An Adaptive Robust Watermarking Algorithm for Audio Signals Using SVD

Malay Kishore Dutta[1], Vinay K. Pathak[2], and Phalguni Gupta[3]

[1] Department of Electronics and Communication Engineering,
Galgotias College of Engineering and Technology, Greater NOIDA, India
[2] Department of Computer Science and Engineering, HBTI – Kanpur, India
[3] Department of Computer Science and Engineering, IIT -Kanpur, India
malay_kishore@rediffmail.com, vinaypathak.hbti@gmail.com,
pg@cse.iitk.ac.in

Abstract. This paper proposes an efficient watermarking algorithm which embeds watermark data adaptively in the audio signal. The algorithm embeds the watermark in the host audio signal in such a way that the degree of embedding (DOE) is adaptive in nature and is chosen in a justified manner according to the localized content of the audio. The watermark embedding regions are selectively chosen in the high energy regions of the audio signal which make the embedding process robust to synchronization attacks. Synchronization codes are added along with the watermark in the wavelet domain and hence the embedded data can be subjected to self synchronization and the synchronization code can be used as a check to combat false alarm that results from data modification due to watermark embedding. The watermark is embedded by quantization of the singular value decompositions in the wavelet domain which makes the process perceptually transparent. The experimental results suggest that the proposed algorithm maintains a good perceptual quality of the audio signal and maintains good robustness against signal processing attacks. Comparative analysis indicates that the proposed algorithm of adaptive DOE has superior performance in comparison to existing uniform DOE.

Keywords: Watermarking, Digital right management, Singular value decomposition, Robustness, Audio Signals.

1 Introduction

Illegal reproduction and unauthorized distribution of digital audio has become a high alarming problem in protecting the copyright of digital media [1]. Digital watermarking is one of the possible solutions for copyright protection and digital right management. A watermark is designed for residing permanently in the original audio data even after repeated reproduction and distribution. Since human auditory system (HAS) is more sensitive than human visual system (HVS) embedding watermark to the audio signal is more difficult than embedding in an image. According to IFPI (International Federation of the phonographic Industry) [2], a good audio watermarking algorithm should meet requirements of imperceptibility, robustness and security.

M.L. Gavrilova et al. (Eds.): Trans. on Comput. Sci. X, LNCS 6340, pp. 131–153, 2010.

Imperceptibility means that embedded watermark should be imperceptible to human auditory system (HAS). The watermark should be robust so that it can survive intentional and unintentional signal processing attacks. The watermarking algorithm should be secure which means that the watermark can only be detected by the authorized person. These requirements are often contradictory with each other and there is a need to make a trade-off among them. There are some well known watermarking algorithms in time domain [3], [4] and in frequency domain [5], [6]. Some algorithms are proposed using quantization methods and cepstrum domain [7], [8]. It has been seen in the methods that synchronization attacks cause a severe problem in detection and recovery of watermark. In such an attack the watermark is actually present in the audio signal but cannot be detected because the synchronization is lost. Synchronization attacks such as cropping and TSM (time-scale modification) cause dislocation between embedding and detection in the time domain and hence although the watermark is present in the audio signal it is difficult to recover it. Some methods proposed to solve the problem of synchronization attacks are exhaustive search [9], peak point extraction with special shaping [10], content based localized watermarking [11], high energy reference points based watermarking [12] and self synchronization for audio watermarking [13]. In [21] a sound synthesized process in digital instruments is proposed as a real-time watermarking method. Both musical performance and the insertion of watermark can be actualized in real time. A method to enhance the security of vocal communication over an open network is proposed in [22]. The method divides speech data using the secret sharing scheme and transfers the shared data using the multipath routing technique to realize secure voice communication over the network. To solve the problems associated with de-synchronization attacks, an audio watermarking scheme is proposed in [23] based on support-vector-machine (SVM) theory by using audio statistics characteristics and a synchronization code technique. In [24] a Multiplicative Patchwork Method (MPM) for audio watermarking is presented. The watermark signal is embedded by selecting two subsets of the host signal features and modifying one subset multiplicatively regarding the watermark data, whereas another subset is left unchanged. In [25], the issue of audio source separation from a single channel is addressed, i.e., the estimation of several source signals from a single observation of their mixture. The presented results open up new perspectives in both underdetermined source separation and audio watermarking domains

The synchronization codes are used to locate the positions where the watermark is embedded in the audio. In time domain the embedding strength is limited to maintain perceptual transparency and hence not robust to signal processing attacks. If synchronization codes are embedded in frequency domain the robustness increases to a great extent, but in doing so the computational cost for searching the codes also increases. High energy points used as reference for watermark embedding regions are also used in watermarking [12]. These peak points after special shaping [10] serve as reference points for embedding and detection of the watermark. The performance of these peak point extraction methods is found to be moderate and there is a requirement for enhancing the performance of such methods.

This paper embeds a synchronization code in the audio signal with reference to the high energy peaks. In doing so the accuracy in detecting the watermark increases in comparison to normal high energy reference point method. This synchronization code is useful to combat false alarm which is generated by the modification of audio data on watermark embedding. Since the synchronization code is embedded only in selected high energy regions, the computation load in searching such codes decreases to a great extent. Hence in the proposed method the synchronization is maintained with higher accuracy and at lower computational cost in comparison to other existing methods.

Singular value decomposition (SVD) based watermarking methods [14], [15] have been proposed for image watermarking, but not enough research has been reported for SVD based audio watermarking. In this paper an adaptive SVD based audio watermarking method is proposed which is localized according to the content of the audio signal. The basis of the SVD based image watermarking method is that the singular values of the image remains unaltered even if some alterations are made in the image. Accordingly the inverse of this property where the singular values are modified without changing the perceptual property of the signal is used in watermarking the signals.

The paper is organized as follows; Section 2 gives an overview of singular value decomposition (SVD). The watermark generation from an image and enhancing its security are discussed in Section 3. The method for finding watermark embedding region and determining the degree of embedding for each region are described in Section 4. The synchronization code generation and its implementation are explained in Section 5. The Section 6 comprehensively describes the watermark embedding and detection in the SVD domain in the audio signals. Various parameters used to measure the performance of the proposed method have been discussed in Section 7. The experimental results are given in Section 8 and the last section concludes the paper.

2 Singular Value Decomposition

Singular value decomposition (SVD) is used to diagonalize matrices. It packs most of the signal energy into a few singular values. The SVD belongs to the group of orthogonal transformations, which decompose the input matrix into several matrices and one of which has only nonzero values in the main diagonal. SVD has been a successful method for image watermarking and in this paper it is proposed to use the SVD based method for watermarking of audio signals. An arbitrary matrix A of size $M \times N$ can be represented by its SVD as:

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathbf{T}}$$

where **U** and **V** are $M \times M$ and $N \times N$ matrices respectively. The columns of **U** and **V** are mutually orthogonal unit vectors. The $M \times N$ matrix **S** is a pseudo-diagonal matrix and its diagonal elements, which are arranged by descending gradation, are all non-negative values. They are called SVs and the first value is far larger than others. While both **U** and **V** are not unique, the singular values are fully determined by **A.**

To apply the SVD in an audio signal each audio frames (coefficients in time domain or any other domain like DWT, DCT, FFT domain etc.) is converted into two dimensional matrix. Once the SVD operation is done, the matrix **S** which has diagonal elements in the descending order can be modified or quantized as per the watermark bit to be embedded. To explain the method let us consider the original audio frame as A and W is the watermark bits to be embedded.

$$A = USV^T$$

$$S_W = \text{Modified / Quantized value of S}$$
$$A_W = US_wV^T$$

where S_W is the modified singular values and A_W is the watermarked audio frame whose SVs are modified.

