A Scheme for Robust Biometric Watermarking in Web Databases for Ownership Proof with Identification

Conference Paper · August 2014

DOI: 10.1007/978-3-319-09912-5_18

Vidhi Khanduja
Netaji Subhas Institute of Technology
14 PUBLICATIONS **138** CITATIONS
SEE PROFILE

Om Prakash Verma
Delhi Technological University
87 PUBLICATIONS **1,630** CITATIONS
SEE PROFILE

A Robust Biometric Watermarking Technique of Web Databases for Ownership Proof with Identification

Vidhi Khanduja[1], Shampa Chakraverty[1], Om Prakash Verma[2] and Neha Singh[1]

[1] Department of Computer Engineering, Netaji Subhas Institute of Technology, Delhi, India
{vidhikhanduja9, apmahs.nsit}@gmail.com, nehalohchubh@yahoo.com
[2] Department of Information Technology, Delhi Technological University, Delhi, India
opverma.dce@gmail.com

Abstract. We propose a robust technique for watermarking relational databases using voice as a biometric identifier of ownership. The choice of voice as the distinguishing factor is influenced by its uniqueness, stability, universality and ease of use. The watermark is generated by creating a statistical model of the features extracted from the owner's voice. This biometric watermark is then securely embedded into selected positions of the fractional parts of selected numeric attributes using a reversible bit encoding technique. In case of a dispute regarding true ownership, the relative scores of the extracted watermark are generated by comparing features of the disputed voices with the extracted one. We experimentally demonstrate the robustness of the proposed technique against various attacks. Results show that watermark is extracted with 100% accuracy even when 97% tuples are added or when 90% tuples are deleted or when 80% tuples are altered. More significantly, biometric identification helped identify the correct owner even when 60% of the watermark suffered degradation.

Keywords: Relational Databases, Ownership Protection, Robust Watermarking, Voice Biometrics.

1 Introduction

Digital convergence technologies have enabled the seamless integration of human activities with powerful machine capabilities and thereby helped humanity achieve new milestones in almost every aspect of life. Ubiquitous intelligent interfaces perceive snippets of information continually from numerous environmental sources. The data collected are then transferred through the internet to build up huge compendiums of information. As a result, the web is today populated by large collections of digital databases. Unfortunately, the menace of piracy and rights infringement has also reached dangerous proportions by feeding upon these very technological advancements. Malicious or benign attacks pose continuous threats to the owners of digital and multimedia content. This has stimulated the need for powerful combative measures which can protect the ownership and integrity of databases efficiently.

Technologically Protective Measures (TPM) backed by anti-circumvention laws, have been adopted by many nations to deter such threats [1]. Digital watermarking is a cost-effective and widely used self help technological measure to protect multimedia content and, in recent years, digital databases [2]. In this paper, we focus on watermarking relational databases augmented by biometrics as a means to safeguard the interests of owners against illegal claims of ownership. It is well known that biometrics provide a promising technique to uniquely identify an individual based on his biological features and gives little scope for duplications. Contrary to traditional watermarking techniques which entail any random pattern as a watermark to be embedded into the database, we generate a meaningful sequence using the voice of the owner. The choice of voice is influenced by the fact that voice biometric is unique to an individual, is difficult to forge or fake, is universally available and is easy to capture through cheap audio sensors in a man-machine environment. The proposed technique is robust and reversible and can protect the ownership rights of the owners of web databases with a high degree of accuracy.

The rest of this paper is organized as follows. Section 2 gives a glimpse of the work done in the domain concerned. Section 3 describes the implementation of the proposed watermarking technique. Section 4 presents the experiments performed to check the robustness achieved in a biometrically watermarked database. Section 5 concludes our paper.

2 Related Work

One of the earliest works that aimed at protecting ownership rights of relational databases was proposed in [3]. The authors proposed a bit level watermarking technique that effectively marks numeric attributes by modifying the least significant bit (*lsb*) of selected numeric attributes. Later, in a bid to enhance security, Xinchun *et. al.* proposed a modification to the simple bit level watermarking technique by introducing the concept of weighted attributes wherein all candidate attributes for marking are given different weights by the owner in order to model the marking rate of each attribute as per her preference [4]. However, these *lsb* based techniques are not resilient to simple bit based attacks such as random bit flipping which can degrade the watermark.

