AUDIO WATERMARKING USING HYBRID DWT-DCT-SVD TECHNIQUE

[1]Nalini Tomar, [*2]Sarika Singh, [3]C. V. Lakshmi, [4]C. Patvardhan

[1]Research Scholar, Faculty of Science, Dayalbagh, Agra (UP)
[2] Guest Faculty, Faculty of Engineering, Dayalbagh, Agra (UP)
[3] Professor, Faculty of Science, Dayalbagh, Agra (UP)
[4] Professor, Faculty of Engineering, Dayalbagh, Agra (UP)

Corresponding author's details-
[*2]Sarika Singh
[2]Guest Faculty, Faculty of Engineering, Dayalbagh Agra (UP)*
Email id: sarika_singh@yahoo.com

ABSTRACT:

Copyright protection of multimedia signals is a major problem in today's world. Efforts are made in the literature to protect the multimedia signals by adding a watermark to the original signal. But the watermark has to be robust enough as attackers are also equally on the job to destroy the embedded watermark. This work is about how to protect an audio signal using watermarking technique. Extensive stress testing is done on possible attacks to destroy the watermark. The results are reported on 64 such attacks. A hybridization of three main techniques Discrete Wavelet Transform (DWT), Discrete Cosine Transform (DCT) and Singular Value Decomposition (SVD) is used to gain the advantages of their complementary features to provide maximum possible security and copyright protection for audio signals. This hybridization is done in a slightly different order from what is mentioned in the literature which provided better results. The results show that this technique is robust against a large number of attacks and is imperceptible.

Keywords: *Audio Watermarking, DWT, DCT, SVD, Arnold Transform, Hybrid*

1. Introduction

Copyright protection of multimedia signals is of utmost importance these days owing to the advances in digital communication technologies. Anti-social elements are at every nook and corner waiting for an opportunity to steal or copy information, modify it and spread it out as their own, with the use of new advanced internet and multimedia technologies. Invisible digital watermarking is one of the most promising techniques for copyright protection, where, a watermark in the form of text, image or audio can be embedded in the original signal intelligently and can be extracted at a later stage and verify the source where the signal is tampered. The multimedia signal to be protected can be an image, video or audio. Audio watermarking is not easy because of the sensitivity of human auditory system. A little change makes a huge change in audio sample [1]. The challenge therefore is to embed something into the audio signal in such a way that the original audio is unaffected and at the same time robust to several attacks. Watermarking techniques can also be classified based on the methods of embedding and extracting, i.e. Non-Blind (requires both the original signal and secret key to detect and extract the watermark), Semi-Blind (requires the secret key and information of watermarking bits) and Blind (requires only the secret key to extract the watermark). The main focus of this work is on Blind Audio Watermarking, in which secret information in the form of image (watermark image) is embedded into the audio signal based on a secret key. The general process for embedding and extraction of an audio watermark is shown in figures Figure 1. Block **diagram for embedding** Figure .

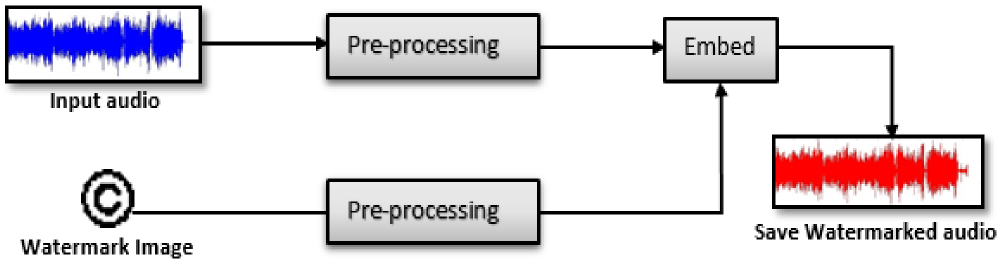


Figure 1. Block diagram for embedding

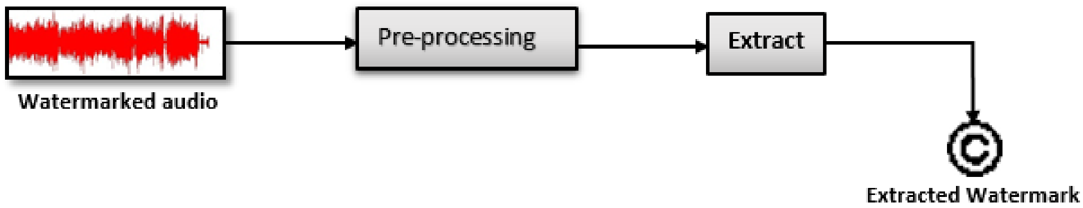


Figure 2. Block diagram for extracting

Singha [2] and Elshazly [3] proposed a digital watermarking technique by using multi-level DWT and SVD. Many researchers have tried combination of these two techniques. Kanhe [4] presented an audio watermarking technique which is based on DCT and SVD. Rizk [5] used all three techniques DCT, DWT and SVD (DCT-SVD-DWT-SVD) for watermarking an audio signal in that order and claimed minimum NC of 0.9581. Nayyar [6] proposed an audio watermarking approach using DWT-SVD and Arnold transform. Navjot Kaur [7] proposed a technique for audio watermarking using Arnold Transform with DWT-DCT, reporting NC values ranging from 0.82 to 1.00 and preserved imperceptibility. DCT techniques are good at resisting audio compression attacks [8]. Compression based on DWT preserves scalability. One such attempt [9] presented a synchronization invariant audio watermarking scheme based on DWT and DCT and reported high NC values ranging from 0.89 to 1.00 with low BER between 0 and 0.07. But it is less robust against pitch shifting, time scale modification. Hooman Nikmehr [10] used the combination of Discrete Wavelet and Cosine Transforms and showed that this combination is robust to re-quantization attack. Sujata Pathak [11] used the same combination at 3rd level. These combinations are robust to some attacks but not all. A few considered SVD as the best tool [12] in combination with the other two. CAI Yong-mei [13] proposed a blind audio watermarking scheme using the combination of DWT and SVD, and reported that their algorithm preserves better transparency and high capacity but also has a drawback as it does not preserve robustness against random cropping and time scale modification. Huan Zhao [14] used SVD-DWT combination for audio watermarking and reported NC values between 0.90 to 1.0. Khalid A. Darabkh [15] also proposed an imperceptible and robust DWT-SVD based digital audio watermarking algorithm, and mentioned high NC values. Recently some researchers have shown that the combination of the three DWT, DCT and SVD gives better results as compared to others for image watermarking [8][9]. D. Ambika tried the same combination of DWT-DCT-SVD and mentioned that this combination works best for audio watermarking but the specific details of how this combination is used are not mentioned [1]. Their results were given on a set of four speech audio signals but the details of the signals are not listed. Also no attacks were performed on those signals. Therefore, the proposed algorithm is an attempt to effectively combine the three transforms DWT-DCT-SVD combination. This combination is tested for robustness against various attacks. In this scheme, first, a vector is obtained by collecting maximum energy values of singular matrix, which is obtained by applying SVD on the audio signal. Then, DWT is applied on that vector and after that DCT is applied on the vector obtained as the result of DWT. The last vector is modified to embed watermark bits at appropriate positions.

