Robust Audio Watermarking Scheme with Synchronization Code and QIM

N.V.Lalitha [#1], Ch.Srinivasa Rao [*2], P.V.Y.Jaya Sree [$3]

[#] Department of ECE, GMR Institute of Technology
RAJAM-532 127, A.P. INDIA.

[*] Department of ECE, JNTU-K University,
Vizianagaram. A.P. INDIA.

[$] Department of ECE, GIT, GITAM University,
Visakhapatnam, A.P. INDIA.

[1] lalitha.nv@gmrit.org

[2] chsrao.ece@jntukucev.ac.in

[3] pvyjayasree@gitam.edu

Abstract— **Many blind audio watermarking schemes using FFT are available and are robust to signal processing attacks, but fail at de-synchronization attacks. A simple blind audio watermarking using Fast Fourier Transform (FFT) is explored in this work and its robustness is improved by using a synchronization code. In this process, initially the entire audio stream is segmented and each audio segment is divided into two parts. In the first part of the audio segment, a synchronization code is embedded and in the second part watermark is embedded. In the process of watermark embedding, the original audio is segmented into non-overlapping frames. A binary watermark image is encrypted using Gauss Map and embedded into each frame of FFT coefficients of the audio signal using Quantization Index Modulation (QIM). The analysis and results upon experimentation demonstrate that this method is superior to other state-of-art methods in terms of imperceptibility, security and payload. At the same time, it is effective against common signal processing attacks and de-synchronization attacks like signal addition, signal subtraction, cropping, MP3 compression and time-scale modification.**

Keyword-Audio watermarking, Quantization Index Modulation, Fast Fourier Transform, Gauss Map, Synchronization, Bit Error Rate, Precision, Payload, Mean Opinion Score.

I. INTRODUCTION

Advancement in audio technology increases the transmission and distribution of audio files is very easy, but at the same time this also allows illegal copying and distribution. Therefore, copyright protection and copy protection of audio files become a challenging issue. One feasible solution to overcome this problem is audio watermarking, in which copyright information is embedded in audio files to facilitate authenticity [1].

International Federation of the Phonographic Industry (IFPI) [2] states that the embedded audio must maintain more than 20dB Signal to Noise Ratio (SNR). An efficient watermarking method must satisfy four requirements, i.e., imperceptibility, robustness, security and payload [3]. Imperceptibility means embedding the watermark in the host audio should not be perceivable. Robustness means capability to extract the watermark from the attacked watermark embedded audio signal. Security refers to watermark should be detectable only by authorized persons. The payload is that, how many numbers of bits that can be embedded into the audio without loss of imperceptibility. A trade-off will always exist between these requirements.

Digital audio watermarking algorithms are categorized into two types, i.e., Time domain [4] and Frequency domain [5]. A watermark is inserted into the host signal directly in the time domain algorithms. In frequency domain algorithms, a watermark is inserted into frequency coefficients of an audio signal. Time domain algorithms are very easy to implement at the same time robustness is somewhat less compared to frequency domain algorithms [6], [7]. Among the transform techniques FFT is very domain because of its translation-invariant property and its less computational cost.

Quantization Index Modulation (QIM) is a non-linear technique used for digital watermarking and information hiding methods proposed in [8], [9]. The choice of step size in QIM should maintain the trade-off among imperceptibility, robustness and capacity. In the proposed method, the use of QIM maintains the trade-off for the above parameters and detection of watermark in blind approach is possible.

Synchronization attack is one of the main issues in the audio watermarking. In this paper, the synchronization code [5], [10], [11] is generated by the logistic chaotic method to embed into the audio for reducing the problem of de-synchronization attacks. Synchronization code is detected from the attacked watermarked audio with the help of correlation.