Some watermarking techniques have been proposed that optimize the statistical properties of watermarked attributes in order to maximize discernability while respecting usability constraints [5, 6, 7]. The technique proposed by M.Shehab *et.al.* is primary key dependent and computationally inefficient [5]. In [6] the authors proposed a primary-key independent, optimization based watermarking technique using Bacterial Foraging Algorithm with increased robustness and efficiency. Recently, Kamran and Farooq proposed an information-preserving watermarking technique on numeric attributes, highlighting the need of rights protection in Electronic Medical Records (EMR) systems [7]. Researchers have also focused on non-numeric and categorical attributes for rights protection [8, 9, 10].

Scant attention has been paid towards investigating the potential of biometrics to prove the ownership of digital databases. In [11], Haiqing Wang proposed a technique for embedding the owner's voice to establish ownership rights. There are some serious shortcomings in their approach. Firstly, it is not reversible and hence any changes made to embed the watermark are permanent and can't be undone. Secondly, the authors have not outlined any specific approach to reach a decision about the ownership of the data in question, which is the primary objective of the whole process. In this paper, we too adopt voice as a biometric source of watermark to establish ownership with identification. However, in sharp contrast with [11] we adopt a reversible technique which not only reverses all modifications made to the database during the process of watermark insertion but also enables a higher degree of robustness. Furthermore, we demonstrate systematically how ownership conflicts can be resolved by identifying the contending voices against the extracted watermark.

3 Proposed Technique

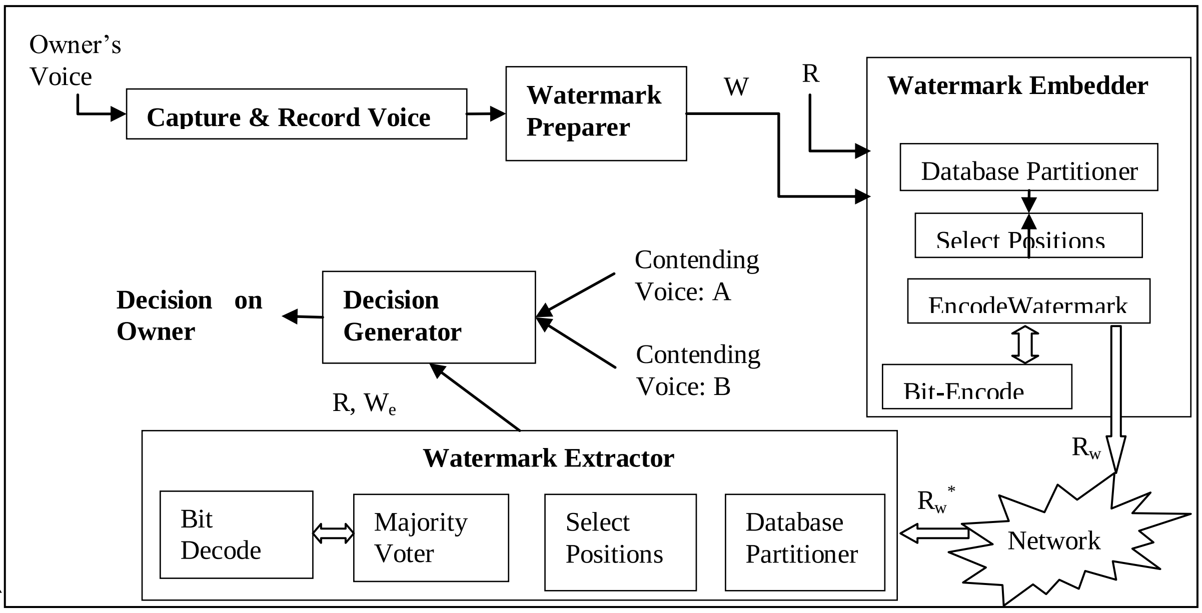


Fig. 1. Architecture of biometric watermarking system

Fig. 1 gives the overall architecture of the proposed voice biometric based watermarking technique for a relational database. It consists of the sub-modules: *Watermark Prepararer*, *Watermark Embedder* and *Watermark Extractor* and *Decision Generator*. We now describe each of these modules in the following sub-sections.

3.1 Watermark Preparer

The foremost step is to generate the watermark bit sequence using the database owner's biometric characteristic. The process *Watermark Preparer* does this by using the owner's voice as the source to construct the watermark.

The process starts by capturing and digitally recording the owner's voice sample using audio sensors, analog to digital converters and voice recording software. The digital representation of voice data is used as a training sample to extract its features using Mel Frequency Cepstral Coefficients (MFCC). This involves a series of processing steps which ensure that the final compressed signal adequately represents what the human auditory system can truly discern. These include, performing pre-emphasis of the voice data, framing and windowing, generating the Fast Fourier Transform (FFT) coefficients of the signal that represent its complex frequency spectrum, generating the power spectrum from the complex spectrum, applying Mel filtering to the power spectrum using Mel filterbanks, taking logarithms of the Mel filterbank energies, performing Discrete Cosine Transform (DCT) on the log Mel filterbank energies, selecting appropriate DCT coefficients and appending delta energy features [12].