The rest of the document is organized as follow: In Section 2 a brief description of DWT, DCT and SVD is given. The proposed methodology of combining DWT, DCT and SVD is explained in Section 3. A brief description of the attacks used for testing the robustness of this combination is given in Section 4 along with experimental results. Conclusions are given in Section 5.

2. Preliminaries

2.1 Discrete Wavelet Transform (DWT)

Discrete Wavelet Transform is a popular fast wavelet transform based on sub-bands. It decomposes the signal into two parts i.e. low frequency and high frequency bands (shown in equations 1 and 2) also called as approximate part and detailed part of signal respectively. Low frequency signals change slowly and contain much of the energy of the audio signal. High frequency signals change rapidly and have less energy. Hence, the approximate part is chosen for embedding in the proposed algorithm.

$$y_{low}[n] = \sum_{k=-\infty}^{\infty} x[k] \times g \times [2n - k] \tag{1}$$

$$y_{high}[n] = \sum_{k=-\infty}^{\infty} x[k] \times h \times [2n - k] \tag{2}$$

where, y_{low} and y_{high} are low frequency and high frequency bands respectively.
g and h are low pass and high pass filter respectively,
x is the audio signal on which DWT is performed.

Wavelet filter and level of decomposition are chosen based on the characteristics of the algorithm. Therefore, digital watermarking is flexible [13] and the signal can be decomposed at multiple levels in the algorithm. In the proposed algorithm, the signal is decomposed at level 3 and the 3rd level approximate part is considered for embedding as shown in Fig. Haar wavelet filter is used for decomposition, in which the filters used for decomposition and reconstruction of signal are orthogonal to each other. Haar wavelet is best suited for real-time processing because of its fast computation speed and simple implementation as compared to other types of wavelet.

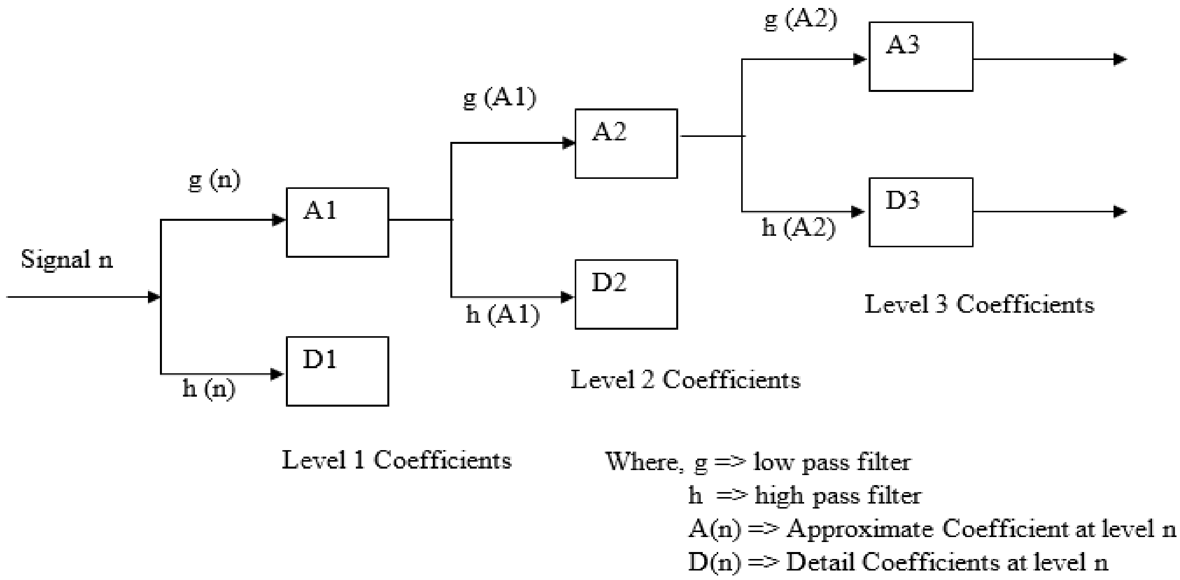


Figure 3. DWT decomposition at level 3

2.2 Discrete Cosine Transform (DCT)

$$y(k) = w(k) \sum_{n=1}^{N} x(n) \cos\left(\frac{\pi(2n-1)(k-1)}{2N}\right) \quad k = 1,2,3,\ldots\ldots,N \tag{3}$$

$$x(n) = w(k) \sum_{n=1}^{N} y(k) \cos\left(\frac{\pi(2n-1)(k-1)}{2N}\right) \quad n = 1,2,3,\ldots\ldots,N \tag{4}$$

where, $w(k) = \begin{cases} \frac{1}{\sqrt{N}} & k = 1 \\ \sqrt{\frac{2}{N}} & 2 \le k \le N \end{cases}$, N is the length of x . The size of x and y is same.

2.3 Singular Value Decomposition (SVD)

SVD is a factorization technique of any real or complex matrix. It is popular in the field of signal processing and statistics. SVD is preferred for digital audio watermarking as it helps in making the embedded watermark in the audio robust to transpose, scaling and geometric distortions.