Dhar P.K.et al. [3] proposed a scheme using SVD and Cartesian-Polar Transform (CPT). The CPT component of the highest singular values of the low frequency FFT coefficients of each frame are used to embed the watermark bits. In [5], Wu et al presented a DWT based blind watermarking method which uses QIM. It is a self-synchronization technique and the synchronization code and the watermark data are embedded in the low frequency coefficients. In Lei B. et al. [6] work, the watermark is embedded in the low frequency components of Lifting Wavelet Transform (LWT) of the original audio signal using QIM technique and SVD. V. Bhat et al. [7] improved and made it adaptive using SVD and DWT and it also resist de-synchronization attack. V.K. Bhat et al. [12] proposed an audio watermarking algorithm using SVD and dither modulation quantization. In this paper, host audio is divided into blocks and SVD is applied to each block. Using dither-modulation quantization, the watermark bits are embedded in the singular values of each block. Lei B. et al. [13] proposed a blind algorithm using Discrete Cosine Transform (DCT) and SVD values. The synchronization code is generated using chaotic sequence and is inserted in the DCT-SVD coefficients of the host audio signal. M. Fallahpour et al. [14] exploited the Absolute Threshold of Hearing (ATH) of the HAS to choose the frequency band and segmented them into small frames for quantization. The payload is high in this algorithm.

The rest of this paper is organized as follows. FFT fundamentals, QIM and Synchronization code are discussed in section 2. In section 3, the proposed watermarking and extracting algorithms are explained. Section 4 demonstrates the performance of our algorithm and the obtained results. The concluding remarks are given in section 5.

II. TRANSFORM BASICS

Fast Fourier Transform (FFT) provides a useful analysis for audio signal applications. The translation invariant property of FFT is well utilized to embed watermarks since the coefficients can withstand slight variations in the signal in time domain. The transform based methods, as compared to time domain schemes, offer better imperceptibility and robustness against common attacks but with increased computational effort.

A. Fast Fourier Transform

Mathematically, FFT can be written as:

$$\mathrm{Y(k)} = \sum_{\mathrm{n=0}}^{\mathrm{N-1}} \mathrm{y(n)} e^{-2j\pi nk}, \qquad k = 0,1, \ldots\ldots N-1 \tag{1}$$

where $\mathrm{y(n)}$ is time domain input signal and $\mathrm{Y(k)}$ is in the form of the frequency domain. The complex nature of the FFT coefficients will be useful to embed the watermark either in magnitude or phase coefficients.

B. Quantization Index Modulation

Quantization Index Modulation (QIM) [12] is one of the watermark embedding methods to achieve blind watermarking. This method provides better rate-distortion-robustness trade-offs than previous methods [8].

C. Synchronization Code

The watermark regions will be dislocated due to de-synchronization attacks. Desynchronization attack means the watermark cannot be detected from the watermarked audio because of lack of synchronization. Such attacks are cropping, shifting and MP3 compression, they will change the audio signal length, which leads to unsuccessful extraction of the watermark. To overcome this problem, the watermark's actual position must be recognized before its extraction. A logistic chaotic sequence is used to generate synchronization code. The logistic chaotic sequence is given below:

$$z_{n+1} = \gamma\, z_n (1 - z_n) \tag{2}$$

where z_n is the initial value that is in between 0 and 1, γ is the real parameter.
Synchronization code is generated using eq (2) based on the following condition.

$$S_n = \begin{cases} 1, & \text{if } z_n > 1/2 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

III. PROPOSED METHOD

In this scheme, the entire audio stream is segmented and each audio segment is divided into two parts. The synchronization code is inserted in the first part to withstand de-synchronization attacks. The watermark is pre-processed and is inserted into the remaining portion of the host signal using FFT and QIM techniques. Fig.1 shows the block diagram of the proposed embedding algorithm.