3 Watermark Generation

An image is used as the watermark. To ensure the security and to improve the robustness of the proposed method, the watermark should be pre-processed before embedded into the host signal. Due to the periodicity of the Arnold transform, the image can be recovered easily after permutation. So, the Arnold transform is applied to the original binary image watermark [16]. To use Arnold transform we make $M=N$. If the size of the image is $N \times N$, $(x, y)^T$ is the coordinate of the watermark image's pixel, $(x', y')^T$ is the coordinate after the transform. Arnold transform can be expressed as:

$$\begin{bmatrix} \overline{x} \\ \overline{y} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} (\text{mod N})\text{ï}$$

The steps in converting the image into a watermark for audio signals are given below:

1. Compute a global threshold that can be used to convert an intensity image to a binary image. The threshold has to be normalized intensity value that lies in the range [0, 1].
2. Using the above threshold convert the image into a BW image.
3. Resize the image into $M \times N$ as per the design requirement of the Watermarking model.
4. The resized image is scrambled by applying the Arnold transform.
5. Convert the scrambled image I into a vector W as:

```
for i =1:M
      for j = 1:N
            W (k) = I (i , j)
            k = k + 1;
       end
end
```

6. Rescale the vector W as $\alpha \times W$ where α is the strength of the watermark.

The choice of α depends on the design requirements of the watermarking method. Proper values of α can optimize imperceptibility and robustness of the watermarking method. Lower values of α makes the watermark imperceptible while higher values of α makes the watermark robust against signal processing attacks.

4 Selection of Watermark Embedding Regions

Finding the watermark embedding region is one of the most challenging steps in audio watermarking. If the embedding regions are not properly selected, the detection and recovery of the watermark under signal processing attacks can be very difficult. This paper proposes a method of selecting such regions on the original audio waveform which is based on selecting high energy peaks as reference points.

Prominent instruments like drum, tabla (an Indian instrument), piano etc. form a sequence of high energy peaks. These peaks are so dominant that the other sounds are normally masked at that instant. Also considering the pre-masking and post masking of these peaks, the region around these peaks can be modified without affecting the quality of the audio for human auditory system. These peaks are normally 0.1 to 0.2 second in length, so under the sampling rate of 44100 KHz they spread over 4410 to 8820 samples. These sharp transients are less prone to synchronization attack. Thus these regions are ideal for watermark embedding. It is shown in [11] that after TSM (time scale modification) attack although the absolute time-domain positions of those local regions with high energy have some change after time scaling, these shapes do not change much. Fig. 1 shows an original signal and time scaled signal of a sample audio. It can be seen from the figure that although the positions of the high energy peak has changed but the surrounding localized region does not change much. Thus, by embedding the watermark in these areas, it is reasonable to believe that the watermark be safe under TSM attacks to some extent.

This proposed method chooses the high-energy peaks and these peaks act as reference points for region of watermark embedding. For selecting high-energy peaks a threshold is chosen above which all such peaks are considered as reference points. This threshold is taken as a fraction of the maximum value of the sample in the time domain signal. The number of regions for watermark embedding (ROE) depends on the selected threshold. So the threshold for deciding the number of regions for embedding of watermark data has to be properly chosen as per the characteristics and size of the watermark data.

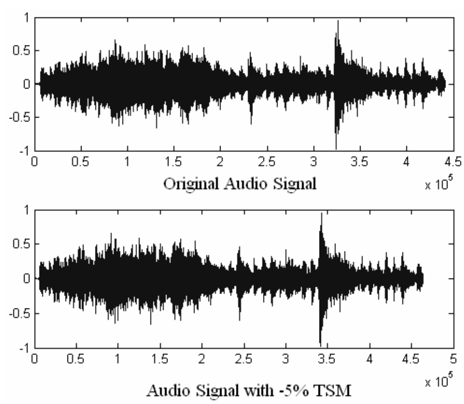

Fig. 1. Waveform of Original and -5 % TSM Audio Signal sample

It has been shown in [17] that in synchronization attacks like time scale modifications (TSM) are performed on the harmonic components and the residual components separately. The harmonic portion is changed in time scale by modulating each harmonic component to DC, interpolating and decimating the DC signal and then demodulating each component back to its original frequency. The residual portion, which can be further separated into high energy transients and noise in the wavelet transform domain. In doing so the edges and the relative distances between the edges are preserved and the noise component is time scaled. In contemporary music, there is use of percussion instruments like drum, tabla etc. The beats of these instruments are the high energy music edges which can be used as reference points. Also since these high energy edges maintain the rhythm of the music, cropping degrades the quality of the music to an annoying extent. Hence we can conclude that the time scale modification method changes the audio signal in the low energy regions (minimum transients) and tends to conserve the high energy transients. Hence it is clearly apparent that if the watermark is embedded in audio segments near the high energy peaks, the effect of synchronization attacks is minimum. On the other hand, if the watermark is embedded in the minimum transient regions synchronization attacks severely effect the detection and recovery of the watermark from the audio signal.

Based on the above discussion the watermark embedding regions are chosen in high energy transients. The amount of watermark information to be embedded in these regions is kept adaptive. It is intended to embed more information in the more sharp music edges and less information in less sharp music edges which makes the embedding process adaptive in nature. The reason for this adaptive embedding process is that more sharp edges are more resistant to synchronization attacks and hence more watermark information can be embedded in such regions. The number of watermark bits to be embedded in a high energy region is decided by some local characteristics of that particular region.

Algorithm 1

C is vector containing reference points and F is a zero vector equal to the size of C. n is the degree of embedding.

$K = zeros\ [length(C), n]$
$F[i] = C[i]/max(C)$
F[i] mapped to degree of embedding
for i = 1: length(C)
  $K\ [F\ (i) \times n]\ [i] = F[i]$
for i = 1: length(c) (there is one non-zero entry in every column)
  if K (i, j) > 0 then
j is the number of watermark bits to be embedded for the ROEWM(i) corresponding to i^{th} *element of vector C.*

The number of watermark bits to be embedded in the *ROE* is called as degree of embedding *(DOE)* and depend on the sharpness of the music edge at that region. More watermark data is embedded in more sharp regions whereas less watermark data is embedded in less sharp regions. So the degree of embedding of watermark is made localized according to the sharpness of the music edge which acts as the reference point for the embedding region. In doing so the watermark can embedded in a justified manner and the watermark becomes robust to synchronization attacks. The point *i* (high energy peak) is excluded from *ROE* so that it is not modified in the process of watermark embedding and creates a possibility of not detecting it in the watermark detection process. This point *i* is used as a reference and not for embedding of watermark. A generic method of selecting *DOE* for the reference points is given in Algorithm 1. For example if ten levels of adaptive *DOE* are used then using Algorithm 1 *DOE* can be selected from a set of pre-decided ranges as shown in Table 1.

