. Single watermark bit is embedded repeatedly into secretly decided partitions. This adds security to the algorithm in comparison to the work of Li[13] *et al.* where two consecutive tuples are compared. Moreover, to prove ownership rights, the robust technique is applied in contrast to fragile technique of Li[13].

3. PROPOSED WATERMARKING TECHNIQUE

The proposed watermarking system consists of two subsystems: the Watermark Encoder and its corresponding Decoder. Suppose R is a relation whose schema is $R(A_0, A_1, \ldots, A_{Na-1})$. There are N_t number of tuples in R and N_a is number of attributes in relation R. N_a includes categorical attributes such as social security number, gender, etc.

3.1 Watermark Encoder

The watermark dncoder first prepares and then embeds the prepared watermark into the database R as shown in Fig. 1. It shows the flow of information through four phases of encoder.

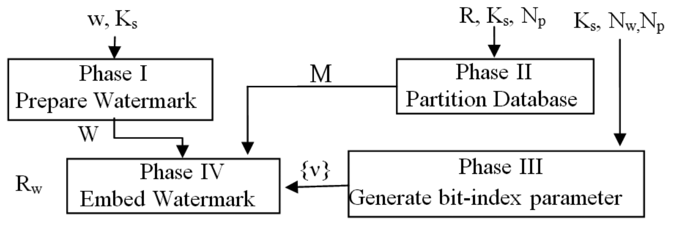


Figure 1. Block diagram of watermark encoder.

3.1.1 Prepare Watermark

This phase securely prepares the watermark. Watermark w is first selected by owner of the database reflecting the owner's identity in some unique way. This watermark is made secure by generating a playfair cipher[17]using secret key K_s. The playfair cipher possesses distinct secure characteristics. Firstly, the cipher, being digraph, destroys single-letter frequencies. The monographic analysis of frequencies is destroyed. Secondly, for smaller data as required in our application, it is almost infeasible to break the code. Digraph ciphers halve the number of elements available for frequency analysis. For example if message contains 34 letters then, there are 17 digraph substitutions rather than 34 single-letter substitutions.

The cipher-text thus prepared is converted into bits to get the final N_w-bit watermark. This step securely encapsulates the identity of database's copyright holder in the form of a watermark and acts as first level of protective shield to protect the entire embedding process.

Authors used playfair algorithm for watermark preparation to avoid high code complexity. However, note that this can be replaced by a stronger encryption scheme such as advanced encryption standard (AES) to enhance security.

3.1.2 Partition Database

This phase creates N_p virtual partitions of the database R. The parameter N_p is secretly decided by the owner of the database.

Firstly, derived primary key p_k of all the tuples of R is calculated individually using Eqn (1) that acts as a substitute for the primary key[18].

$$p_k = MSB(t.A_1)||MSB(t.A_2) \quad (1)$$

where MSB $(t.A_1)$ and MSB $(t.A_2)$ are the most significant bits of two of the candidate attribute in relation R and || is the concatenation operator. These candidate attributes are selected by owner such that any change in these MSBs will violate usability constraints. Further, to defeat linear transformation attack, a normalization of these attribute values is carried out using a common divider that is applied to all items.

Next, hash of each tuple $t \varepsilon R$ is computed using Eqn (2). Here, one-way hash functions are used along with key and thus are referred as message authentication code (MAC)[19]. Inclusion of key makes partitioning phase more secure. For any tuple t, its hash value[6] $H(t)$ is given by:

$$H(t)=Hash\ (K_s||p_k||K_s) \tag{2}$$

where K_s is a secret key selected by owner and || is the concatenation operator. Lastly, each tuple t is assigned a partition M_{Id}. We define partitioning index, $I_d(t)$ as:

$$I_d(t) = H(t) mod\ N_p \tag{3}$$

Depending on hash value of a tuple, its partition is decided. Thus, tuples in a partition are out of order. This means any two consecutive tuples may or may not be in same partition. It is difficult for an attacker to guess the tuple-to-partition assignment. This phase acts as a second level of protective shield by virtually reordering the tuples before embedding the watermark.