SVD factorizes a $(m \times n)$ matrix A as follows:

$$M = USV^T = U \begin{vmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & \dots \\ \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \sigma_r \end{vmatrix} V^T \qquad (5)$$

Where, U and V are unitary matrices of size $m \times m$ and $n \times n$ respectively. The columns of U and V are known as left singular and right singular vectors respectively. And S is a $m \times n$ diagonal matrix whose elements are real positive singular values of M, represented as eigenvalues. The elements of S are arranged in decreasing order as follows:

$$\sigma_1 \ge \sigma_2 \ge \sigma_3 \ge \cdots \ge \sigma_r \qquad (6)$$

Small changes in singular values of matrix S, do not result in large changes in the original matrix M, when it is reconstructed. Also the singular values mainly focus on the high energy part of audio signal. Hence, in the proposed algorithm matrix S is used to embed watermark bits.

2.4 Arnold Transform

Arnold Transform is a scrambling method for images also known as cat face transform (Li et al.,2013). It is periodic in nature. To protect watermark embedded into the signal and to make the algorithm more robust, the watermark image is scrambled using Arnold Transform before embedding. It changes the pixels of an image and produces a scrambled image after some predefined iteration, and makes it hard to identify the original image. The 2-D Arnold Transform can be defined as follows:

$$\left[\frac{A'}{B'}\right] = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \left[\frac{A}{B}\right] \qquad (7)$$

Where, $\left[\frac{A}{B}\right]$ represent the co-ordinates of original image, and $\left[\frac{A'}{B'}\right]$ represents the co-ordinates of scrambled image.

Because of its periodic nature the original image can be reconstructed back by applying Inverse Arnold Transform that can be defined as follows:

$$\left[\frac{A_1'}{B_1'}\right] = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} \left[\frac{A_1}{B_1}\right] \qquad (8)$$

Where, $\begin{bmatrix} \frac{A_1}{B_1} \end{bmatrix}$ represent the co-ordinates of scrambled image, and $\begin{bmatrix} \frac{A_1'}{B_1'} \end{bmatrix}$ represents the co-ordinates of original image.

3. Proposed Hybrid Algorithm

In the proposed algorithm, SVD is first applied on the audio sample and a vector X of maximum values of S is created and DWT is applied on X. Approximate coefficients obtained through DWT are taken and DCT is applied on them. The watermark bits are embedded in the resulting vector. The Proposed hybrid algorithm utilizing all the concepts desired in section II is outlined in this section.

Part A describes the embedding process and Part B the extraction process.

3.1 Embedding

3.1.1 Pseudo Code for embedding

i. Perform block pre-processing. Read Audio signal and segment it into n, non-overlapping blocks of size 8×8 respectively.

ii. Pre-process the watermark image using Arnold Transform.

iii. Apply SVD transformation on each block.

$$[Ui, Si, Vi] = SVD(bi) \; where \; bi = ith \; block, 1 \leq i \leq n \qquad (9)$$

iv. Get vector S formed by maximum values of Si and then DWT is applied.

$$S = [S1, S2, S3, \dots \dots \dots \dots S4] \qquad (10)$$

$$[A3, D1, D2, D3] = DWT\ (S, 3,' haar') \qquad (11)$$

v. Apply DCT on approximate coefficient A3 which returns vector CA.

$$CA = DCT(A3) \qquad (12)$$

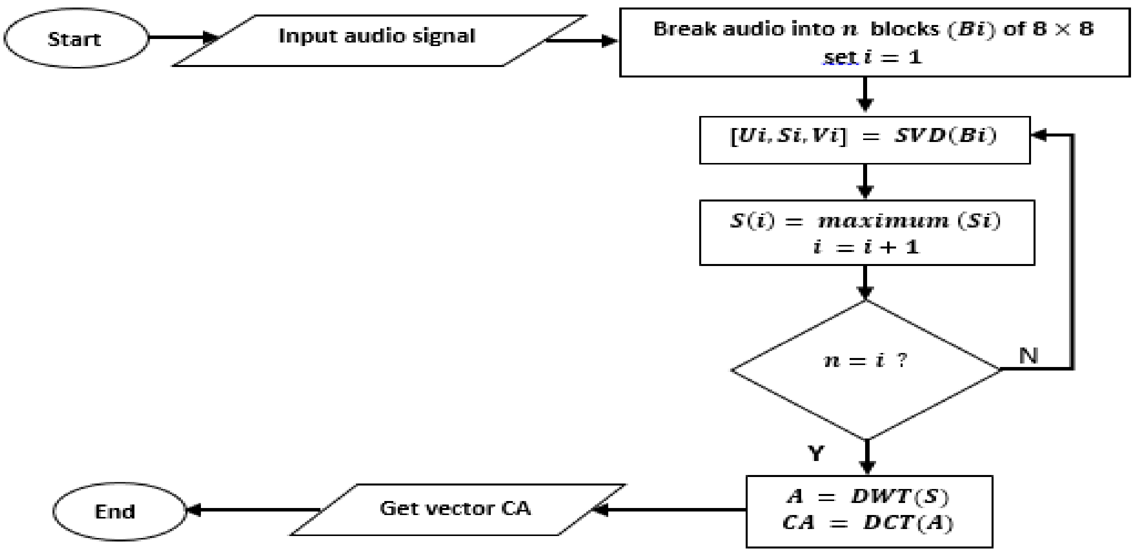


Figure 4. Flow-chart depicting pre-basic steps

vi. Divide vector CA into equal part lwb.

vii. Calculate energy of each j^{th} part and find avg of them, consider it to choose gain factor.

$$Ej = \left(\frac{1}{length(CAj)}\right) * \ sum\left(abs(CAj.^2)\right) \ where \ 1 \leq j \leq lwb \tag{13}$$

$$avg = \frac{E1 + E2 + E3 + \cdots + Ej}{j} \tag{14}$$

viii. Generate two pseudo random sequences and perform embedding with them.

$$CAj = CAj + \ \alpha \times Pn, \quad where, 1 \leq j \leq lwb, n = \{0,1\} \tag{15}$$

ix. Apply inverse DCT followed by inverse DWT.

$$A = IDCT(CA) \tag{16}$$

$$S = IDWT(A, D1, D2, D3) \tag{17}$$

x. Get vector S_i, perform inverse SVD.

$$b_i = U_i \times S_i \times V_i^T \tag{18}$$

xi. Rearrange blocks into vector and save.