Table 1. Degree of Embedding for the Set of Reference Points using Algorithm 1 for Ten Level DOE

	Range of Reference point (i)	**Degree of Embedding (j)**
1	*[max(X(i)] > i ≥ 0.95* [max(X(i)]*	*10*
2	*0.95* [max(X(i)] > i ≥ 0.90* [max(X(i)]*	*9*
3	*0.90* [max(X(i)] > i ≥ 0.85 *[max(X(i)]*	*8*
4	*0.85* [max(X(i)] > i ≥ 0.80 *[max(X(i)]*	*7*
5	*0.80* [max(X(i)] > i ≥ 0.75 *[max(X(i)]*	*6*
6	*0.75* [max(X(i)] > i ≥ 0.70 *[max(X(i)]*	*5*
7	*0.70* [max(X(i)] > i ≥ 0.65 *[max(X(i)]*	*4*
8	*0.65* [max(X(i)] > i ≥ 0.60 *[max(X(i)]*	*3*
9	*0.60* [max(X(i)] > i ≥ 0.55 *[max(X(i)]*	*2*
10	*0.55* [max(X(i)] > i ≥ 0.50 *[max(X(i)]*	*1*

Once the reference points are determined and the *DOE* is selected the region of watermarking *ROE* has to be determined for the reference point. Further the *ROE* has to be divided for embedding the watermark data and a synchronization code (*syncode*) as *ROEWM* and *ROESYNC*. The synchronization code is used as a tool against false alarm due to data modification as a result of watermark embedding. The details of this synchronization code are explained in the next Section. The block diagram of the watermark embedding method is shown in Fig. 2.

Steps in selection of reference points and determination of ROE for watermark embedding are given in Algorithm 2.

Algorithm 2

1. *Let X be the Audio Signal.*
2. *Find the max value of the samples* $X(i)_{max}$ *in X.*
3. *Find all the peaks above* $[(1-n\alpha)\,(X(i)_{max}]$ *where α is a fraction. Store these values in a vector D.*
4. *A new vector C is created as*

   for n =1: |D|

   if D(i+1) – D(i) > |A|+|WM|

   then C(i) = D(i);

   where A = length (Audio signal) /length (watermark +p X *syncode)*

   where p is the length of the block used for quantization to embed one bit of the syncode.

Algorithm 2 (Contd.)

5. *Select DOE for each ROE corresponding to each element of Vector C using Algorithm 1.*
6. *z=0*
   for i = 1:length(C)
   if K(i,j)>0
   z = z + j;
   C is the required set of reference points
   else
   n=n+1
   go to step 3

The region of watermark and synchronization code embedding (ROE) is given as:

$$ROE_i = [C(i) - |A|/2 - p \times length(syncode): C(i) + |A|/2]$$

(Excluding the point i)

The embedding region ROE_i is sub divided into two parts as region of embedding sync code ROESYNC and region of embedding watermark ROEWM as:

$$ROESYNC(i) = ROE_i(1:length(p \times syncode))$$
$$ROEWM(i) = ROE_i(length\ p \times (syncode) + 1: length(R_i))$$

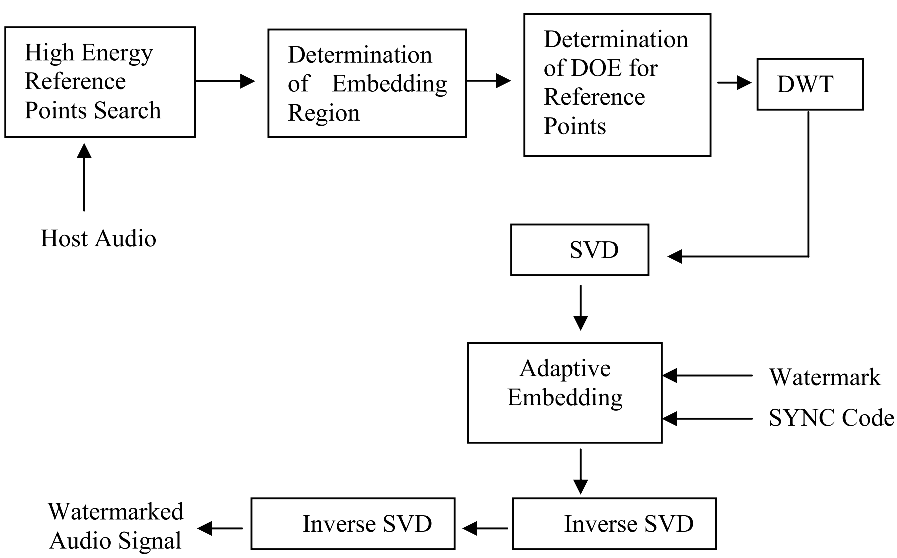

Fig. 2. Watermark Embedding

5 SYNC Code Generation and Embedding

Under the de-synchronization attacks like cropping and time scale modifications the watermarked audio is prone to suffer dislocations of the watermark embedded regions. The synchronization codes embedded into the original audio serves as a method to locate correctly the watermarked embedded regions after the signal has suffered de-synchronization attacks.

The embedding of watermark data in the audio signal causes slight modifications in the audio signal. Although the changes are made in a strategic way so that they are imperceptible to human auditory system (HAS) but there may be other problems that may arise as a result of these modifications. In this proposed method the reference points and the degree of embedding are based on the magnitude of the high energy points considered above a certain threshold. It may happen that as a result of watermark embedding, a certain point which was below the threshold goes above it and a false alarm is generated. The reverse of this problem i.e. a point above threshold going below is not applicable in this method as the reference point is excluded from the *ROE*. To counter this kind of false alarm the synchronization code is embedded in the *ROE* which serves as a check for finding an authentic *ROE* in the detection and recovery process. It can be noted that this self-synchronization check for authentic detection of *ROE* involves less computational load in comparison to existing conventional synchronization check methods. In this proposed method synchronization code is searched only in specific regions and not in all parts of the audio file. Hence the computation load is reduced to a great extent which is a good feature of the proposed method of watermarking. Use of such localized synchronization codes eliminates false alarm generated due to data modification on watermark embedding.

In the proposed method PN sequences based on chaotic maps have been used as a synchronization code. If $\{a_i\}$ is an original synchronization code and $\{b_i\}$ is an unknown sequence both having the same length. If the number of different bits between $\{a_i\}$ and $\{b_i\}$, when compared bit-by-bit, is less than or equal to a predefined threshold τ, the $\{b_i\}$ is determined as the synchronization code.

In order to generate the synchronization code chaotic maps are used. Chaotic systems are deterministic systems that are governed by non-linear dynamics. These systems show deterministic behavior which is very sensitive to initial conditions, in a way that the results are uncorrelated and seem to be random in nature. To increase the security of the sync code, a random chaotic sequence generated by the hybrid chaotic dynamical system is used [18]. The synchronization code (sync code) is generated by thresholding the chaotic map. The initial point of the chaotic sequence generator is a secret key. By using generators of a strongly chaotic nature we can ensure that the system is cryptographically secure, i.e., the sequence generation mechanism cannot be inversely engineered even if an attacker can manage to obtain a part of the sequence. For example a chaotic dynamic is given below:

$$\text{Chaotic}(x) = \begin{cases} 1 - 2x^2, & -1 \leq x < -0.5 \\ 1 - \frac{1}{2}(-2x)^{1.2}, & -0.5 \leq x < 0 \\ 1 - 2x, & 0 \leq x < 0.5 \\ -(2x-1)^{0.7}, & 0.5 < x \leq 1 \end{cases} \tag{1}$$

The above chaotic map (1) can produce almost uncountable random sequences that are extremely sensitive to the initial secret key. Steps in generating the chaos based watermark are as follows:

Step 1: Generate a chaotic sequence S with an initial key (within the given limit) using any of the chaotic equation given in (1) of length M. (M depends on the design choice)

Step 2: Use zero as threshold for the chaotic sequence S. All elements greater than zero are made equal to 1 otherwise –1.