After extracting the MFCC feature vectors χ, a statistical model of the data is generated using Gaussian Mixture Model (GMM). The goal is to estimate the parameters of the GMM that best matches the distribution of training feature vectors χ [13]. This is achieved by employing the maximum log likelihood estimation technique which returns the mean matrix, the nodal diagonal covariance matrix and the mixture weights of the GMM. The individual elements of the matrices are then rounded off and concatenated to yield a bit sequence *W[1: L]* of length L to be used as the watermark.

3.2 Watermark Embedder

Process *Watermark Embedder* embeds the watermark sequence *W[1:L]* in the relational database R. Its attributes includes the primary key attribute *Pk* and η number of float numeric attributes which are candidates for the embedding watermark bits. The watermark is embedded into specific bits of the fractional portions of selected numeric attributes in target tuples.

The database owner supplies a set of secret parameters. A key K is used for securely embedding the watermark. The parameter Υ determines the fraction of all tuples in a database that can be utilized for embedding watermark bits. Parameter Ω defines the total number of potential locations within the fractional part of float numeric attributes where a watermark bit can be inserted. The following processes are carried by the *Watermark Embedder* using these secret parameters.

(i) *Create virtual partitions:* Firstly, subroutine *DB_Patition(.)* shown in Fig. 2, carries out the task of partitioning R into N_p virtual partitions $S_0, \ldots, S_{Np-1}$ which are initially empty. Next, taking each tuple turn by turn, the process concatenates the secret key K at both ends of the unique primary key *t.Pk* of a tuple t, generates its hash by applying the MD5 hash function [14] and finally returns its partition index $i(t)$ which is equal to $H(t) \bmod N_p$. The virtual partition S_i thus gets its new member t.
The purpose of partitioning is to boost the resilience of the watermark against random tuple reordering attacks. As the assignment of each tuple to its virtual partition is a function of its primary key value rather than its spatial position in the database, reordering does not change its membership. The partitioning process uses the secret key K known only to the owner. Further it employs a randomizing hash function, thus making it almost impossible for an attacker to guess the partition assigned to any tuple.

DB_Partition(.)

```
1. Initialize sets $S_0,S_1....S_{Np-1}$ to $\phi$
2. **For each** tuple $t \varepsilon R$ {
3.     Calculate $H(t) = hash (K || t.Pk|| K)$.
4.     Calculate index $i(t)= (H(t) \bmod N_p)$
5.     Assign tuple t to the $i^{th}$ partition, $S_i=S_i \cup t$
6. }
```

Fig. 2. Algorithm for generating virtual partitions of the database

Select_EmbedPositions(.)

```
1. **For each** partition $S_i$ {
2.    Calculate watermark bit index, $z_i= i \bmod L$.
3.    Calculate $A_{i1} = (i \bmod \eta)$ and $A_{i2} = (i \bmod \eta-1)$
4.    **For each** tuple $t \varepsilon S_i$ {
5.        Calculate $Ħ(t) = hash (K || t.Pk)$
6.        **If** $((Ħ(t) \bmod \Upsilon) = = 0)$ {
7.            Calculate $b(t) = ((Ħ(t) \bmod \Omega)+1)$
8.    }}}
```

Fig. 3. Algorithm for selecting positions for embedding watermark bits

(ii) *Select secret embedding positions:* The next step is to select potential locations for embedding the L bits of the watermark $W[1:L]$. For each partition S_i the process *Select_EmbedPositions(.)* shown in Fig. 3, calculates the following position parameters:

(a) A specific watermark bit index z_i within the L-bit watermark W which will be embedded in partition S_i (line 2). Actually, z_i as well as the next bit position z_{i+1} are embedded in partition S_i.

(b) Two among η candidate attributes A_{i1} and A_{i2}, into which the selected bit will be embedded (line 3). Readers may note that more than two attributes per tuple can be selected for embedding the watermark bits in order to increase the level of redundancy. However, we have selected only two watermarking attributes as an illustration for introducing redundancy.