A. Watermark Pre-processing

To improve the security and robustness, the watermark should be pre-processed before embedding. The watermark is a binary image and is pre-processed by Gauss Map chaotic encryption technique which is defined as:

$$y_{n+1} = e^{(-\alpha(y_n)^2)} + \beta \tag{4}$$

where y_1 is the initial value that lies in between 0 and 1. α and β are the real parameters. Then

$$Z_n = \begin{cases} 1, & \text{if } y_n > T \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where T is the predefined threshold. The watermark in matrix format is converted into a vector w_n with length N X N. This vector w_n is encrypted by Z_n with the following condition:

$$G_n = \mathrm{XOR}(Z_{n,} w_n) \tag{6}$$

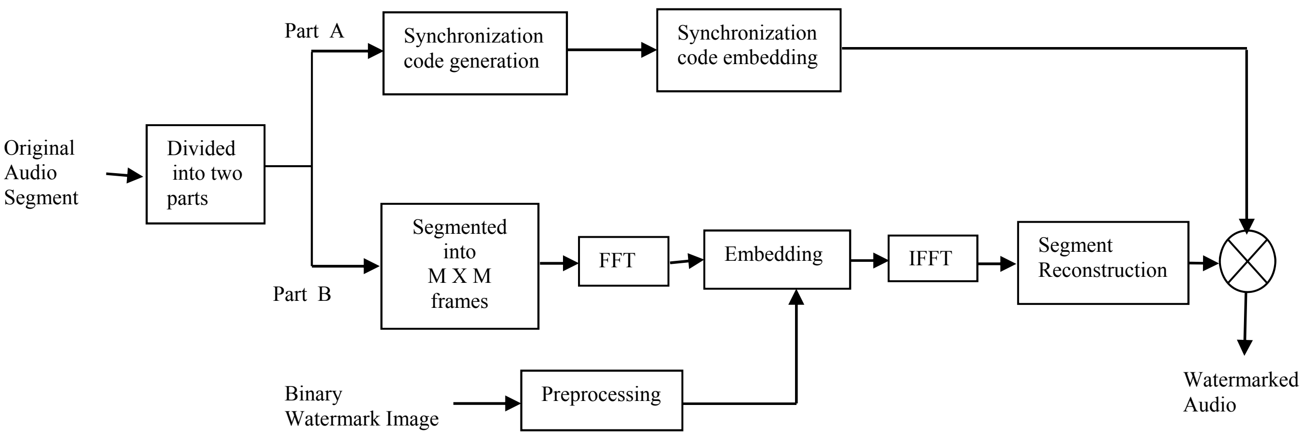

Fig.1 Block diagram of Embedding process

B. Embedding Algorithm

QIM method is used to embed both synchronization code and binary watermark. QIM offers good robustness and its blind in nature makes it very popular technique [8].

The cover audio A is divided into segments and each segment is partitioned into two parts A_S and A_w. Synchronization code that is generated from the eq (3) is inserted into the initial part of the audio segment A_S with length L_S is embedded as shown in Fig.2 and using Eq. 7.

$$A'_S(n) = \begin{cases} \mathrm{round}\left(\frac{A_S(n)}{\delta}\right) * \delta, & \text{if } S_n = 0 \\ (\mathrm{floor}(\frac{A_S(n)}{\delta}) * \delta) + \frac{\delta}{2}, & \text{if } S_n = 1 \end{cases} \tag{7}$$

where δ is the embedding strength.

Segment 1		Segment 2		…………….	Segment N	
Sync Code	Watermark bits	Sync Code	Watermark bits	…………….	Sync Code	Watermark bits

Fig.2 Embedding Process of Synchronization Code

The embedding process of watermark in the second part of the audio is as follows:
Step1: The second part of the audio A_w is segmented into frames. Number of frames in an audio depend upon the size of the watermark i.e., N X N.

$$A_w = A_1, A_2, A_3, \ldots.. A_{(NXN)}$$

Step2: Perform FFT on each audio frame A_i.

$$F_i = FFT(A_i)$$

Step3: The binary watermark image W is chaotically encrypted using Gauss map and is represented as G.
Step4: The encrypted watermark image is embedded in the following way.

$$F'_w(n) = \begin{cases} \text{round}\left(\frac{F_i(n)}{Q}\right) * Q, & \text{if } G_n = 0 \\ (\text{floor}(\frac{F_i(n)}{Q}) * Q) + \frac{Q}{2}, & \text{if } G_n = 1 \end{cases} \quad (8)$$

where Q is the embedding strength.
Step5: Apply inverse FFT to the modified coefficients of each frame.
Step6: Build the modified audio sequence from the frames.