Step 3: Repeat the Step 1 and Step 2 for different values of initial keys to generate different sequences that can be used in different audio samples.

After the Synchronization points *C(i)* and the embedding segments *R(i)* are determined, the sync code is embedded in the audio signal. The embedding segment should be large enough to have room for the sync code and some watermark bits. In the proposed method we choose quantization index modulation (QIM) for embedding the sync code because of its good robust nature. Also by using QIM method the search for sync is blind in nature, which means the original audio is not needed in sync code extraction.

The quantization parameter is made adaptive in nature by using the mean of each *ROESYNC* segment in the QIM process. The quantization parameter is made localized depending on the nature of the signal. The steps used for embedding the sync code in the audio signal is given in Algorithm 3.

Algorithm 3

1. Divide the coefficients of 3rd level DWT using Haar filter of *ROESYNC(i)* into p sub-segments where p is the length of the syncode.

2. The mean value of the coefficients of p^{th} sub-segment of i^{th} reference point is calculated as: $\overline{ROESYNC\ (i)\ (p)} = \sum ROESYNC\ (i)\ (p)\ /\ length(p^{th}\ sub\text{-}segment)$ where $\sum$ is the summation of all the DWT coefficients in the p^{th} sub-segment of the i^{th} *ROEWM.*

3. Embed each bit of the synchronization code in each sub-segment p as

   if syncode(p) = 1

   $ROESYNC\ '\ (i)\ (p) = ROESYNC\ (i)(p) + 2 * \overline{ROESYNC\ (i)(p)}$

   else

   $ROESYNC\ '\ (i)(p) = ROESYNC\ (i)(p) - 2 * \overline{ROESYNC\ (i)(p)}$

   where *ROESYNC(i)(p)* and *ROESYNC'(i)(p)* are the original and modified p^{th} sub-segment of the i^{th} *ROESYNC* respectively.

4. Take IDWT of the *ROESYNC'(i)(p)* to convert back to the time domain

6 Watermark Embedding and Detection

Once the watermark embedding regions and the degree of embedding for every region are decided, the watermarking is done in a content based adaptive manner in the discrete wavelet domain (DWT). The choice of DWT for watermarking has several advantages such as it needs lower computation load in comparison to DCT and DFT and it has variable decomposition level.

6.1 Watermark Embedding

The high energy reference points are determined as discussed in Section 4. The indices of these points is stored in the vector *C* and the region of embedding for the ith point is determined as discussed in Section 4 and is given by:

$$ROE\ (i) = [C\ (i) - |A|\ /2 - p \times length\ (syncode) : C\ (i) + |A|\ /2 - 1] \quad (2)$$

DWT is performed on this *ROE* segment. The syncode is embedded successively into the low frequency sub-band of segments. The length of this *ROE* segment depends on the amount of data that is to be embedded. It should be large enough to accommodate the synchronization code and some watermark bits. The number of watermark bits to be added in a *ROE* segment is decided by the degree of embedding of that segment as was discussed in Algorithm 1 which makes the watermark embedding adaptive in nature according to the content of the audio.

As discussed in Algorithm 1 for the matrix *K,* there is one non-zero element in every row corresponding to one high energy reference point. This non-zero element will indicates the degree of embedding. For mathematical simplicity this is restricted to a limited number of quantization levels and the value of this non-zero entry indicated the number of watermark bits to be embedded in that *ROE*. However a more complicated relation can be customized as per contents and nature of the host audio.

If *K* is and *i X j* matrix, where *i* indicate the number of reference points and *j* indicates the degree of embedding. We need to search all the non-zero elements (every row has one) in *K* and then decide the degree of embedding. The watermarking technique used here is quantization index modulation (QIM) [19], [20] because of its robustness to signal processing attacks and since it is blind in nature (original signal not required for watermark extraction). The watermark embedding steps are given in Algorithm 4.

Algorithm 4

Step 1: Determine the region of embedding *ROE(i)* using Equation 4.2.

Step 2: Obtain region of embedding the watermark *ROEWM* as discussed in Section 4.4 is given as:
ROEWM(i) = ROE(i) (length (syncode) +1: length (R(i))

Step 3: Apply third level DWT to the audio segment using Haar wavelet.

Algorithm 4 (Contd.)

Step 4: Find the non-zero element in each row of matrix *K*. The index of this non zero element is the degree of embedding *(DOE)*. So for all the *i* rows of the matrix *K*, there is one *DOE(i)*.

Step 5: Divide the low frequency approximate wavelet components of each *ROEWM(i)* segment into *j* equal sub-segments where *j* is the degree of embedding *DOE(i)* for that corresponding *ROEWM(i)*

Step 6: Convert the sub-segments into blocks *ROEWM(i)(j)* of size *m X m* (blocks are converted into matrix to apply SVD, Zero padding may be done to achieve *m X m* size).

Step 7: Calculate SVD for each *ROEWM(i)(j)* as $ROEWM(i)(j) = USV^T$
Let $S_{ij} = (S_{11}\ S_{22} \ldots\ldots\ldots. S_{mm})$ be the non zero diagonal elements of the matrix S for the j^{th} sub- segment of the i^{th} reference point.

Step 8: Embed the watermark using QIM. The embedding is done as follows:

for i = 1: length(wm)

if K[i][j] >0

$$S_{ij}' = \begin{cases} \lfloor S_{ij}/\mu \rfloor \cdot \mu + 3\mu/4 & \text{if } wm(i) = 1 \\ \lfloor S_{ij}/\mu \rfloor \cdot \mu + \mu/4 & \text{if } wm(i) = 0 \end{cases}$$

i =i +1;

end for

where $\lfloor\ \rfloor$ indicates the floor function and S_{ij} and S_{ij}' are the SVD of DWT coefficients of the low frequency sub-segment of the original and watermarked audio data respectively. By increasing the value of μ one increases the robustness but decreases imperceptibility. This value of μ has to be maximized in such a way that the watermark maintains perceptual transparency.

Step 9: Obtain the watermarked sub-segment $\overline{ROEWM(i)(j)}$ by applying inverse SVD to the modified singular values.

Step10: Convert the modified audio segment from all the modified sub-segments. The inverse DWT is performed to get the watermarked signal.

6.2 Watermark Extraction

The watermark extraction is the reverse process of the watermark embedding process. The first step in this process is to identify the embedding regions *(ROE)*. Once the embedding regions are identified then the watermark detection and recovery can be performed. The synchronization points are to be determined first and then *ROE* is to be estimated. As discussed in the Section 5 there is a synchronization code that is embedded in the *ROE*. This synchronization code can now be a check in determining the *ROE* for all the high energy reference points. The steps for detection and recovery of the watermark are given in Algorithm 5.

Algorithm 5

Step 1: Determine vector *C* containing the index (*i* elements) of all the high energy reference points as discussed in Section 4.4 using the same value of threshold that is used in the embedding process.

Step 2: Determine the *ROESYNC(i)* for all reference points *i*, as discussed in Section 4. Divide the *ROESYNC(i)* into *p* sub-segments as discussed in the embedding process, where *p* is the length of the SYNCODE.

Step 3: Calculate the mean value of each sub-segment *ROESYNC(i)(j)*. If the mean value is greater than or equal to zero, a bit "1" is detected otherwise bit "0" is detected.