(c) Specific tuples which are earmarked for embedding (line 5-6). To decide which tuples in a partition will participate in the embedding process, the secret key K is concatenated with the tuple's primary key and its hash $Ħ(t)$ is generated. If the value of $Ħ(t) \bmod \Upsilon$ is *zero*, then it is selected for watermarking (line 6). In effect, $1/\Upsilon$ fraction of the total number of tuples are thus earmarked. This secure hashing mechanism serves to ward off attackers who may venture to guess the watermarked tuples.

(d) A specific bit position $b(t)$ within the fractional part of selected attributes which will contain the watermark bit z_i and z_{i+1} (line 7). The hash $Ħ(t)$ is also used to choose $b(t)$.

(iii) *Encode watermark bit:* For each selected tuple t in a virtual partition S_i two consecutive watermark bits $W[z_i]$ and $W[z_i+1]$ are embedded at the chosen bit position $b(t)$ of selected attributes A_{i1} and A_{i2} respectively. Fig. 4 shows the reversible bit encoding sub-routine *Bit-Encode(.)* that inserts a watermark bit w into the b^{th} bit of a numeric attribute A of tuple t. The process *Watermark_Encoder(.)* in Fig. 1, calls *Bit_Encode(.)* twice for each selected tuple t: $Bit\_Encode(Ħ(t),A_{i1},b(t),W[z_i])$ and then $Bit\_Encode(Ħ(t),A_{i1},b(t),W[z_i])$.

The reversible encoding process is based on the technique proposed in [15]. The fractional part is first separated from its integer part, subtracted from the tuple's hash and converted to its binary form *Bin* (lines 1-3). Next, its value *Bin(b)* at b position and the chosen watermark bit w are re-arranged so that b now contains w while *Bin(b)* is appended at the end (lines 4-6). The modified fraction is converted back to decimal, subtracted again from the tuple's hash and added to the integer part I (lines 7-9).

The above secure encoding process serves two functions. Firstly, the secure hash function introduces a random perturbation in each attribute' value, making it impossible for an attacker to detect any pattern that can correlate the original and new values of the watermarked attributes. Secondly, the attribute can revert back to its original value after extracting the watermark bit. Fig. 6 illustrates this reversibility with the help of an example. Fig. 6a (Encode) illustrates the watermark bit encoding procedure by tracing through the steps of *Bit_Encode(.)*.

The watermarked database R_w is now prepared and can be transferred through a network where it may be exposed to different kinds of attacks.

Bit_Encode(Ħ, A, b, w)

```
1. Segregate integral part $I$ and fractional part $F$ of
   attribute A
2. Calculate $P_e = Ħ - F$
3. Convert $P_e$ to binary, $Bin = binary(P_e)$.
4. $Temp = Bin[b]$
5. $Bin[b] = w$
6. Concatenate $Temp$ at the end of $Bin$
7. Convert $Bin$ to decimal, $P_e' = decimal(Bin)$
8. Calculate $F' = Ħ - P_e'$
9. Calculate updated attribute value $A_j' = I + F'$
```

Fig. 4. Algorithm for embedding watermark bits

Majority_Voter(.)

```
1.  Assign $Count[1:L][0] = Count[1:L][1] = 0$
2.  For each partition $S_i^*$ {
3.      For each tuple $t \in S_i^*${
4.          If $((Ħ(t) \bmod \Upsilon) == 0)$ {
5.              $B_1^* = Bit\_Decode(Ħ(t), A_{i1}, b(t))$
6.              $B_2^* = Bit\_Decode(Ħ(t), A_{i2}, b(t))$
7.              Assign $Count[z_i][B_1^*] += 1$ and
                $Count[z_{i+1}][B_2^*] += 1$
8.  }}}
9.  For g =1 to $L${
10.     If ($Count[g][1] > Count[g][0]$)
11.     $W^*[g] = 1$
12.     Else
13.     $W^*[g] = 0$ }
14. $Generate\_Decision(W^*[1:L])$
```

Fig. 5. Algorithm for extracting the watermark

(a) Encode	**(b) Decode**
Let, A_i = 1.9800, *Ħ(t)*= 177, *W[i]* = 0 and position *b(t)*=3. 1. I=1, F=98 2. P_e = *Ħ(t)- F= 177-98=79* 3. *Bin= binary(*P_e*)= 1001111* 4. *Temp= Bin[b] => Temp=1* 5. *Bin[b]=W[i]=0 =>Bin[b]=0* 6. Concatenate *Temp , Bin=10010111* **7.** P_e'*= decimal(Bin)=151* 8. *F′= Ħ(t) -* P_e'*=26* **9.** A_j'*=I+F′=1.2600*	*Let*, A_i = 1.2600, *Ħ(t)*= 177, *W[i]* = 0 and position *b(t)*=3. 1. *I′=1, F′=26* 2. P_e' *= Ħ - F′=151* 3. *Bin= binary (*P_e'*)*=10010111 4. *Temp*= lsb(B_{in}) =1 and delete from *Bin* => *Bin=1001011* 5. *W[i]* as b^{th} lsb of *Bin=> W[i]=0* 6. *Bin[b] = Temp=>Bin=1001111* 7. P_e *= decimal(Bin)=79* 8. *F = Ħ(t) -* P_e*= 98* 9. A_j*=I+F=1.98*