C. Synchronization Code Detection

The de-synchronization attacks will relocate the watermark and disturb the synchronization code; hence the watermark is extracted upon detecting the synchronization code.
The entire embedded and attacked watermarked audio signal A''_S searches for synchronization code, it will be detected with the condition below.

$$S'_n = \begin{cases} 0, & \text{if } \delta/4 \le \text{mod}(A''_S(n), \delta) < 3\delta/4 \\ 1, & \text{otherwise} \end{cases} \quad (9)$$

The similarity between the extracted synchronization code and original synchronization code is evaluated by the eq.10.

$$NC = \frac{\sum_{k=1}^{n} S_n(k) S'_n(k)}{\sqrt{\sum_{k=1}^{n} S_n(k)^2} \sqrt{\sum_{k=1}^{n} S'_n(k)^2}} \quad (10)$$

The similarity coefficient should be more than a predefined threshold for better watermark extraction. The threshold is fixed by obtaining the response of the synchronization code detector and is shown in Fig.3. Here, based on the response the threshold is fixed at 0.75. If the similarity coefficient is less than the threshold, the search for synchronization code happens from the next sample of the audio.

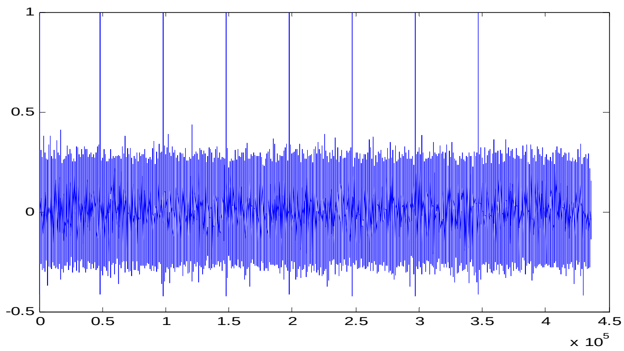

Fig.3 Synchronization Code Detector Response

D. Watermark Extraction Algorithm

The process of watermark extraction is detailed below.
Step1: Second part of attacked watermarked audio signal is segmented into frames same as embedding process.
Step2: Perform FFT on each frame.

Step3: Binary encrypted watermark vector is extracted from the following equation.

$$g'_n = \begin{cases} 0, & \text{if } Q/4 \leq \mathrm{mod}(F''_w(n), Q) < 3Q/4 \\ 1, & \text{otherwise} \end{cases} \quad (11)$$

Step4: The decryption process is same as encryption to determine the binary watermark sequence.

Step5: Finally convert the one dimensional extracted and decrypted binary sequence into two dimensional watermark image of size N X N.

IV. Experimental Results

In this section, the performance of the blind audio watermarking based on FFT is evaluated using speech, pop music, rock music and jazz instrumental audio. The cover audio signals considered are mono files in WAV format. The sampling and quantization rate is 44.1 kHz and 16 bits per sample respectively. An original jazz instrumental audio and its embedded audio signals are shown in Fig. 4. A 128 X 128 binary watermark image and its encrypted image are shown in Fig. 5. Thus the watermark sequence and number of non-overlapping audio frames are 16384.