Step 4: Use the normalized correlation (NC) to find the similarity between the extracted syncode and the original syncode as follows:

$$NC\,(sync, sync^*) = \frac{\sum_{i=1}^{M} sync(i)\, sync^*(i)}{\sqrt{\sum_{i=1}^{M} sync(i)^2}\;\sqrt{\sum_{i=1}^{M} sync^*(i)^2}}$$

where *sync* and *sync** are the original and extracted syncode, respectively and *M* is the length of the sub-segment. If the NC between sync and sync* is greater than or equal to a pre-defined threshold σ then *sync** is accepted as a synchronization code and then go to Step 7. Otherwise go to Step 5 and Step 6 in an alternate repetitive way, i.e. go to step 5 and in next turn go to step 6 and repeat this till it goes to Step 7.

Step 5: Shift the *ROESYNC* by 1 sample to the left, i.e. *ROESYNCL(i) =ROESYNC (i- 1)* and repeat Step2 to Step 4.

Step 6: Shift the *ROESYNC* by 1 sample to the right, i.e *ROESYNCR(i) = ROESYNC(i+1)*and repeat Step 2 to Step 4.

Step 7: Once a NC above the pre-determined value is determined it indicates the identification of a *ROESYNC(i)*. The updated value of *i* is used to determine the *ROE(i)*. From this updated value of *i*, calculate the degree of embedding *j*.

Step 8: Calculate the *ROEWM(i)* from the *ROE(i)* calculated in step 7. Take DWT of this segment. This DWT segment *ROEWM(i)* is divided into *j* equal sub- segments. The sub-segments are converted into blocks *ROEWM(i)(j)* of size $m \times m$ (blocks are

Algorithm 5 (Contd.)

converted into matrix to apply SVD, Zero padding may be done to achieve *m X m* size).

Step 9: Calculate SVD for each *ROEWM(i)(j)* as

$$ROEWM(i)(j) = USV^T$$

Let $S_{ij} = (S_{11}\ S_{22} \ldots\ldots\ldots S_{mm})$ be the non zero diagonal elements of the matrix S for the j^{th} sub-segment of the i^{th} reference point.

Step 10: Extract the watermark *wm'* as follows:

$$wm'(k) = 1 \quad if\ S_{ij}^* - \lfloor S_{ij}^* / S \rfloor \cdot S \geq S/2$$
$$wm'(k) = 0 \quad if\ S_{ij}^* - \lfloor S_{ij}^* / S \rfloor \cdot S < S/2$$

k = 1: length(wm). The extracted watermark is de-scrambled using inverse Arnold transform to obtain the original watermark image.

It is clear from the above that for extracting the watermark we need to search bit by bit the synchronization code for every reference points. After the synchronization codes are found the embedding regions of the watermark *ROEWM* is determined. In case no synchronization codes are found the search window is shifted and searched again alternately in both directions till the code is found. It can be seen that the original SVD coefficients are not required in the extraction process and thus the algorithm is blind in nature.

7 Performance Evaluation Parameters

The performance of the proposed method of audio watermarking is evaluated by some performance coefficients as discussed below:

a. Signal to noise ratio (SNR)

The SNR is the objective quality measure to evaluate the perceptual transparency of the watermarked signal. It can be defined as:

$$SNR = 10\log_{10}\frac{\sum_{i=1}^{L} X^2(i)}{\sum_{i=1}^{L}\left[X^2(i) - \bar{X}(i)^2\right]}\ dB$$

where X and X' are the original and watermarked audio signals and M is the length of the audio signal.

b. Normalized Correlation (NC)

It is used to evaluate the similarity measurement of extracted binary watermark, which can be defined as:

$$NC\left(W,\bar{W}\right) = \frac{\sum_{i=1}^{M} W(i)\ \bar{W}(i)}{\sqrt{\sum_{i=1}^{M} W(i)^2}\ \sqrt{\sum_{i=1}^{M} \bar{W}(i)^2}}$$

where W and W^* are original and extracted watermark respectively, i is the index of the watermark and M is the length of the watermark.

c. Bit Error Rate (BER)

The bit error rate (BER) is used to find the percentage of error bits between original watermark and extracted watermark. The BER is given by:

$$BER = \frac{1}{M} \sum_{i=1}^{1 \le i \le M} W(i) \oplus \bar{W}(i)$$

where W and $\bar{W}$ are the original and the extracted watermark respectively, $\oplus$ is exclusive OR (XOR) operator and M is the length of the watermark.

c. Subjective Listening Test

To evaluate the audio quality, subjective listening tests can be performed to find the mean opinion score (MOS). These MOS is one of the most widely used subjective methods for watermarked audio signal quality evaluation. Ten listeners of different age groups are provided with the original and the watermarked audio signal and they are asked to classify the difference in terms the MOS grades. The MOS grades are defined in Table 2.

Table 2. MOS Grades

Effect of Watermark	Quality of Audio	Score
Imperceptible	Excellent	0
Perceptible but not annoying	Good	-1
Slightly annoying	Fair	-2
Annoying	Poor	-3
Very Annoying	Very Poor	-4

8 Experimental Results

Experiments have been performed with different types of audio files. As discussed in Section 4 the degree of embedding is dependent on the localized audio characteristics which make a justified method of embedding data in the audio. In case of non-adaptive data embedding more number of ROE may be required which can be achieved by decreasing the threshold (which is a fraction of the maximum value of the audio signal, as shown in Table 1). In doing so, data will be embedded in less sharp music edges which is more prone to synchronization attacks. In addition to that for embedding watermark data in the less sharp music edges, the threshold for determining ROE has to be decreased. This in turn may cause false alarm due to data modification on watermark embedding.

It is found that on watermark embedding the audio signal undergoes modification and the SNR for the watermarked signal is shown in Table 3. Although there is a modification in the audio signal, but it remains imperceptible to human auditory system because the masking effects. The length of the SYNC code used in the experiments is 64 and the size of the watermark image is 32*32 =1024. Since the audio samples are sampled at 44100 KHz and the high energy peaks are 0.1 sec length, these peaks spreads over 4410 samples. (Some peaks like that of drum last for 0.2 sec and hence spreads over 8820 samples). So it is clearly seen that these high energy peaks and the pre and post-masking associated with them provides enough room for embedding watermark in these regions that remain imperceptible to human auditory system.

Table 3. The SNR of the Watermarked Audio Signal and the Mean Opinion Score (MOS) from Subjective Listening Test

No.	Watermarked Audio Sample	SNR (dB)	MOS	No.	Watermarked Audio Sample	SNR (dB)	MOS
1	Classical 1	36.1	0	14	Piano	31.1	0
2	Classical2	33.2	0	15	Flute	26.2	0
3	Classical 3	34.2	0	16	Guitar1	32.1	-1
4	Country 1	36.1	0	17	Guitar 2	33.4	0
5	Country 2	39.0	0	18	Vocal1	27.1	-1
6	Country 3	32.1	0	19	Vocal2	23.1	-2
7	Pop1	33.0	0	20	Vocal3	28.3	-1
8	Pop2	32.9	0	21	Sitar 1	28.9	0
9	Pop 3	32.7	-1	22	Sitar 2	23.1	0
10	Blues	31.3	0	23	Violin 1	28.1	0
11	Folk 1	36.1	0	24	Violin2	26.1	0
12	Folk 2	29.5	0	25	Tabla	38.1	0
13	Folk 3	23.3	0				