Fig. 6. Illustration of reversible watermark bit encoding

3.3 Watermark Extractor

The block *Watermark Extractor* in Fig. 1 contains all the processes needed for extracting the embedded watermark sequence from a suspected database R_w* using a majority voting technique. The process also restores back the altered values of the database to its original form by exploiting the reversible nature of the embedding technique.

The process of watermark extraction starts by repeating the first two steps of watermark embedding. It calls *DB_Paritions(.)* to generate the virtual partitions and *Select_EmbedPositions(.)* to select the watermarked tuples, attributes and bit positions. Next, it calls the process *Majority_Voter (.)* described by the pseudo-code in Fig. 5, to reconstruct the watermark. *Majority_Voter(.)* takes each virtual partition in turn and initializes all elements of the *Count[1:L][0:1]* matrix to *zero*. An element *Count[x][0]* counts the number of times *W[x]* is extracted as zero and *Coun[x][1]* counts the number of times a *W[x]* bit is extracted as one. The sub-routine *Bit_Decode(.)* given in Fig. 7, retrieves an embedded watermark bit value from a watermarked tuple *t* from its possibly compromised attribute *A′* from bit position *b*. *Majority_Voter(.)* iteratively calls *Bit_Decode(.)* for each watermarked tuple and uses the retrieved values corresponding to the *z* and $(z+1)^{th}$ bit of the watermark to update the corresponding elements of the *Count[.][.]*. Finally, the full watermark sequence is reconstructed as W_e using majority voting on the pair *Count[x][0]* and *Count[x][0]* for each watermark bit $x \in 1..L$.

Fig. 6b (Decode) illustrates how the modification done in an attribute's value due to the encoding carried out by the example in Fig. 6a is reversed by tracing through the steps of *Bit_Decode(.).*

Bit_ Decode(Ħ, A′, b)	***Decision_Generator(.)***
1. Separate A' into fractional part *F′* & integer part *I′* 2. Calculate P_e' = *Ħ(t) - F′* 3. Convert P_e' to binary; *Bin= binary (*P_e'*).* 4. Store *lsb* of B_{in} in *Temp* & delete from *Bin* 5. Extract *w* as *Bin[b]* 6. Assign *Bin[b] = Temp.* 7. Convert B_{in} to decimal; P_e = *decimal(Bin)* 8. Calculate *F = Ħ(t) -* P_e 9. Calculate original attribute value *A=I+F*	1. Decode the extracted watermark W_e into mean, variance and weight matrices as parameters of the trained GMM λ. 2. Input the voice samples *A* and *B* of contenders. 3. Extract MFCC feature vectors for *A* and *B*. These are the observation sequences χ_A and χ_B. 4. Calculate Identification scores using log-likelihood of χ_A and χ_B with respect to model λ. 5. The speaker who has obtained maximum score is declared as the owner of the database.
Fig. 7. Algorithm of Decode subroutine	**Fig.** 8. Algorithm of score-based technique

3.4 Decision Generator

The *Majority_Voter(.)* hands over the extracted bit sequence W_e to the process *Decision_Generator(.)*, described in Fig. 8. *Decision_Generator(.)* first converts back W_e into the mean matrix, the nodal diagonal covariance matrix and the mixture weight matrix as parameters of the prior trained GMM. Let us denote this model as λ. Next, *Decision_Generator(.)* inputs voice samples *A* and *B* of two contenders for ownership, presumably one of whom is the genuine owner, and extracts their respective *MFCC* feature vectors. These serve as test observation sequences χ_A and χ_B. For testing, the multigaussian log-likelihood is computed for each observation feature vector using the trained model λ as the reference to yield the respective identification scores for *A* and *B* [13]. These comparative scores aid in conclusively identifying the owner of the database; the speaker with the maximum score is declared the owner.