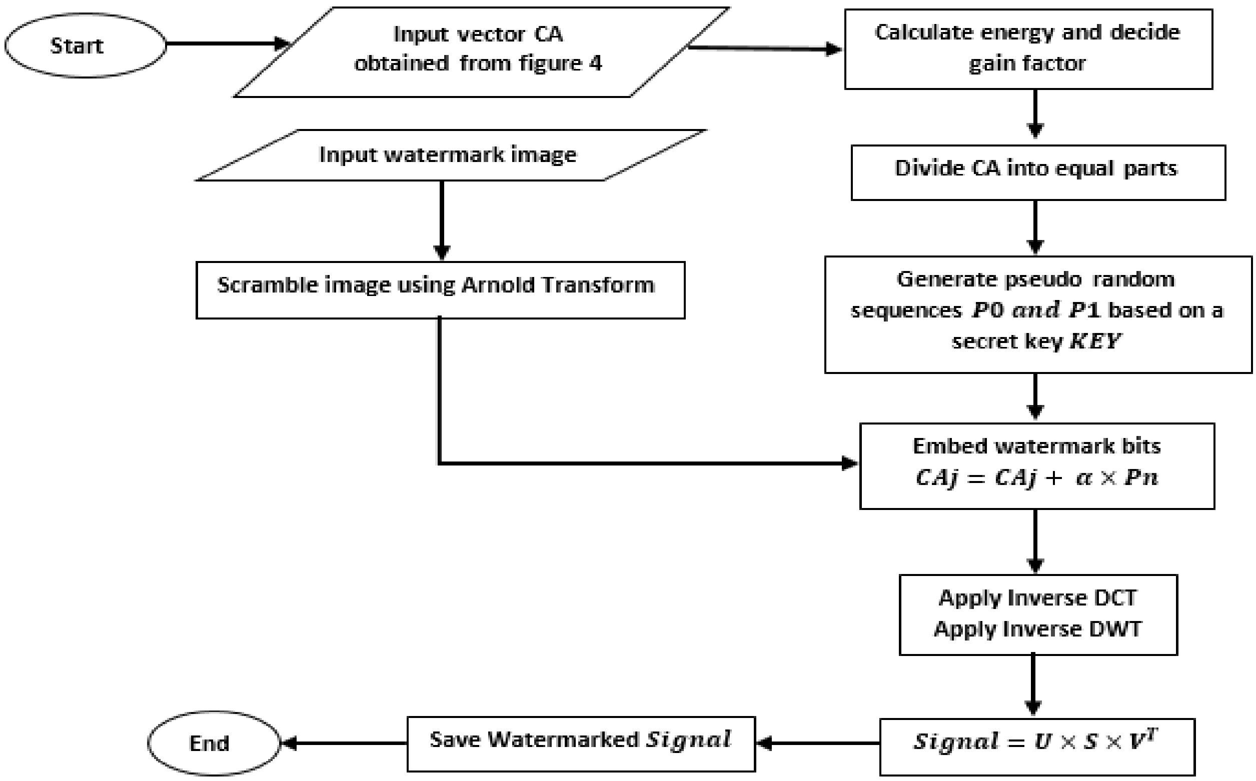


Figure 5. Flow-chart to depict watermark embedding

3.1.2 Detailed Description of pseudo code

i. First some block pre-processing steps are carried out, because it is found to be more robust than serial processing, especially against the cropping and compression attacks [6]. An audio file is read and stored as a vector. It is then segmented into $n,$ non-overlapping blocks of size 8×8 respectively.

ii. To pre-process the watermark image, Arnold Transform is used to scramble it in order to increase the security of embedded information [6].

iii. SVD is performed on each i^{th} block to get vectors Ui, Si and Vi. This is to provides more stability to the algorithm as small changes made to SVD vectors (especially S vector) does not result in much variation [8].

iv. Vector S is generated by considering maximum values of Si of each block bi, It is decomposed using DWT at 3^{rd} level to get $A3, D1, D2$ and $D3$. $A3$ is the approximate coefficient matrix of level 3 and $D1, D2, D3$ represent detailed coefficients at levels 1,2 and 3, respectively.

v. Approximate coefficient A3 is chosen for embedding as they represent low frequency part of audio signal which is less affected when some modifications are made to it [14]. To enhance robustness against compression and noise attacks the vector A3 is transformed using DCT which returns vector CA. DCT resists these attacks by converting a signal into sum of sinusoids and returns middle level coefficients that are disturbed by small changes.

vi. CA is divided into equal parts lwb which is calculated based on length of audio signal and watermark image vector. First set $G = 50$. And evaluate lwb and G as given in equation 13 and 14.

$$lwb = \begin{cases} nbit & if\ div \geq 50 \\ \lfloor L_{CA}/50 \rfloor & otherwise \end{cases} \qquad (19)$$

$$G = \begin{cases} \lfloor div \rfloor & if\ div \geq 50 \\ G & otherwise \end{cases} \qquad (20)$$

Where, $L_{CA} = length(CA)$z

$$nbit = length\ (watermark\ bits)$$

vii. Energy of each j^{th} part is calculated to find the average avg of these energies $E1, E2 \ldots\ldots\ldots, Ej$ represented in equations 15 and 16. Gain factor is chosen depending on $avg\ as\ \alpha$ ($\alpha1$ or $\alpha2$). If $Ej \geq avg$ then $\alpha1$ otherwise $\alpha2$ is chosen, where $\alpha1 > \alpha2$. Choosing α in this manner increases reliability and imperceptibility of the algorithm as low energy signal is modified by tiny gain factor and high energy signal is modified by large gain factor.

viii. Two pseudo random sequences $P0$ and $P1$ are generated using a secret key of size $G/2$ which increases the level of security. $P0$ and $P1$ Sequences are used to embed watermark bits 0 and 1 respectively as depicted in equation 17.

ix. Inverse DCT is applied on modified vector CA to get vector A. Inverse DWT is applied using A, D1, D2, D3 to get vector S.

x. Original signal S_i is modified using stored index of the previously selected (maximum) elements $S(i)$ of each block. Inverse SVD is applied to restore each block b_i.

xi. Blocks are rearranged into vector to save as watermark image.