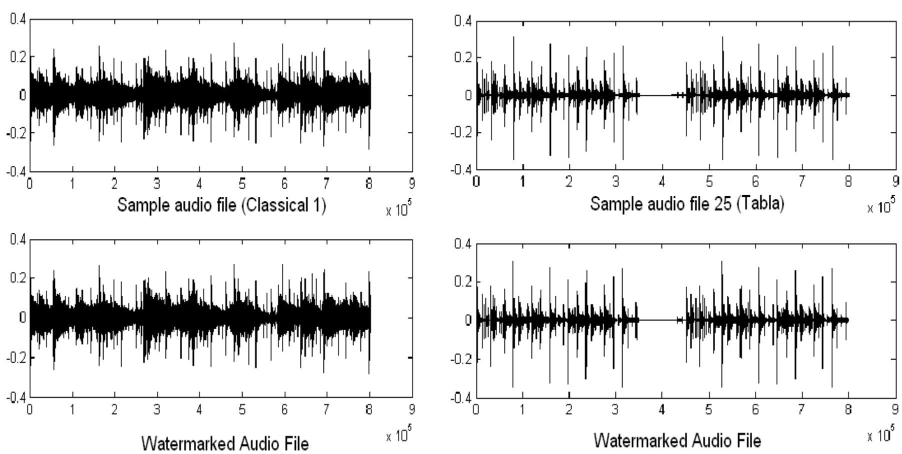

Fig. 3. Original and Watermarked Audio Signals

The scores MOS of the subjective listening tests are presented in Table 3. It can be en from Table 3 that the watermark embedded in the audio signal is imperceptible to human auditory system and also the SNR computed for watermarked signals is above 20dB which is considered good according to IFPI (International Federation of the phonographic Industry)

In order to test the robustness of the proposed method, different types of signal processing attacks are performed on the watermarked audio signal as described below:

a) Filtering: Low pass filtering with a cut off frequency of 8 KHz. The filter used is a second order Butterworth filter.
b) Resampling: The watermarked signal originally was sampled at 44.1 KHz, re sampled at 22 KHz and restored by sampling again at 44.1KHz.
c) AWGN: White Gaussian noise is added to the signal so that the resulting signal has a SNR more than 20 dB.
d) Time scale modification: TSM processing is done in the watermarked audio signal to change the time scale to an extent of +10% and -10%.
e) MP3 Compression: The MPEG -1 layer 3 compression with 32 kbps is applied.
f) Cropping: Segments of 500 samples are removed randomly from the watermarked signal.

In the watermark recovery process the first step is the detection of correct regions of embedding. To evaluate the performance of the method a parameter called ratio of correctly detected regions (ROCDR) is defined. It is the ratio of the number of correct embedded regions detected to the total embedding regions detected. It gives an indication of the false alarm that is generated in the detection process.

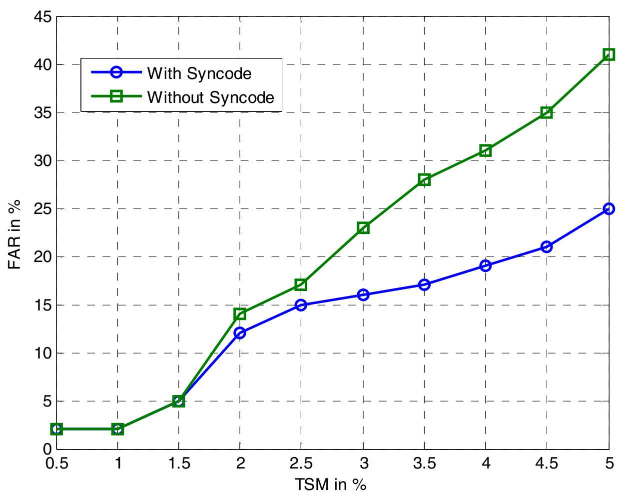

Fig. 4. (a) Positive FAR under different values of TSM foe audio sample Classical 1 with and without Syncode

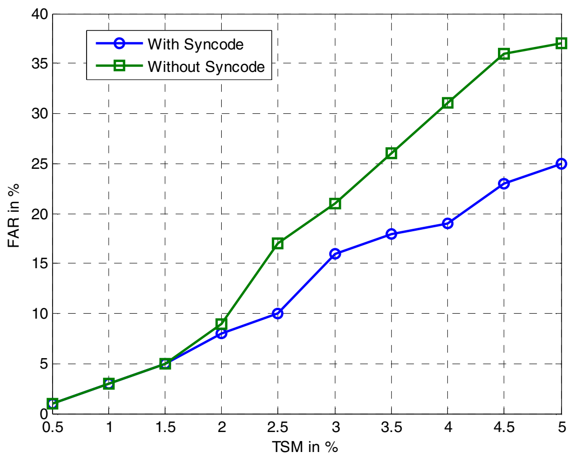

Fig. 4. (b) Positive FAR under different values of TSM foe audio sample Tabla with and without Syncode

The ROCDR for the proposed method for various samples under different signal processing attacks are shown in Table 4. To make a comparison with existing methods comparative results are also shown in Table 4 which reveals that the proposed method is better in comparison to the other two methods. The integration of the synchronization code makes the proposed scheme perform better since the positive false alarm is countered by it. Negative FAR condition does not apply since the peak reference point has not been used for data embedding. Fig. 4(a) and Fig. 4(b) shows the positive false alarm rate with and without the sync code under different degrees of TSM. It can be seen that the positive FAR is reduced specially under high degree of time scale modification attacks. A comparison of robustness tests against signal processing attacks for the proposed adaptive method (variable DOE) with non-adaptive method (uniform DOE) is given in Table 5. It can be seen clearly that performance of the proposed adaptive method is better than the non-adaptive uniform method. An image used as a watermark as discussed in Section 2 is shown in Fig. 5.

Fig. 5. Binary image watermark

The recovered watermark image under different signal processing attacks is presented in Fig 6. For comparison purpose, along with the performance of the proposed method the performance of the non-adaptive method is also presented. It can be seen the watermark recovered under serious synchronization attacks is of decent quality. The proposed method has good performance against low pass filtering, addition of white Gaussian noise, MP3 compression, cropping, re-sampling and time scale modification up to ± 5%.

Table 4. RCDR of Audio Samples under Audio Signal Processing

Attack	Audio Sample	ROCDR		
		Scheme [10]	Scheme [12]	Proposed Scheme
Cropping	Classical 1	91%	88%	91%
	Piano	92%	86%	94%
	Multiple Instruments	91%	88%	94%
	Pop 1	94%	91%	95%
	Vocal1	92%	91%	94%
	Tabla	88%	85%	89%
Re sampling	Classical 1	83%	80%	91.1%
	Piano	94%	88%	97.2%
	Multiple Instruments	75%	70%	93.1%
	Pop 1	73%	71%	89%
	Vocal1	91%	82%	97%
	Tabla	93%	91%	99%
AWGN	Classical 1	77%	72%	87%
	Piano	73%	55%	81%
	Multiple Instruments	88%	79%	86%
	Pop 1	69%	71%	79%
	Vocal1	71%	67%	86%
	Tabla	92%	87%	97%
Low pass filtering	Classical 1	64%	56%	79%
	Piano	85%	67%	88%
	Multiple Instruments	70%	66%	85%
	Pop 1	76%	71%	87%
	Vocal1	85%	70%	92%
	Tabla	84%	80%	88%
MP3 compression	Classical 1	70%	45%	71%
	Piano	56%	53%	77%
	Multiple Instruments	68%	56%	71%
	Pop 1	78%	69%	81%
	Vocal1	61%	56%	69%
	Tabla	79%	71%	89%
TSM -1%	Classical 1	81%	91%%	97%
	Piano	60%	56%	61%
	Multiple Instruments	66%	59%	62%
	Pop 1	70%	66%	72%
	Vocal1	61%	57%	64%
	Tabla	82%	81%	97%
TSM -2%	Classical 1	81%	83%	88%
	Piano	62%	51%	67%
	Multiple Instruments	39%	50%	59%
	Pop 1	47%	57%	64%
	Vocal1	28%	49%	57%
	Tabla	79%	78%	91%
TSM +1%	Classical 1	82%	91%%	93%
	Piano	42%	51%	57%
	Multiple Instruments	39%	50%	59%
	Pop 1	47%	57%	64%
	Vocal1	28%	49%	57%
	Tabla	81%	86%	97%
TSM +2%	Classical 1	79%	78%	91%
	Piano	42%	51%	57%
	Multiple Instruments	39%	50%	59%
	Pop 1	47%	57%	64%
	Vocal1	28%	49%	57%
	Tabla	41%	51%	51%