4 Experimental Results and Discussions

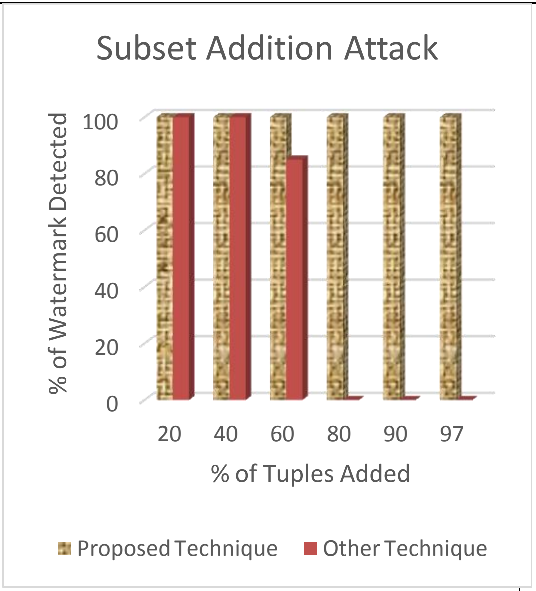
We used a 2.2 GHz Intel i5 core processor with 4 GB RAM Windows 7 OS and the MATLAB computing environment for our experiments. We applied the proposed voice-enabled watermarking algorithm on a national geochemical survey database of the US [16]. This database comprises a complete geochemical coverage of the United States and has been used for a wide variety of studies in geological and environmental sciences. For our experiments, we used a database of 76071 records. The number of virtual partitions was fixed to be 732.

4.1 Robustness Analysis

An attacker Mallory may try to execute malicious attacks or benign updates on a watermarked database with the intention of degrading its contents or destroying the embedded watermark. The resilience of any watermark scheme against such attacks is a measure of its usability and efficiency. We now illustrate the robustness of our proposed technique against various subset attacks and draw a comparison with [11].

(i) Subset addition attack: In this attack, Mallory inserts records from other sources into the watermarked relation. The unmarked tuples fall under different partitions randomly and contribute to noise. However the extraction process utilizes the repetitive embedding of each bit of the watermark in numerous tuples and attributes by conducting a majority voting to reconstruct the original watermark.
We simulated the subset addition attack by adding different quantities of random tuples to the database and tested what percentage of the watermark could still be recovered as a result. The results are plotted in Fig. 9 alongside the results of subset addition attack in [11]. On adding a subset of size 97% of the original database, the full watermark could be fully recovered in our case, whereas in [11] with the watermark could not be recovered at all under the same situation, dropping sharply to 85% on addition of 60% extra tuples and then degrading beyond recognition. This clearly indicates the superior robustness of our technique and justifies the selection of multiple attributes for embedding against the technique in [11], where the watermark is embedded into a single attribute.

(ii) Subset Alteration Attack: In this attack Mallory alters certain values in the database in order to distort the watermark. But in doing so, Mallory is faced with the dilemma of compromising the usability of the database. Therefore, Mallory can only resort to randomly altering certain values or altering the least significant bits of certain attributes. But no matter what she chooses, our technique is highly resilient against this class of attacks by enforcing secure mechanisms of partitioning, selecting tuples and attributes and deciding locations for embedding.
We simulated the subset alteration attack by altering some fraction of the all tuples in the database randomly. We plotted the results and compared the results reported in the work of [11], as shown in Fig. 10. Till the point where 80% of the tuples were altered, we could correctly extract the watermark and on 97% tuple alteration, 60% of the watermark could be extracted. In contrast, the results for subset alteration attacks as reported in [11], show that their technique failed to extract the watermark on 90% alteration. This clearly establishes the superior performance of our reversible and multi-attribute insertion technique.