3.2 Extraction

Extraction process is as follows and also depicted in Figure 6. **Flow-chart depicting Watermark extraction**

i. Steps 1-4 are applied as in embedding to get CA which is then divided it into equal parts lwb.

ii. Pseudo random sequences are generated $P0$ and $P1$ using same key as in embedding.

iii. correlation between each part CA_i to both sequences $P0$ and $P1$ $is\ checked$ to assign bit associated with the corresponding pseudo random sequence for which correlation is greater.

$$Wi = \begin{cases} 0, & CORR(CAi, P0) \geq CORR(CAi, P1) \\ 1, & CORR(CAi, P0) < CORR(CAi, P1) \end{cases} \quad where\ 1 \leq i \leq lwb \tag{21}$$

iv. Perform inverse Arnold transform on W and save obtained image as extracted watermark.

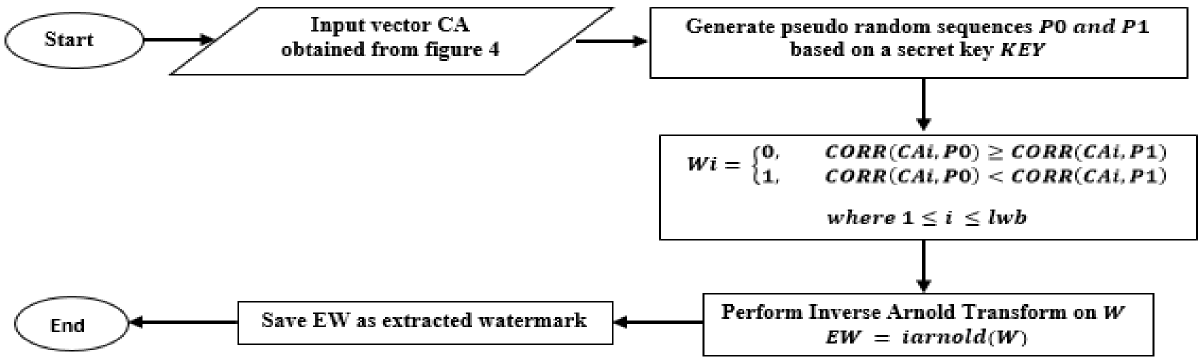


Figure 6. Flow-chart depicting Watermark extraction

4. Experiments & Results

Experiments are performed on 4 different audio signals (given in table 1) using 2 watermark images (shown in Figure 7. ***Flow-chart* depicting Watermark extraction**

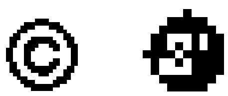


Figure 7. Flow-chart depicting Watermark extraction

Table 1. Tested Audio

S. No.	Audio signal	Length
1.	Classical	1 min 44 sec
2.	Pop	5 sec
3.	Loopy	2 min 56 sec
4.	Human Voice	2 min 53 sec

Each audio file mentioned in the table is a wave file of mono type whose sampling rate is 44100 Hz. Both watermarks are square images of size 10×10. Classical and Pop signal are standard signals taken from literature [14][16][17]. A song titled "You and me" is chosen as Loopy Music signal taken from a movie album. Human voice audio signal is a song sung by one of the authors.

4.1 Robustness Test (Attacks)

The algorithm is tested against 21 attacks. The proposed algorithm is shown to be robust against pitch shifting too which is often quoted as tough [14][17]. Details of the attacks tested for are as follows:

4.1.1 Resample: The audio signal is first down sampled from 44100 Hz to 8000 and then again up-sampled back to 44100 Hz.

4.1.2 Re-quantization: The 16 bit audio signal is Re-quantized to 8 bit sample and again reconverted to 16 bits.

4.1.3 Crop: Cropping is done in 3 different ways. In first case, the audio signal is cropped at three random positions. Few consecutive bits starting at each position are deleted. The second case is similar to the first case except that two random positions are chosen as against three. In third case, a few bits picked at random from various positions over the length of the audio are deleted.

4.1.4 Add White Noise: To add white Gaussian noise the signal power is measured and white noise is then added at 0.1 db signal-to-noise ratio.

4.1.5 Compression: In compression low frequency signals of the audio being watermarked are modified. The higher frequency signals are attenuated resulting in compression of the signal. Compression rate is set at -0.5 with filter at rate 0.5.

4.1.6 Expand Signal: In this operation, the audio signal is expanded at rate 0.5 with filter rate 0.5 in this method.

4.1.7 Repeat: A random portion of the audio signal is repeated.

4.1.8 Ring Modulation: To create this effect audio signal x(n) is modulated by multiplying it with a sine wave m(n) having a carrier frequency $f_c = 540$.

a. $$y(n) = x(n) \times m(n) \qquad (22)$$

4.1.9 Echo Delay, Flanger Effect and Chorus Effect: These effects are generated by using a combination of Comb filter (FIR or IIR) and modulation techniques.

Table 2. Delay Range and Modulation of Echo, Flanger and Chorus Effect

Effect	Delay Range (ms)	Modulation
Flanger Effect	0 – 15	Sinusoidal (_ 1 Hz)
Chorus Effect	10 – 25	Random
Echo Delay	>50	None

Parameters used for chorus effect are, delay length = 0.013sec, modulation depth = 0.003 sec, modulation rate = 1.00 Hz, feedback = 0.30%, low shelf frequency = 600 Hz, low shelf gain

= -7 dB and dry wet balance = 0.40. Approximately 4 sec delay is performed for echo. Flanger effect is conducted at the rate 1 Hz with 3 ms delay in a second.

4.1.10 Flip Sample: *A* set of 1000 data bits of the audio are swapped from two randomly selected positions.

4.1.11 Filter: A basic Butterworth high pass filter is applied on the audio with normalized cut-off frequency 0.001Hz.

4.1.12 Shelving Effect: Four types of shelving filters are applied on the signal namely Cut shelf, Base shelf, Treble shelf and Treble Cut shelf. Parameters used for shelving are listed in Table 3.