Table 5. Comparison of performance of the proposed adaptive method against signal processing attacks to non-adaptive method

Non-adaptive method (Uniform DOE)

Audio Sample	No. of ROE used for water-marking	Threshold value required	+5% TSM		-5% TSM		Low pass filtering		MP3 compression	
			BER	NC	BER	NC	BER	NC	BER	NC
Tabla	4096	0.65	49%	0.52	48%	0.52	14%	0.91	17%	0.89
Classical 1	4096	0.58	46%	0.53	47%	0.52	15%	0.90	21%	0.86
Classical 2	4096	0.53	42%	0.57	36%	0.63	21%	0.77	25%	0.73
Instruments	4096	0.49	24%	0.82	24%	0.83	11%	0.93	19%	0.88
Country 1	4096	0.51	27%	0.80	29%	0.78	12%	0.92	21%	0.86
Country 2	4096	0.57	31%	0.68	34%	0.66	19%	0.88	25%	0.79
Pop 1	4096	0.56	32%	0.68	34%	0.66	18%	0.88	27%	0.80
Pop 2	4096	0.58	31%	0.66	33%	0.68	20%	0.82	26%	0.81
Blues	4096	0.54	22%	0.89	31%	0.72	13%	0.9	22%	0.75
Folk1	4096	0.49	43%	0.57	36%	0.63	21%	0.78	24%	0.73
Folk2	4096	0.51	31%	0.65	34%	0.67	19%	0.88	28%	0.81
Piano	4096	0.56	26%	0.76	29%	0.69	18%	0.88	27%	0.80
Flute	4096	0.44	28%	0.71	31%	0.67	21%	0.80	26%	0.72
Guitar1	4096	0.53	31%	0.67	34%	0.67	20%	0.81	22%	0.80
Guitar2	4096	0.57	30%	0.66	34%	0.66	19%	0.80	22%	0.78
Vocal 1	4096	0.48	23%	0.82	24%	0.77	12%	0.93	20%	0.81
Vocal 2	4096	0.43	22%	0.79	28%	0.73	26%	0.73	22%	0.79
Sitar 1	4096	0.57	21%	0.80	26%	0.71	22%	0.81	19%	0.82
Sitar 2	4096	0.56	31%	0.67	34%	0.67	19%	0.81	22%	0.78
Violin 1	4096	0.42	36%	0.62	33%	0.67	24%	0.73	26%	0.73

Proposed adaptive method (DOE based on localized property)

Tabla	887	0.78	41%	0.61	40%	0.62	8%	0.97	11%	0.93
Classical 1	1022	0.69	41%	0.58	40%	0.59	6%	0.96	12%	0.92
Classical 2	1019	0.68	33%	0.66	22%	0.75	9%	0.96	15%	0.90
Instruments	918	0.72	16%	0.90	15%	0.90	8%	0.97	11%	0.93
Country 1	987	0.77	15%	0.90	16%	0.90	7%	0.96	10%	0.94
Country 2	991	0.71	20%	0.79	19%	0.79	14%	0.89	14%	0.81
Pop 1	894	0.76	13%	0.91	14%	0.91	11%	0.94	15%	0.90
Pop 2	892	0.79	12%	0.90	12%	0.93	15%	0.92	16%	0.91
Blues	966	0.76	11%	0.92	23%	0.81	9%	0.93	15%	0.86
Folk1	1012	0.69	23%	0.68	22%	0.74	9%	0.96	14%	0.91
Folk2	899	0.78	13%	0.90	17%	0.91	12%	0.91	17%	0.88
Piano	981	0.73	17%	0.84	21%	0.80	11%	0.88	16%	0.86
Flute	1028	0.68	18%	81%	20%	0.81	0.12	0.88	18%	0.83
Guitar1	881	0.77	13%	0.92	14%	0.95	11%	0.93	11%	0.91
Guitar2	902	0.74	17%	0.84	13%	0.93	14%	0.95	13%	0.88
Vocal 1	919	0.71	15%	0.87	15%	0.92	8%	0.93	11%	0.90
Vocal 2	1031	0.61	19%	0.81	22%	0.77	19%	0.81	15%	0.86
Sitar 1	921	0.77	16%	0.82	19%	0.79	12%	0.89	12%	0.89
Sitar 2	913	0.73	16%	0.83	13%	0.94	14%	0.96	13%	0.87
Violin 1	1011	0.61	22%	0.77	25%	0.74	17%	0.82	17%	0.84