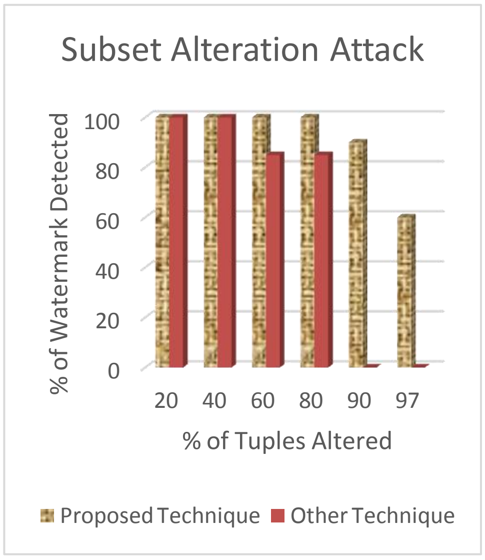
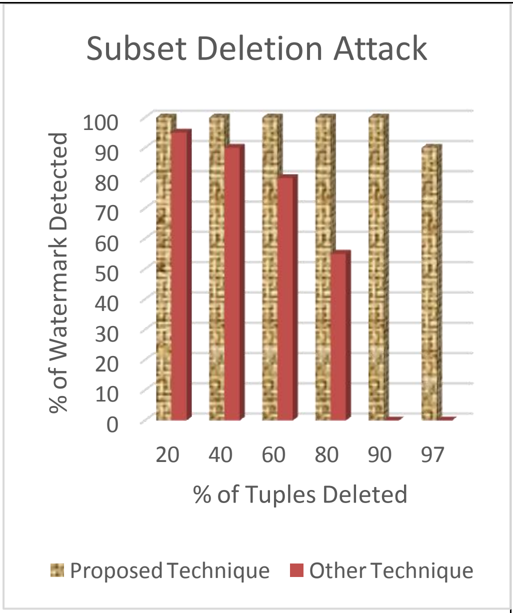


Fig. 6. Subset Addition Attack	**Fig. 7.** Subset Alteration Attack

(iii) Subset Deletion Attack: Now Mallory tries to delete randomly selected subsets of tuples in the watermarked database so as to obliterate the watermark. There is a dilemma for her again as excessive deletions runs the risk of rendering the database useless. Mallory may delete tuples randomly which will result in the loss of some watermarked tuples from all partitions. The lost bit can be recovered from other positions and decoded using majority voting. Thus, after deletion even if one marked tuple is found from a partition, it helps in extracting the watermark sequence.
We simulated the subset deletion attack by deleting randomly selected tuples of the database. The watermark extracted was recorded as shown in Fig. 11 below. Experimental result reveals that even when 90% of tuples were deleted, 100% of watermark could be extracted correctly. In contrast, the results for subset deletion attack as reported in [11] show that their technique failed to extract the watermark on 90% deletion and in fact, steadily degrades from 20% tuple deletion onwards.



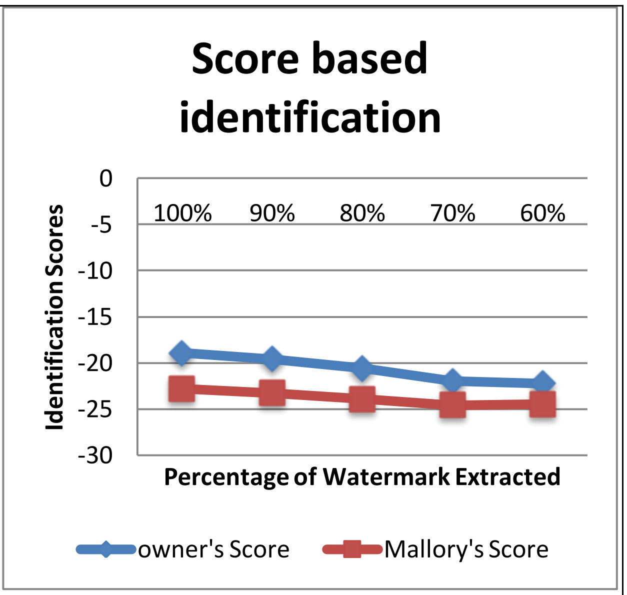


Fig. 8. Subset Deletion Attack	**Fig. 9.** Graph showing scores of Owner & Mallory by altering the watermark

4.2 Score based Identification

Earlier experiments revealed that due to the redundancy introduced in embedding the watermark coupled with majority voting, we were able to extract it fully against all subset attacks till 80% modifications were introduced. Beyond that level of noise, the watermark does suffer degradation. We now show that the use of biometrics gives the added advantage of correctly identifying the owner despite the degraded watermark.

Fig. 12 shows the variation of relative scores of the owner's and Mallory's identification when compared with altered voice extracted from the database. We can infer that even when only 60% of the original watermark is detected, the relative score of owner's voice against the features recovered from the extracted watermark is always higher than that of Mallory.

In [11], the authors have completely ignored the issue of monitoring the degree to which the voice can be detected when the extracted watermark suffers degradation, thereby suggesting that a perfect match is needed. On the other hand, we have adopted a comparative score based detection of voice identification. Even when there is an alteration in the extracted voice sequence, the owner's voice scores higher than the attacking contender's. Thus, there is always a sense of clarity regarding the decision which makes the final conclusion more trustworthy. Further, it is clear from Fig. 12 that the watermark degrades even for a mere deletion of 20% tuples in case of [11] and eventually gets completely destroyed. The detection mechanism would obviously fail under such high levels of watermark degradation.