Table 3. ***Parameters used for shelving***

Shelving Method	Logarithmic Gain	Centre frequency	Slope
Base Shelf & Cut Shelf	4 dB	300 Hz	3
Treble Shelf & Treble Cut Shelf	4 dB	600 Hz	3

4.1.13 FIR Filter: This filter adds a time (T) delayed version of the audio signal to the original signal at some amplitude gain g = 0.5.

4.1.14 IIR: It delays the original signal by a delay time T, attenuates it by $g = 0.5$ and scales it by a degree c to compensate for high amplification of the structure.

4.1.15 Universal Comb Filter (Uni-Comb): The FIR and IIR filters are combined together in Uni-Comb filter.

4.1.16 Limiter: This effect is used for lowering the peaks of an audio signal. It measures all the high peaks and down scales them if they are higher than a fixed threshold, by which the overall signal is reduced. This effect becomes an attack for audio because in many of the algorithms, the watermark bits are embedded in higher energy part. In this work, limiter attack is performed using slope 1, trash 0.5 and threshold rate 0.01.

4.1.17 Matpan: Matrix based panning of mono sound to stereo is performed with -40 degrees initial angle and +40 degrees final angle of 40°.

4.1.18 Overdrive: Symmetrical clipping is performed, in which the positive and negative level of clipping is same. The soft-clipping at three-layers is tested , which is defined as follows:

$$f(x) = \begin{cases} 2x & for\ 0 \le x \le t \\ \dfrac{3-(2-3x)^2}{3} & for\ t \le x \le 2t \\ 1 & for\ \ 2t \le x \le 1 \end{cases} \qquad (29)$$

where threshold $t = 1/3$.

4.1.19 Fuzz/Distortion: The distortion can be defined by a non-linear exponential function as follows:

$$f(x) = \frac{x}{|x|}(1 - e^{\alpha x^2/|x|}) \qquad (30)$$

where the gain factor $\alpha = 6$. It controls the level of distortion.

4.1.20 Vibrato: Time delay is periodically based on a delay line and a low frequency oscillator is used to control changes in delay. Typical delay time = 5–10 Ms and LFO rate =5–14Hz.

4.1.21 Schroeder: To perform this attack reverberator design of Schroeder is used, which combines 4 Comb filters together with 2 All Pass filters. The gain factor for all pass filter and direct signal is taken as 0.5.

To check the similarity between embedded watermark and extracted watermark, two estimation formulae are used, which are as follows:

NC (Normalized Correlation Coefficient) defined as:

$$NC(W, W') = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} W(i,j)W'(i,j)}{\sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} W^2(i,j)}\sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} {W'}^2(i,j)}} \qquad (31)$$

where, W and W' are original and extracted watermarks, respectively and MSE defined as:

$$MSE = \left(\frac{1}{MN}\right)\sum_{i=1}^{M}\sum_{j=1}^{N}(W_{ij} - W'_{ij})^2 \qquad (32)$$

Where, W_{ij} and W'_{ij}, are the co-ordinates of original watermark and extracted watermark, M and N are the number of column and rows of both watermark image.

4.2 Imperceptibility Test

To test for imperceptibility of the audio watermarking algorithm, SNR (signal-to-noise ratio) is calculated between original and watermarked audio signals and tabulated in Table 4. SNR is calculated as given in equation 33.

$$SNR(dB) = 10log_{10}\frac{\sum_n A_n^2}{\sum_n (A_n - A'_n)^2} \qquad (33)$$

Table 4. SNR between original and watermarked audio

Audio Signal	SNR(dB)
Classical	19.48
Human Audio	23.56
Pop	20.33
Loopy Music	23.10

In most of the attacks the audio signal is highly distorted and Signal-to-noise ratio (SNR) value ranges from 15 DB to 20 DB, but this algorithm is found to be imperceptible. In the case of down SNR too, this algorithm works better for example, in ring modulation SNR goes down in negative as -4.5. But the proposed algorithm found to be imperceptible and watermark would be extracted with little distortion.

This algorithm has been tested for four audio signals which are mentioned in table 1. Depending on the type of audio signal different values for α1 and α2 are selected within the range 0.1 to 2. Values chosen for these four audio signals are given in table 5.

Table 5. Values of α chosen for different audio signals

Signal	$\alpha 1$	$\alpha 2$
Classical	0.2	0.5
Pop	1	1.5
Human Audio	0.1	0.5
Loopy Music	1.5	2

All measures for audio signals with respect to both watermark images against different attacks are mentioned in tables 6 – 9. It can be clearly observed from these tables that best results are found for the classical signal. Most attacks affect high peaks and this signal has less high peaks and thus is affected less.

Table 6. *Result for Classical Audio*

Attacks & Effects	Watermark 1			Watermark 2		
	NC	MSE	Extracted Water - mark	NC	MSE	Extracted Water - mark
Without Attack	1	0		1	0	
Resample 44khz to 8khz	1	0		1	0	
Re-quantize 16 bit to 8bit	1	0		1	0	
Crop 2 points	0.98	0.02		1	0	
Crop 3 points	0.92	0.09		0.98	0.01	
Add white noise	0.99	0.01		0.98	0.01	
Chorus	0.99	0.01		0.98	0.01	
Crop	0.99	0.01		1	0	
Echo Delay	1	0		0.98	0.01	
Flip Sample	1	0		1	0	
Repeat	1	0		1	0	
Base Shelf	1	0		1	0	
Trebel Shelf	0.99	0.01		0.97	0.02	
Treble Cut Shelf	1	0		1	0	
Base Cut Shelf	1	0		1	0	
Compressed	0.99	0.01		1	0	
Expander	0.99	0.01		0.98	0.01	
Filter	1	0		1	0	
Falnger	1	0		1	0	

FIR	1	0		1	0	
Fuzz	1	0		1	0	
IIR	1	0		1	0	
Limiter	1	0		1	0	
Matpan	1	0		1	0	
Over -drive	1	0		1	0	
Ringmod	1	0		1	0	
Schro-eder	0.94	0.06		0.87	0.11	
Unicomb	1	0		1	0	
Vibrato 0.00005	1	0		1	0	
Vibrato 0.0008	1	0		1	0	