3.1.3 Generate sequence of bit-index parameter

This step uses cryptographic pseudorandom sequence generator (G)[19] to generate a sequence of bit-index parameter. G is defined as a procedure that maps a random seed to pseudorandom string. The random seed is a binary string-based on which sequence is generated. G is seeded with the hash of the secret key K_s and generates v where $v=\{v_1, v_2, ..., v_{Np}\}$ is set of computationally infeasible sequence of numbers.The series of numbers generated by G decides the value of secret parameter for each partition. The output of G is given by

$$\mu = G\ (hash\ (K_s)) \tag{4}$$

Linear feedback shift registers are implemented to generate pseudorandom numbers. Since the hash of the key is seeded, the generated sequence will be longer enough to satisfy the mapping of numbers to points. However, if needed a new sequence can be generated using another seed value. Now, to ensure that the value lies within range $1 <= v_{Id} <= N_w$, we calculate v_{Id} using Eqn (5).

$$v_{Id} = (\mu_{Id} mod\ N_w)+1 \tag{5}$$

For any tuple t_j ε $\{M_{Id}\}$, v_{Id} is its secret bit-index parameter, which tells the index of the watermarking bit to be embedded. For each partition, a different v_{Id} is calculated. The idea is to split database into small groups of tuples such that single watermark bit is embedded in each group. Thus, for erasing a watermark, an attacker needs to correctly guess the secret key K_s selected by the owner of the database and then guess the cryptographically generated bit sequence output by the G. This security strategy ensures that it is indeed difficult for an attacker to correctly guess v for a particular partition and likewise for all partitions of database. Thus, acts as third level security shield for the database.

3.1.4 Embed Watermark

This process embeds the prepared watermark W by re-arranging the tuples partition-wise. The core logic for exchange is that if the watermark bit at the selected bit index is one, all the tuples are physically re-arranged in descending order according to their hash values calculated using Eqn (2) and vice versa. This process continues for the tuples in all the partitions. Since the number of watermark bits is very less as compared with number of partitions virtually created, each watermark bit is embedded multiple times in different partitions depending on $\{v\}$.

The watermarked relation obtained after embedding of watermark is referred as R_w. It may be noted that there is no alteration in the database. Since the watermark is embedded purely by reordering tuples. Hence, the proposed technique is entirely distortion free.

3.2 Watermark Decoder

Let Relation R^* defines the maliciously altered watermarked relation that the owner receives and successfully extracts watermark bits. R^* is equivalent to R_w; if not perturbed. Figure 2 illustrates various phases in watermark decoder. During detection process, the first three phases of watermark encoder are followed. Once the database is partitioned and bit-index parameter $\{v\}$ is selected, extract watermark phase is executed.

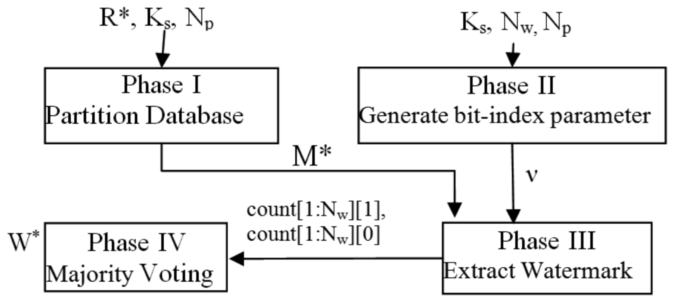


Figure 2. Various phases of the watermark decoder.

3.2.1 Extract Watermark

The algorithm extracts the watermark bits from the suspected relation R* as illustrated in Fig. 3.

In the algorithm *Extract_Watermark*(.) variables $count[v_{Id}][0]$ and $count[v_{Id}][1]$ stores the number of times zero and one is extracted for v_{Id} watermark bit. It accounts for entire watermark length N_w. In a partition, we compare the hash of all the tuples in that partition. If more than half of the tuples are found to be arranged in descending order according to their hash values, bit extracted is one and we increment $count[v_{Id}][1]$ by one [line 3,4,5]. Else, $count[v_{Id}][0]$ is incremented by one [line 7]. Similarly all the watermark bits are extracted from N_p partitions. It may be noted that the process of detecting the watermark is blind as it does not require original database R.

```
Algorithm: Extract_Watermark(.)
1. Initialize $count[1:N_w-1][0]=count[1:N_w-1][1]=0$
2. For each partition $M_{Id}^* \varepsilon R^*$ do
3.     Check the arrangement of all the tuples in $M_{Id}^*$ as per their hash values.
4.     If hash of majority of tuples in a partition are arranged in descending order
5.         $count[v_{Id}][1]=count[v_{Id}][1]+1$
6.     Else
7.         $count[v_{Id}][0]=count[v_{Id}][0]+1$
8.     End if
9. End for
```