5 Conclusion

At the core of our proposed watermarking scheme lies the principle of using a biometric feature for watermarking a digital database. Specifically, we have used the owner's voice as a stable, readily available and reliable mark of identification to prepare the watermark. This increases the accuracy of the decision regarding ownership of the database.

The encoding technique is reversible. This ensures that changes made during the embedding process are fully reversed. Hence, the quality of information is not compromised for the sake of ensuring ownership protection. The fact that our technique is completely reversible allows us to introduce a greater degree of redundancy in embedding the watermark, thereby increasing robustness significantly. This is borne out by our experimental results. We were able to extract the watermark with 100% accuracy despite 97% subset addition or 90% subset deletion or 80% tuple alteration attacks.

Instead of applying absolute voice recognition to identify the owner in case of conflicts, we use comparative scores of the voice samples of the contenders. This relative evaluation approach allows us to correctly identify the owner for a large range of degradation in the quality of the watermark that is extracted from a compromised database. We are currently investigating the potential of multiple biometrics for improving the reliability of ownership proof in noisy environments.

References

1. Prateek Chakraverty, Vidhi Khanduja, "Digital Database Protection-A Technico legal perspective", 7th National conference on computing for nation development, BVICAM, Delhi, pp-361-364, 2013.
2. Dr. Ian R. Kerr, Alana Maurushat and Christian S. Tacit, "Technical Protection Measures: Tilting At Copyright's Windmill", Ottawa Law Review, vol. 34, No. 1, pp-6-82, 2002-2003.
3. R.Agrawal & J.Kiernan, "Watermarking relational databases", In proc. of the 28th Very Large DataBases conference (VLDB), vol. 28, pp.155–166, Aug. 2002.
4. CUI Xinchun, QIN Xiaolin, SHENG Gang, "A Weighted Algorithm for Watermarking Relational Databases", Wuhan University Journal of Natural Sciences, vol. 12 No.1, pp-79-82, 2007
5. M. Shehab, E. Bertino, and A. Ghafoor, "Watermarking Relational Databases Using Optimization-Based Techniques," IEEE Trans. Knowledge and Data Eng. (TKDE), vol. 20, no. 1, pp. 116-129, Jan. 2008.
6. Vidhi Khanduja, Om Prakash Verma, Shampa Chakraverty, "Watermarking Relational databases using Bacterial Foraging Algorithm", Multimedia Tools and Applications, Springer, 2013, Doi: 10.1007/s11042-013-1700-9.
7. M.Kamran and Muddassor farooq , "An information Preserving Watermarking Scheme for right protection of EMR systems", IEEE Transactions on knowledge and data engineering, Vol. 24, No.11, pp. 1950-1962, November 2012.
8. Vidhi Khanduja, Anik Khandelwal, Ankur Madharaia, DipakSaraf, Tushar Kumar, "A Robust Watermarking Approach for Non-Numeric Relational Database", IEEE International Conference on Communication, Information & Computing Technology (ICCICT), pp. 1 – 5, 2012.
9. R.Sion, M.Atallah and S.Prabhakar," Rights protection of categorical data", IEEE TKDE, vol. 17, no. 7, pp. 912-926, 2005.

10. Vidhi Khanduja, Shampa Chakraverty, Om Prakash Verma, “Robust Watermarking for Categorical data”, IEEE International conference on Control, Computing, communication and materials ICCCCM, Allahabad, pp.174- 178, 2013.
11. Haiqing Wang, Xinchun Cui and Zaihui Cao, “A Speech based Algorithm for Watermarking Relational Databases”, International symposium on Information Processing, pp.603-606, 2008.
12. http://www.cic.unb.br/~lamar/te073/Aulas/mfcc.pdf
13. D Reynolds, Richard C.Rose, “Speaker identification using gaussian mixture speaker models”, IEEE transactions on speech and audio processing”, vol.3, No. 1, pp.72-83, Jan 1995.
14. Schneier, B. Applied cryptography Protocolss, algorithms and source in C, 2nd ed., Wiley:India, pp. 429-445, 1996.
15. Mahmoud E. Farfoura, Shi-Jinn Horng, Jui-Lin Lai, Ray-Shine Run, Rong-Jian Chen, Muhammad Khurram Khan, “A blind reversible method for watermarking relational databases based on a time-stamping protocol”, Expert Systems with Applications 39, pp. 3185–3196, 2012.
16. National geochemical Database, http://mrd ata.usgs.gov/geochem.