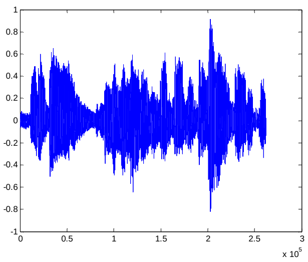

Fig. 4(a). Original Audio Signal

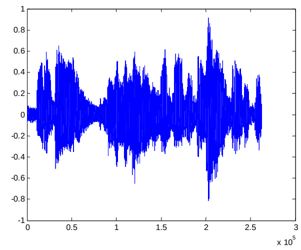

Fig.4(b). Watermarked audio signal

Fig.5(a). Watermark Image

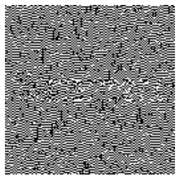

Fig. 5(b). Encrypted Watermark

In this experiment, the parameters x_n=0.3, γ=3.8 are used for the synchronization code generation that is based on logistic chaotic sequence. Parameters $y_{n=0.4}$, α=5.9, β=-0.39, T=0.25 are used for watermark pre-processing. These parameters act as secret key and should be in the specified range in order to have better imperceptibility and robustness. The parameter δ=0.01 is synchronization insertion strength. The variation in the SNR of the watermarked audio signal for different values of δ is explored and shown in the Fig. 6 (a). As δ increases SNR decreases. The parameter Q=0.02 is used as a quantization strength in the watermark embedding process. The SNR of the watermarked audio signal decreases with increasing Q and is shown in Fig. 6 (b).

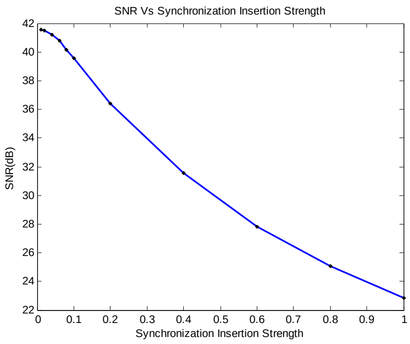

Fig. 6(a) SNR Vs Synchronization Insertion Strength

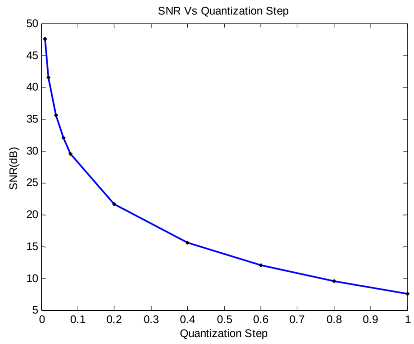

Fig. 6(b) SNR Vs Quantization Step

Audio watermarking algorithm performance is generally assessed with respect to two common performance metrics, i.e., imperceptibility and robustness [15].

A. *Imperceptibility (Inaudibility)*

Imperceptibility is the perceptual quality measure of the watermarked audio signal. Imperceptibility measures are two types: subjective measure and objective measure [16].

In order to calculate the quality of the watermarked signal in terms of objective means, the following equation is used:

$$\mathrm{SNR} = 10\log_{10}\frac{\sum_{i=1}^{L} y^2(n)}{\sum_{i=1}^{L}[y(n)-y'(n)]^2} \tag{12}$$

where $y(n)$ and $y'(n)$ are the original and embedded audio signal, respectively. The SNR values of all selected audio signals after embedding using the proposed scheme with a quantization strength of Q=0.02 are above 20dB (according to the According to the IFPI) are shown in Table I.

TABLE I.

SNR VALUES FOR DIFFERENT AUDIO SIGNALS

Type of an audio signal	**SNR**
POP Music	41.5689
ROCK Music	36.6898
JAZZ instrumental audio	36.9295
Speech signal	45.5995

SNR is a common means to measure the audio quality. Perceptual Audio Quality Measure (PAQM) can be used to measure the quality because it takes into account the HAS characteristics.