Figure 3. Algorithm for extracting the watermark from suspected watermarked database *R.**

3.2.2 *Majority Voting*

Since each watermark bit is extracted multiple times, majority voting is applied to extract the final embedded bits of the watermark. Let N_i be number of times the i^{th} watermark bit is extracted. Considering, number of tuples in database as N_t, number of partitions; N_p and length of watermark; N_w. Number of times each watermark bit is embedded in entire database is equal to

$$N_i = \lfloor N_p / N_w \rfloor \tag{6}$$

For each marked bit b_i, we count the number of ones and zeros extracted. If fraction obtained by dividing count of ones with N_i is greater that detection parameter[3] τ then the extracted bit is considered one else zero. The τ detects majority threshold and its value lie between 0.5 and 1. If number of ones and zeros for a particular bit is zero, then we assign $w^*[i] = -1$ and that bit is ignored.

3.3 Comparison with other Existing Techniques

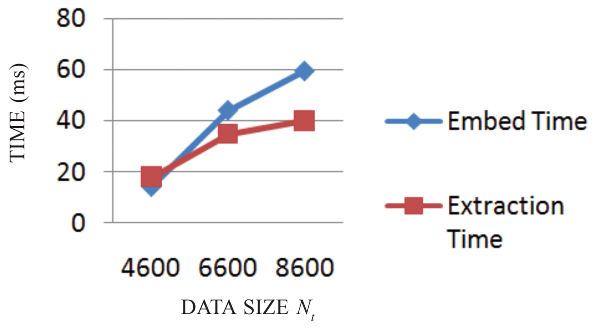
Authors compared their work with two major contributions available in literature in watermarking categorical data. Sion[12] has proposed robust watermarking scheme while fragile watermarking technique is proposed by Li[13]. Authors summarises key features in Table 1.

4. EXPERIMENTAL RESULTS AND DISCUSSION

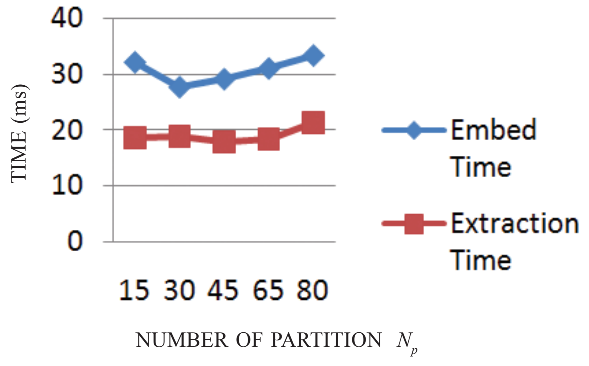
To perform the experiments authors used OS X version 10.10.1 with 1.3GHz Intel core i5 processor and 4 GB 1600 MHz DDR3 RAM. The proposed algorithm has been tested and evaluated on a national geochemical survey database of the US[20]. This data comprises a complete, national-scale geochemical coverage of the US and provide context for a wide variety of studies in the geological and environmental sciences. Database contains 287 attributes of text, numeric and categorical data types. Fields characterising soil samples like color of the sample, grain size, horizon, medium, geological source etc. take finite set values. Such attributes contribute to categorical attribute set. There are 59 categorical data attributes in the database. We have used MySQL for managing the database and PHP for implementation of the algorithm and generating test results.

4.1 Computational Cost

Figure 4 illustrates the experimental results on variation of execution times with (a) Data size and (b) Number of partitions. In Fig. 4(a), we observe that the extraction process consumes less time as compared with the embedding process. The extra time is required to save the reordered tuples back into the database. An average of 1million records can be watermarked in 6.73 s.



(a)



(b)

Figure 4. Execution time dependency on (a) Data size, N_t (b) On number of partitions, N_p.