Table 7 Result for Pop Audio

Attacks & Effects	**Watermark 1**			**Watermark 2**		
	NC	**MSE**	**Extracted Water -mark**	**NC**	**MSE**	**Extracted Water -mark**
Without Attack	0.95	0.05		0.96	0.3	
Resample 44khz to 8khz	0.94	0.06		0.96	0.3	
Re-quantize 16 bit to 8bit	0.95	0.05		0.96	0.3	
Crop 2 points	0.95	0.05		0.95	0.4	
Crop 3 points	0.79	0.24		0.93	0.06	
Add white noise	0.94	0.07		0.96	0.3	
Chorus	0.94	0.07		0.96	0.3	
Crop	0.94	0.06		0.95	0.04	
Echo Delay	0.95	0.05		0.97	0.02	
Flip Sample	0.94	0.06		0.95	0.04	
Repeat	0.93	0.08		0.95	0.04	
Base Shelf	0.94	0.06		0.94	0.05	
Trebel Shelf	0.95	0.05		0.97	0.02	
Treble Cut Shelf	0.95	0.05		0.96	0.3	
Base Cut Shelf	0.95	0.05		0.96	0.3	
Compressed	0.95	0.05		0.97	0.02	
Expander	0.94	0.06		0.95	0.04	
Filter	0.94	0.06		0.96	0.3	
Falnger	0.94	0.06		0.96	0.3	

FIR	0.94	0.06		0.95	0.04	
Fuzz	0.95	0.05		0.97	0.02	
IIR	0.94	0.06		0.95	0.04	
Limiter	0.95	0.05		0.96	0.3	
Matpan	0.96	0.04		0.96	0.3	
Over -drive	0.95	0.05		0.96	0.3	
Ringmod	0.94	0.06		0.97	0.02	
Schro-eder	0.87	0.14		0.83	0.14	
Unicomb	0.94	0.06		0.96	0.3	
Vibrato 0.00005	0.94	0.06		0.95	0.04	
Vibrato 0.0008	0.94	0.06		0.96	0.3	

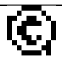
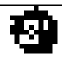
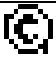
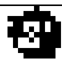
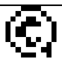
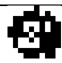
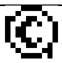
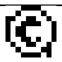
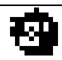
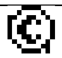
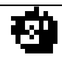
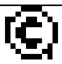
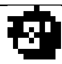
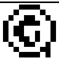
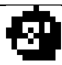
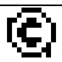
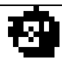
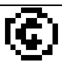
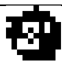
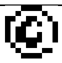
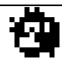
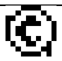
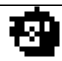
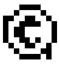
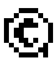
Table 8 Result for Human Audio

Attacks & Effects	**Watermark 1**			**Watermark 2**		
	NC	**MSE**	**Extracted Water -mark**	**NC**	**MSE**	**Extracted Water -mark**
Without Attack	0.94	0.06		0.95	0.04	
Resample 44khz to 8khz	0.94	0.07		0.95	0.04	
Re-quantize 16 bit to 8bit	0.94	0.06		0.95	0.04	
Crop 2 points	0.85	0.17		0.94	0.05	
Crop 3 points	0.89	0.12		0.95	0.04	
Add white noise	0.91	0.1		0.94	0.05	
Chorus	0.94	0.07		0.94	0.05	
Crop	0.95	0.05		0.95	0.04	
Echo Delay	0.96	0.04		0.95	0.04	
Flip Sample	0.9568	0.05		0.9545	0.04	
Repeat	0.93	0.07		0.91	0.07	
Base Shelf	0.96	0.04		0.95	0.04	
Trebel Shelf	0.95	0.05		0.94	0.05	
Treble Cut Shelf	0.95	0.05		0.95	0.04	
Base Cut Shelf	0.95	0.05		0.95	0.04	
Compressed	0.96	0.04		0.96	0.03	
Expander	0.94	0.06		0.93	0.06	

Filter	0.95	0.05		0.93	0.06	
Falnger	0.92	0.09		0.95	0.04	
FIR	0.95	0.05		0.95	0.04	
Fuzz	0.97	0.03		0.96	0.03	
IIR	0.95	0.05		0.95	0.04	
Limiter	0.94	0.06		0.95	0.04	
Matpan	0.90	0.11		0.93	0.06	
Over -drive	0.94	0.06		0.95	0.04	
Ringmod	0.93	0.08		0.95	0.04	
Schro-eder	0.94	0.06		0.88	0.1	
Unicomb	0.94	0.06		0.95	0.04	
Vibrato 0.00005	0.94	0.06		0.95	0.04	
Vibrato 0.0008	0.94	0.07		0.95	0.04	

Table 9. ***Result for Loopy Music***

Attacks & Effects	**Watermark 1**			**Watermark 2**		
	NC	**MSE**	**Extracted Water -mark**	**NC**	**MSE**	**Extracted Water -mark**
Resample 44khz to 8khz	0.94	0.06		0.95	0.04	
Re-quantize 16 bit to 8bit	0.94	0.06		0.95	0.04	
Crop 2 points	0.96	0.04		0.95	0.04	
Crop 3 points	0.94	0.06		0.95	0.04	
Add white noise	0.97	0.03		0.94	0.05	
Chorus	0.97	0.03		0.95	0.04	
Crop	0.94	0.06		0.95	0.04	
Echo Delay	0.96	0.04		0.96	0.03	
Flip Sample	0.97	0.03		0.95	0.04	
Repeat	0.97	0.03		0.95	0.04	
Base Shelf	0.97	0.03		0.95	0.04	
Trebel Shelf	0.99	0.01		0.97	0.02	
Treble Cut Shelf	0.98	0.02		0.95	0.04	
Base Cut Shelf	0.98	0.02		0.95	0.04	
Compressed	0.98	0.02		0.95	0.04	