Subjective test is a listening test; it was performed with ten subjects to estimate the Mean Opinion Score (MOS) for all the four embedded audio signals [17]. The listeners report the difference between the original audio signal and the embedded signal after listening to the pair of signal five times. The 5-point Mean Opinion Score (MOS) criteria are listed in Table II and Table III represents the values of MOS for the proposed method.

TABLE II.

MOS CRITERION

Score	**Watermark imperceptibility**
5	Imperceptibility
4	Perceptibility but not annoying
3	Slightly annoying
2	Annoying
1	Very annoying

TABLE III.

MOS VALUES

Class of audio signal	MOS
Speech	5
Pop	5
Rock	4.8
Jazz instrument	4.7

B. *Robustness Test*

The performance of the algorithm can be evaluated by using robustness tests like BER, Precision and Correlation Coefficient.

The Bit Error Rate (BER) provides the measure of how accurately the watermark is detected after post-processing attacks [7], [18].

$$\text{BER} = \frac{\text{Number of error bits}}{\text{Number of total bits}} \tag{13}$$

Precision [19] also one of the robustness measure. It gives the percentage of watermark bits that are correctly decoded after the post-attack extraction, and is as given below.

$$\text{Precision} = \frac{L - \sum_{i=1}^{L} |W(i) - W'(i)|}{L} \tag{14}$$

where W is the original binary watermark sequence, W' is the extracted binary watermark and L is the length of the watermark sequence.

Normally correlation coefficient provides a measure of the quality of the image. The correlation coefficient is given by:

$$\gamma = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{\sum_m \sum_n (A_{mn} - \bar{A})^2 \sum_m \sum_n (B_{mn} - \bar{B})^2}} \tag{15}$$

Where γ = correlation coefficient

A= extracted image

B = original image

$\bar{A}$ and $\bar{B}$ are the means of A and B respectively.

The capacity or payload is the number of bits that are inserted into the host signal within a unit of time and is measured in terms of bits per second.

$$\text{Payload} = \frac{N}{T} \tag{16}$$

where N is the number of watermark bits, T is the duration of the host audio.
Here N is 16384 and T is 10sec. So, payload in this experiment is 1638 bps.

To evaluate the robustness of the algorithm, signal processing attacks are performed and are listed below.

i) Resampling: The embedded audio signal sampled at 44.1 kHz is resampled at fs/2 i.e., 22.05 kHz, fs/4 i.e., 11.025 kHz, 8 kHz and restored back to 44.1 kHz.

ii) Noise: A random noise of 40 dB distorted the embedded signal.

iii) AWGN: A 60 dB additive white Gaussian noise is added to the watermarked audio signal.

iv) Low-pass filtering: Butterworth filter of second order with cut-off frequency 16 KHz is used.

v) Jittering: Jittering is a small rapid variation. One sample out of every 100,000, 50,000 and 10,000 samples is removed in our experiment.

vi) Echo addition: 0.1% decay and 400 ms delayed audio is added to the watermarked audio signal.

vii) Signal addition: 2000 samples of the original audio signal are added to the beginning of the corresponding samples of the watermarked audio signal.

viii) Signal subtraction: 2000 samples of the original audio signal are subtracted from the beginning of the corresponding samples of the watermarked audio signal.

ix) Cropping: 1000 samples are removed from the beginning, middle and end parts of the watermarked audio signal and then these samples are replaced with 0.

x) MP3 Compression: The watermarked audio signal is compressed using MP3 Compression at the bit rate of 256Kbps, 160 Kbps and 128 Kbps and then back to the .WAV format.

xi) TSM: Time Scale Modification processing is done in the watermark audio signal to change the time scale to $\pm$1%, $\pm$2%, while preserving the pitch.

The extracted watermarks of the four classes of signal for various attacks are shown in Fig. 7-10.