Table 1. Qualitative Comparison of proposed technique with Sion[12], *et al.* and Li[13], *et al.*

Proposed Technique	Sion [12], *et al.*	Li[13], *et al.*
Robust watermarking technique is proposed to establish ownership.	Robust watermarking technique is proposed to establish ownership.	Fragile watermarking technique is proposed to ensure integrity of database.
Primary key independence. Technique is applicable to databases where Primary key is not defined.	Primary key dependence.	Primary key dependence.
Three levels of security 1. Watermark is cryptographically made secure and then embedded. 2. Secret key- based Data Partitioning. 3. Secure watermark bit selection.	Certain tuples are selected for embedding watermark.	No security mechanism while inserting watermark.
Watermark is selected by the owner – can be signature, voice, video etc. representing owner's identity.	No discussion on watermark. Watermark taken as bit stream.	Watermark necessarily created from partition.
Distortion-free technique	Altered the database.	Distortion-free technique
Highly resilient towards attacks experimental results shown in section 4.2.	Less resilient as compared to our technique (section 4.2).	Not a robust technique.

Figure 4(b) shows results of experiment to check the execution time for the combined embedding and extraction process by keeping the data size constant at 4600 records and varying N_p from 15 to 80 partitions. We observe that the execution time shows negligible dependency on N_p, as also expected.

It may be noted that the embedding process is carried out only once before the database is distributed. Similarly, in case of ownership disputes, the extraction process is also carried out once. Therefore, with proof of ownership as the main goal of database protection, the robustness of watermarking is a more critical issue than execution times. In the following subsections, we illustrate experimental results of tests conducted to prove the resilience of our proposed technique against various attacks. Table 2 shows the values of various simulation parameters used in these experiments. Authors implemented secure hash algorithm (SHA-1) for computing hash of the tuple to attain better security. It produces 160-bit hash value and is most widely used hash algorithm till date.

Table 2. Simulation parameters

Parameters	Symbols	Value
Secret key	K_s	'secret'
Tuple count	N_t	5000
Partition count	N_p	50
Watermark length	N_w	7
Hash type	H	SHA-1

4.2 Robustness Analysis

Authors assessed robustness of the algorithm by evaluating probability of false hit and false miss.

4.2.1 False Hit

Authors defined false hit rate P_{FM} as the probability of detecting a valid watermark from a non-watermarked relation. Lower P_{FM} indicates highly robust system. Let a watermark bit b_i be embedded N_i times and τ be the detection parameter that determines the acceptable majority level. Each extracted bit b_i^* from a non-watermarked relation has same probability of *1/2* to match or not match the original embedded bit in the watermark. Thus we define P_{bit} as the probability that at least τ (*majority threshold*) portion out of N_i can be detected from non-watermarked relation by sheer chance as[18]

$$P_{bit} = \sum_{j=\tau N_i}^{N_i} b\left(j; N_i, \frac{1}{2}\right) = B\left(\tau N_i; N_i, \frac{1}{2}\right) \tag{7}$$

For a watermark of length N_w, the false hit rate P_{FH} is given by:

$$P_{FM} = B\,(\tau_w; N_w, P_{bit}) \tag{8}$$

where τ_w is watermark length threshold on entire database, N_w is length of watermark. Hence, one can improve chances of preventing a false hit by selecting higher values of τ as well as τ_w, as the false hit probability reduces substantially.

4.2.2 False Miss

Authors defined false miss rate P_{FM} as the probability of not detecting a valid watermark from a watermarked relation. A robust system should have lower P_{FM}. Assume that Alice is the owner of the data set *R* which is attacked by Mallory an attacker. Mallory has no access to *R* and does not know any of the secret information used in embedding. We have imposed a three-level security strategy by using secret parameters K_s, N_p, and *v* that makes difficult for an attacker to destroy the watermark. We now classify the attacks performed by Mallory as subset deletion, subset alteration, subset insertion, invertibility, and additive attack as follows.

Subset Deletion Attack: In this circumvention technique, Mallory may delete randomly selected subset of tuples of watermarked database with an intention to remove the watermark. This will result in loss of tuples from different partitions. If some tuples are deleted from a partition, the arrangement of other tuples remains same. Hence watermark bit is correctly extracted. Even, if there are two tuples left per partition, we are able to extract watermark correctly. Thus, authors technique is highly resilient towards this attack.

Another possibility could be tuple deletion attack causes deletion of entire partition. However, each bit is embedded multiple times into different partitions. Thus, even if one partition is deleted; the embedded watermark bit can be recovered from other partitions.