Expander	0.97	0.03		0.95	0.04	
Filter	0.97	0.03		0.95	0.04	
Falnger	0.96	0.04		0.94	0.05	
FIR	0.96	0.04		0.94	0.05	
Fuzz	0.97	0.03		0.95	0.04	
IIR	0.96	0.04		0.94	0.05	
Limiter	0.98	0.02		0.95	0.04	
Matpan	0.97	0.03		0.96	0.03	
Over -drive	0.98	0.02		0.95	0.04	
Ringmod	0.97	0.03		0.95	0.04	
Schro-eder	0.95	0.05		0.88	0.1	
Unicomb	0.97	0.03		0.95	0.04	
Vibrato 0.00005	0.97	0.03		0.95	0.04	
Vibrato 0.0008	0.97	0.03		0.95	0.04	

This algorithm takes the same watermark image as taken by Kaur et al. [6] in order to compare the results. Though audio signals in both cases are different, the four signals taken by us are comparable in SNR to two of the signals of Kaur et al, namely wav1 and wav2. The results of the proposed algorithm and Kaur et al. [6] are compared on the basis of NC and given in Table 1 with two of the sixty four attacks that are common in both papers.

Table 10. Comparative Results

Authors	Audio Signal	SNR	Without Attack	White Gaussian Noise	Compressed
[6]	Wav1	21.2496	1	1	0.9971
	WAV2	25.4476	1	1	1
Proposed Approach	Classical	19.48	1	0.99	0.99
	Human Audio	23.56	0.99	0.94	0.95
	Pop	20.33	0.94	0.91	0.96
	Loopy Music	23.10	-	0.97	0.98

All the three techniques DCT DWT and SVD also used by Rizk [5] and tested on a good quality audio file having SNR of 58.1976 and claimed minimum NC of 0.9581. According to proposed algorithm all the three techniques are used and tested on audio file having less than SNR of 23.56 and claimed a range of NC from 0.85 to 1.

5. Conclusions

A hybrid algorithm utilizing DWT, DCT and SVD is presented in this work for watermarking audio signals. The basic idea of using the three together is to increase the robustness of the method. The results show that this algorithm is robust against a large number of attacks that are investigated in the paper. The embedded watermark is imperceptible too. The effectiveness of the algorithms is clearly shown by the high values of NC and the low values of MSE obtained with most of the attacks. None of the methods described in the literature have been checked against so many attacks till now. This is the first attempt that has presented the results of tests in such a comprehensive and detailed manner.

References

1. Ambika, D., Radha V., 2014. Speech Watermarking using DWT, DCT and SVD. International Journal of Computer Science & Engineering Technology (IJCSET). Vol. 5 No. 11, ISSN: 2229-3345, 1089-1093.
2. Singha, A., Ullah, M.A., 2022. Development of an audio watermarking with decentralization of the watermarks. Journal of King Saud University – Computer and Information Sciences. 34, 3055–3061.
3. Elshazly, A.R., Mohamed, E.N., Fouad, M.M., Fathi, E.A.S., 2021 Intelligent High Payload Audio Watermarking Algorithm Using Colour Image in DWT-SVD Domain. Journal of Physics: Conference Series, Journal of Physics: Conference Series, ICaTAS.
4. Kanhe, A., Gnanasekaran, A., 2017. Robust image-in-audio watermarking technique based on DCT-SVD transform. EURASIP journal on Audio, Speech and Music Processing.
5. Rizk, S., Khalifa, F., Mohamed, M.A., Mohy El-Din A. Abou-Soud., 2017. DCT-DWT-Based Audio Watermarking Using SVD. International Journal of Scientific & Engineering Research. 8, ISSUE 11.
6. Nayyar, R., Singh, R., Ritika., 2016. Improved Audio Watermarking Using Arnold Transform, DWT and Modified SVD. International Journal of Innovative Research in Science, Engineering and Technology (An ISO 3297: 2007 Certified Organization). Vol. 5, Issue 7.
7. Navjot Kaur and Usvir Kaur," Audio Watermarking using Arnold transformation with DWT-DCT", International Journal of Computer Science Engineering (IJCSE), Vol. 2 No.06 Nov 2013.
8. Madhesiya, Sangeeta, Ahmed, Shakil, 2013. Advanced Technique of Digital Watermarking based on SVD-DWT-DCT and Arnold Transform. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET). Vol 2, No 5.
9. Harish N J, B. B. S. Kumar, Ashok Kusagur: Hybrid Robust Watermarking Technique Based on DWT, DCT and SVD. ISSN (Print): 2278-8948, Volume-2, Issue-5, 2013.
10. Nikmehr, Hooman, Hashemy , Sina Tayefeh, 2010. A New Approach to Audio Watermarking Using Discrete Wavelet and Cosine Transforms. 1st International Conference on Communications Engineering.
11. Pathak, S., Patel, K.B., Haria, D.J., 2015. Audio Watermarking based on 3-level DWT. IJIRST –International Journal for Innovative Research in Science & Technology. VOL. 1.
12. Chang, C. C., Tsai, P., Lin, C. C., 2005. SVD-based digital image watermarking scheme. Pattern Recognition Letters 26. Vol. 26, 1577–1586.

13. Cai, Yong-mei, Guo, Wen-qiang Ding, , Hai-yang, 2013. An Audio Blind Watermarking Scheme Based on DWT-SVD. Journal Of Software. VOL. 8, NO. 7.
14. Zhao, Huan, Wang, Fei, Chen, Zuo, Liu, Jun, 2014. A Robust Audio Watermarking Algorithm based on SVD-DWT. Elektronika ir elektrotechnika. ISSN 1392-1215, VOL. 20, NO. 1.
15. Darabkh, K. A., 2014. Imperceptible and Robust DWT-SVD Based Digital Audio Watermarking Algorithm. Journal of Software Engineering and Applications. 7, 859-871.
16. Komal Shedge, Mangal Patil, J. S. Chitode: A New Facet in Robust Audio Watermarking in DWT-SVD Domain, 2014, IJERT, Vol. 3 Issue 5.
17. Xiang-Yang Wang, Hong. Zhao: A Novel Synchronization Invariant Audio Watermarking Scheme Based on DWT and DCT. IEEE Transaction on Signal Processing, VOL. 54, NO. 12, December 2006.