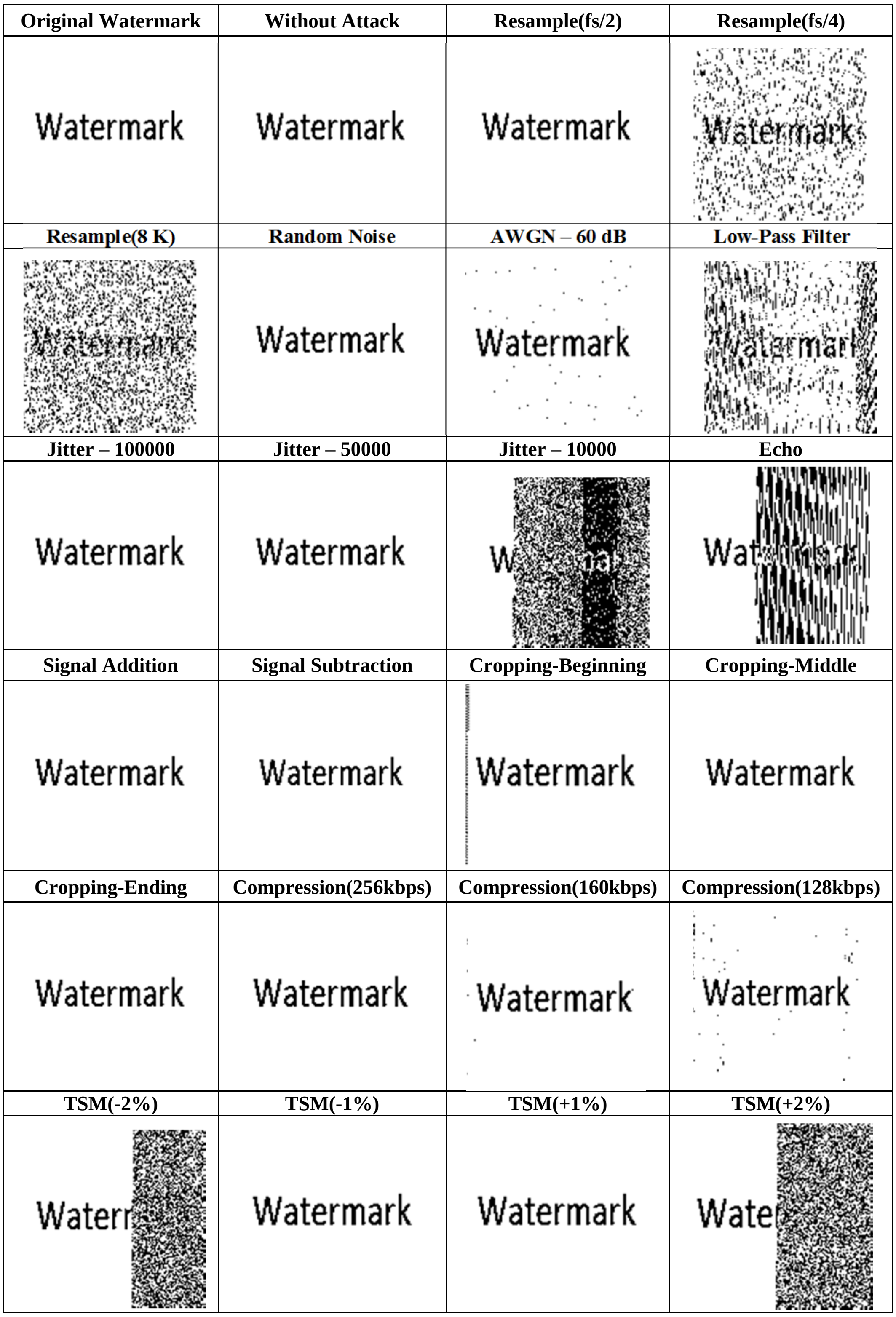

Fig. 7. Extracted Watermarks for a pop music signal