Authors performed the experiment by deleting randomly selected tuples of the database. The results are plotted in Fig. 5. Experiment has revealed that on deletion of 96 per cent of tuples complete watermark is extracted. Even when 99 per cent tuples are deleted, 57.14 per cent of watermark is correctly extracted. However, in Sion[12], *et al.*, on deleting 20 per cent of tuples, the extracted watermark is perturbed, which increases almost linearly with the attack. At 80 per cent deletions, the watermark mismatch is 21 per cent. Thus authors technique is more robust and acts as a proof of ownership.

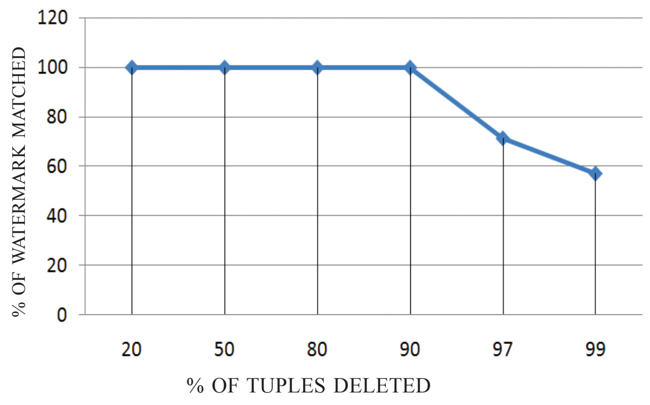


Figure 5. Subset deletion attack.

Table 3 shows the effect of N_p on the percentage of watermark matched on deletion of fraction of tuples from the watermarked dataset. At different values of N_p, watermark is extracted by deleting different percentage of tuples and compared with the original embedded watermark. The results indicate that when very small number of partitions is considered, all the watermark bits are not embedded. Hence watermark is not completely extracted. On the contrary, very large number of partition results in less number of tuples per partition. Thus the probability that entire partition get deleted/

Table 3. Effect of N_p on percentage of watermark matched against tuple deletion attack.

N_p	0 per cent	50 per cent	90 per cent	95 per cent	97 per cent	99 per cent
10	85.71	85.71	85.71	85.71	85.71	85.71
30	100	100	100	100	100	42.86
50	100	100	100	100	71.43	57.14
65	100	71.43	57.14	57.14	57.14	57.14
80	100	71.43	57.14	57.14	57.14	57.14

perturbed increases. Therefore, value of N_p is to be carefully selected to make system more secure.

Subset Addition Attack: In this attack, Mallory may add random tuples to the database with an intention to distort the embedded watermark. Added tuples fall to different partitions. These added tuples will act as additional noise. Till the number of tuples added per partition N_{ap} is less than half of the total tuples present in a partition after adding new ones N_{tp}, the correct watermarking bit is extracted. If $N_{ap}> ½ (N_{tp})$ then extracted bit may differ. However each newly added tuple is equally likely to get arranged in same or reverse order. Thus, every added tuple may not act as a noise. Practically even when $N_{ap}> ½ (N_{tp})$ is reached, watermark is not perturbed. In addition to this, each bit is embedded repeatedly into different partitions. Thus, even if one partition is destroyed, watermark bit can be correctly extracted from other.

Authors performed an experiment by adding random tuples to the database. On adding 400 per cent new tuples, the watermark extracted is same as watermark embedded. Thus the proposed technique is robust against subset addition attack. Initially the number of tuples in a partition is in particular order and every added tuple would either affect the order or it would maintain the order. Therefore, the head start makes it practically impossible to affect the order by addition attacks. Table 4 summarizes the effect of N_p on tuple addition attack. For smaller values of N_p, all watermark bits are not embedded, hence the watermark is not completely extracted. For higher values of N_p the resilience is 100 per cent.

Table 4. Effect of N_p on percentage of watermark matched against tuple addition attack.

N_p	0 per cent	50 per cent	100 per cent	200 per cent	300 per cent	400 per cent
10	85.71	85.71	85.71	85.71	85.71	85.71
50	100	100	100	100	100	100
80	100	100	100	100	100	100

Subset Alteration Attack: In this attack Mallory may alter certain values in the database to distort the database. In authors technique the hash values calculated depends on the derived primary key; unless there is a change in this primary key, the hash value will not be affected. An attacker can make small changes to data within the usability constraints, thus altering only the least significant bits of the attributes within a tuple but cannot tamper with *MSBs*. Hence, it will not affect the derived primary key and the same partitions will be created. Moreover, tuple reordering is based on the hash values generated and hence cannot be modified either. Thus the proposed technique is 100 per cent resilient towards subset alteration attack which is verified experimentally also. However in Sion[12], *et.al.*, perturbations in extracted watermark starts at 25 per cent data alteration which further increases linearly. At 80 per cent data alterations, 28 per cent perturbations exist. Hence, our technique is more robust against this attack.