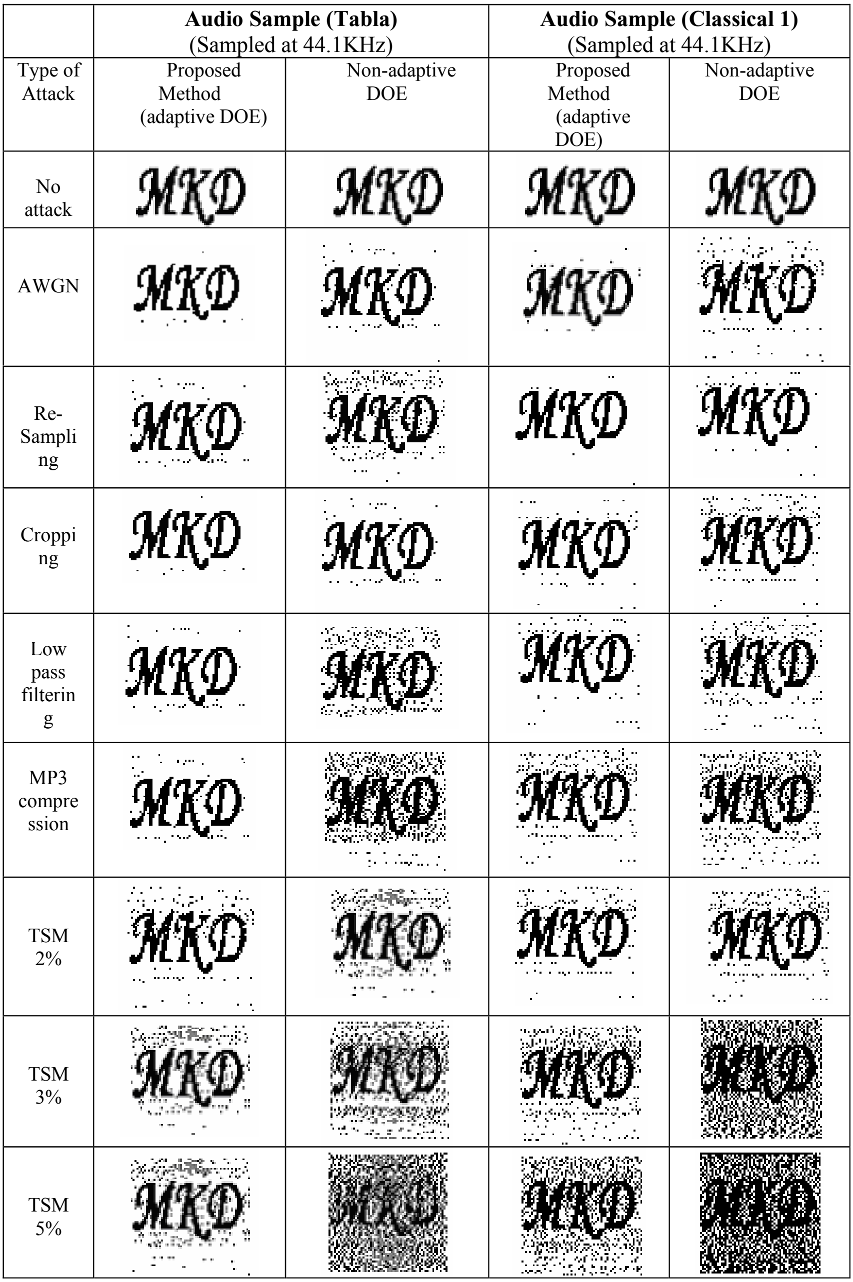

Fig. 6. Recovery of Binary Watermark under Signal Processing Attacks

9 Conclusion

An audio watermarking method presented in this paper is robust to seriously challenging synchronization attacks. High energy regions in the audio are selected for watermark embedding regions since these peaks tend to mask the neighboring audio data. A synchronization code is used for countering the positive false alarm generated due to data modification as a result of watermark embedding. Since synchronization code is embedded in localized regions, searching for these codes during watermark detection is computationally cheap. The watermark data is embedded in an adaptive manner in the audio signal. Since sharp transients are more resistant to synchronization attacks more data is embedded in these regions while less watermark data is embedded in less sharp transients. The watermarking is done in the SVD domain which makes the process perceptually transparent. The subjective listening tests have confirmed that the watermarking process is imperceptible to the human auditory system. The results obtained from robustness tests against signal processing attacks conclude that the proposed method is quite robust to attacks. Comparative results of the proposed adaptive watermarking method against uniform watermarking method reveals that the proposed method has comparatively better performance. However more optimized calculations for deciding *DOE* for reference points are required in which some other characteristics of the audio also needs to be taken care of.

References

1. Bender, W., Gruhl, D., Morimoto, N., Lu, A.: Techniques for data hiding. IBM Systems Journal 35(3-4), 313–336 (1996)
2. Katzenbeisser, S., Petitcolas, F.A.P. (eds.): Information Hiding Techniques for Steganography and Digital Watermarking. Artech House, Inc., Norwood (2000)
3. Gerzon, M.A., Graven, P.G.: A high-rate buried-data channel for audio CD. Journal of the Audio Engineering Society 43, 3–22 (1995)
4. Gruhl, D., et al.: Echo hiding. In: Anderson, R. (ed.) IH 1996. LNCS, vol. 1174, pp. 295–315. Springer, Heidelberg (1996)
5. Cooperman, M., Moskowitz, S.: Steganographic Method and Device, U.S. Patent 5613 004 (March 1997)
6. Huang, J., Wang, Y., Shi, Y.Q.: A blind audio watermarking algorithm with self-synchronization. In: Proc. IEEE Int. Symp. Circuits and Systems, vol. 3, pp. 627–630 (2002)
7. Akhaee, M.A., Saberian, M.J., Feizi, S., Marvasti, F.: Robust Audio Data Hiding Using Correlated Quantization With Histogram-Based Detector. IEEE Transactions on Multimedia, 834–842 (2009)
8. Bhat, V., Sengupta, I., Das, A.: Audio Watermarking Based on Mean Quantizationin Cepstrum Domain. In: International Conference on Advance Computing and Communications, pp. 73–77 (2008)
9. Arnold, M.: Audio watermarking: Features, applications, and algorithms. In: Proc. IEEE Int. Conf. Multimedia & Expo., vol. 2, pp. 1013–1016 (2000)
10. Wei, F.S., Feng, X., Mengyuan, L.: A Blind Audio Watermarking Scheme Using Peak Point Extraction. In: Proceedings of IEEE International Symposium on Circuits and Systems, vol. 5, pp. 4409–4412 (2005)

11. Li, W., Xue, X., Lu, P.: Localized Audio Watermarking Technique Robust Against Time-Scale Modification. IEEE Transactions on Multimedia 8, 60–69 (2006)
12. Dutta, M.K., Gupta, P., Pathak, V.K.: Biometric Based Audio Watermarking. In: Proceedings of International Conference on Multimedia Information Networking and Security, pp. 10–14 (2009)
13. Wu, S., Huang, J., Huang, D.: Efficiently Self-Synchronized Audio Watermarking for Assured Audio Data Transmission. IEEE Transactions on Broadcasting 51, 69–76 (2005)
14. Liu, Tan, T.: A SVD-Based Watermarking Scheme for Protecting Rightful Ownership. IEEE Transactions on Multimedia 4(1), 121–128 (2002)
15. Yavuz, E., Telatar, Z.: Improved SVD-DWT based digital image watermarking against watermark ambiguity. In: Proceedings of ACM Symposium on Applied Computing, pp. 1051–1055 (2007)
16. Voyatzis, G., Pitas, I.: Applications of toral automorphisms in image watermarking. In: Proceedings of International Conference on Image Procesing, vol. 1, pp. 273–240 (1996)
17. Duxbury, C., Davies, M.E., Sandier, M.B.: Separation of transient information in audio using multi-resolution analysis techniques. In: Proceedings of International Workshop on Digital Audio Effects, pp. 21–27 (December 2001)
18. Kennedy, M.P., Kolumban, G.: Digital communication using chaos. Elsevier Signal Processing 80, 1307–1320 (2000)
19. Liu, J.: The application of wavelet in image compression and digital watermarking. In: Doctoral dissertation, Zhejiang University, China (April 2001)
20. Chen, B., Wornell, G.W.: Quantization index modulation methods for digital watermarking and information embedding of multimedia. J. VLSI Signal Processing 27, 7–33 (2001)
21. Yamamoto, K., Iwakiri, M.: Real-Time Audio Watermarking Based on Characteristicsof PCM in Digital Instrument. Journal of Information Hiding and Multimedia Signal Processing 1(2), 59–71 (2010)
22. Nishimura, R., Abe, S.-i., Fujita, N., Suzuki, Y.: Reinforcement of VoIP Security with Multipath Routing and Secret Sharing Scheme. Journal of Information Hiding and Multimedia Signal Processing 1(3), 204–219 (2010)
23. Wang, X.-Y., Niu, P.-P., Yang, H.Y.: A Robust, Digital-Audio Watermarking Method" Multimedia. Multimedia 16(3), 60–69 (2009)
24. Kalantari, N.K., Akhaee, M.A., Ahadi, S.M., Amindavar, H.: Robust Multiplicative Patchwork Method for Audio Watermarking. IEEE Transactions on Audio, Speech, and Language Processing 17(6), 1133–1141 (2009)
25. Girin Parvaix, M., Brossier, L.: A Watermarking-Based Method for Informed Source Separation of Audio Signals With a Single Sensor. IEEE Transactions on Audio, Speech, and Language Processing 18(6), 1467–1475 (2010)