Invertibility Attack: In this attack, Mallory claims ownership by finding a key which extracts the embedded watermark from pirated relation fortuitously. In proposed technique, the probability of success of this attack is lowered by setting proper size of secret parameter K_s and watermark length N_w. As probability of guessing the Key K_s is $1/(2^{|K_s|})$ and watermark is $1/(2^{|N_w|})$. By selecting larger values of K_s and length of watermark N_w, this probability can be made close to zero. Similarly, in order to guess *v*, Mallory has to correctly guess K_s, which is significantly made larger to avoid this attack.

Additive Attack: In this attack, Mallory embeds her own watermark to a watermarked relation and claims the ownership. In such case, we propose the same solution as discussed by Agrawal[2], *et.al.* Solution is to ask Alice and Mallory to produce the original relation into which watermark is embedded. Since, Mallory has embedded her watermark in Alice's watermarked database; she will present the pirated relation which contains Alice's watermark. Alice can extract her watermark from that relation and claim ownership while Mallory fails to do so.

Synchronization Error: In the proposed technique, every partition contains single watermark bit. Thus, on adding random tuples or on deleting tuples randomly, there is no effect on order of watermark bits retrieved. Bit index is securely calculated independent of tuple addition or deletion attack. Hence, there is no synchronization error in the proposed work.

Tuple Reordering Attack: We can save the order of all watermarked tuples and get it registered along with watermark. While decoding one can first arrange the tuples in the order saved and then extract the watermark. This extra storage space is acceptable at the cost of no distortion at all.

5. CONCLUSION AND FUTURE SCOPE

Authors proposed a robust watermarking technique for relational databases to resolve ownership issues. Major contributions of this study:

- The proposed technique watermarks categorical data without any perturbations to the dataset.
- Since technique is distortion less, it is suitable for any data type attribute such as numeric, non-numeric etc.
- The robust technique resolves ownership issues.
- Primary key independent hash partitioning technique is proposed.
- Three-level security strategy is implemented to make technique secure.
- Technique proposed is 400 per cent robust against tuple addition attack, 100 per cent against alteration attack and 96 per cent robust against deletion attacks.
- The technique is resilient to additive and inevitability attacks.

The area of digital watermarking is rife with challenges and ample research is still ongoing. Our future research will be directed towards increasing the level of resilience against several sources of attacks in the watermarking method. Additionally, multilevel authentication aspects will be added to enhance the security of the proposed technique.

With the rise in social networking sites, several web databases have emerged to resolve problems such as the scalability and agility. To name a few ORDBs, OODBs and NoSQL databases encompasses different database technologies to support a rise in the volume of data stored about users, objects and products, the frequency of data accessed, and performance and processing needs. Protection against ownership of such web databases is the current research challenge.

REFERENCES

1. Brown, Ian. E. The evolution of anti-circumvention law. *Int. Rev. Law Comput. Techno.*, 2006, **20**(3), 239-60. doi: 10.1080/13600860600852119
2. Agrawal, R.; Haas, P.J. & Kiernan, J. Watermarking relational data: framework, algorithms and analysis. *VLDB J.*, 2003, **12**(2), 157-69. doi: 10.1007/s00778-003-0097-x
3. Farfoura, M.E.; Horng, Shi-Jinn; Lai, Jui-Lin; Run, Ray-Shine; Chen, Rong-Jian & Khan, Muhammad Khurram. A blind reversible method for watermarking relational databases based on a time-stamping protocol. *Expert Sys. Appl.,* 2012, **39**(3), 3185–96. doi: 10.1016/j.eswa.2011.09.005
4. Shehab, M.; Bertino, E. & Ghafoor, A. Watermarking relational databases using optimization-based techniques. *IEEE Trans. Knowl. Data Eng.,* 2008, **20**(1), 116-29. doi: 10.1109/TKDE.2007.190668
5. Khanduja, V.; Verma, O. & Chakraverty, S. Watermarking relational databases using bacterial foraging algorithm, multimed. *Tools Appl., Springer,* 2013, **74**(3), 813-839. doi: 10.1007/s11042-013-1700-9
6. Khanduja, V. & Verma, O.P. Identification and proof of ownership by watermarking relational databases. *Int. J. Inf. Electro. Eng*., 2012, **2**(2), 274-77. doi: 10.7763/ijiee.2012.v2.97
7. Khanduja, V.; Chakraverty, S.; Verma, O.P. & Singh, N. A scheme for robust biometric watermarking in web databases for ownership proof with identification. *In* Proceedings of the International conference on Active Media Technology, 2014,LNCS 8610, Poland. doi: 10.1007/978-3-319-09912-5_18
8. Kamran, M.; Suhsail, S. & Farooq, M. A robust, distortion minimizing technique for watermarking relational databases using once-for-all usability constraints. *IEEE Trans. Knowl. Data Eng*., 2013, **25**(12), 2694-707. doi: 10.1109/TKDE.2012.227
9. Khanduja, V.; Chakraverty, S.; Verma, O.P.; Tandon, R. & Goel, S. A robust multiple watermarking technique for information recovery. *In* Proceedings of the IEEE International Advance Computing Conference, Delhi, 2014. doi: 10.1109/iadcc.2014.6779329
10. Khanduja, V.; Khandelwal, A.; Madharaia, A.; Saraf, D. & Kumar, T. A robust watermarking approach for non-numeric relational database. *In* Proceedings of the IEEE International Conference on Communication, Information & Computing Technology, Mumbai, 2012. doi: 10.1109/iccict.2012.6398095
11. Al-Haj, A. & Odeh, A. Robust and blind watermarking of relational database systems. *J. Comput. Sci*., 2008, **4**(12), 1024-29. doi: 10.3844/jcssp.2008.1024.1029
12. Sion, R.; Atallah, M.& Prabhakar, S. Rights protection of categorical data. *IEEE Trans. Knowl. Data Eng*., 2005, **17**(7), 912-26. doi: 10.1109/TKDE.2005.116
13. Li, Y.; Guo, H. & Jajodia, S. Tamper detection and localization for categorical data using fragile watermarks. *In* Proceeding of the 4th ACM workshop on Digital Rights Management, NewYork, 2004. doi: 10.1145/1029146.1029159
14. Guo, H.; Li,Y.; Lui, A. & Jajodia, S. Fragile watermarking scheme for detecting malicious modifications of database. *Information Sciences,* 2006, **176**, 1350–1378. doi: 10.1016/j.ins.2005.06.003
15. Khan, A. & Husain, S.A. A fragile zero watermarking scheme to detect and characterize malicious modifications in database relations. *Scientific World J.,* 2013, Article ID 796726. doi: 10.1155/2013/796726
16. Camara, L.; Li, J.; Li, R. & Xie, W. Distortion-free watermarking approach for relational database integrity checking. *Mathematical Problems in Engineering,* 2014, Article ID 697165. doi: 10.1155/2014/697165
17. Stallings, W. Cryptography and network security. 4thed., Pearson Education India, 2005. pp. 40-42.
18. Khanduja, V.; Chakraverty, S. & Verma, O.P. Robust watermarking for categorical data. *In* Proceedings of the IEEE International Conference on Control, Computing, Communication and Materials, Allahabad, 2013.
19. Schneier, Bruce. Applied cryptography, protocols, algorithms, and source code in C. 2nd ed., New York: John Wiley, 1996. pp. 455-59, 369-79.
20. National geochemical survey database of the US, http://mrdata.usgs.gov/geochem/select.php [Accessed on 5 February 2015].

ACKNOWLEDGEMENTS

The authors wish to thank Mr Divjot Singh for his collaborative efforts towards the work present in this paper. He is studying at Netaji Subhas Institute of Technology, Delhi, India under department of Computer Engineering.

CONTRIBUTORS

Vidhi Khanduja has proposed the idea and implemented the watermarking scheme. She has significantly contributed in writing the paper.

Shampa Chakraverty Under her esteemed guidance, the overall scheme is implemented. She has appreciably contributed in paper writing and the experiment section.

Mr O.P.Verma Under his esteemed guidance, the overall scheme is implemented. He has guided the comparison and theoretical analysis section.