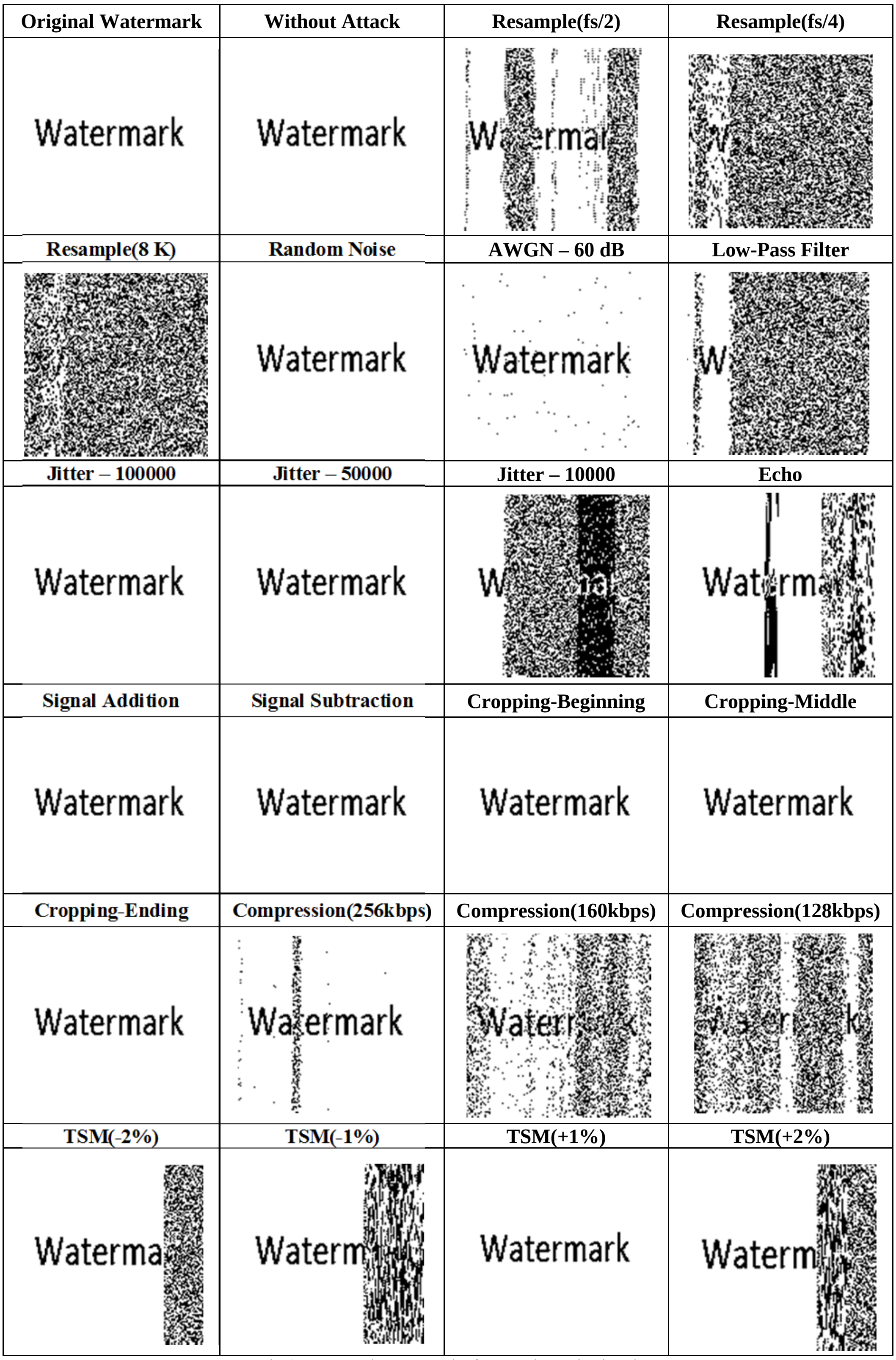

Fig.8. Extracted Watermarks for a rock music signal

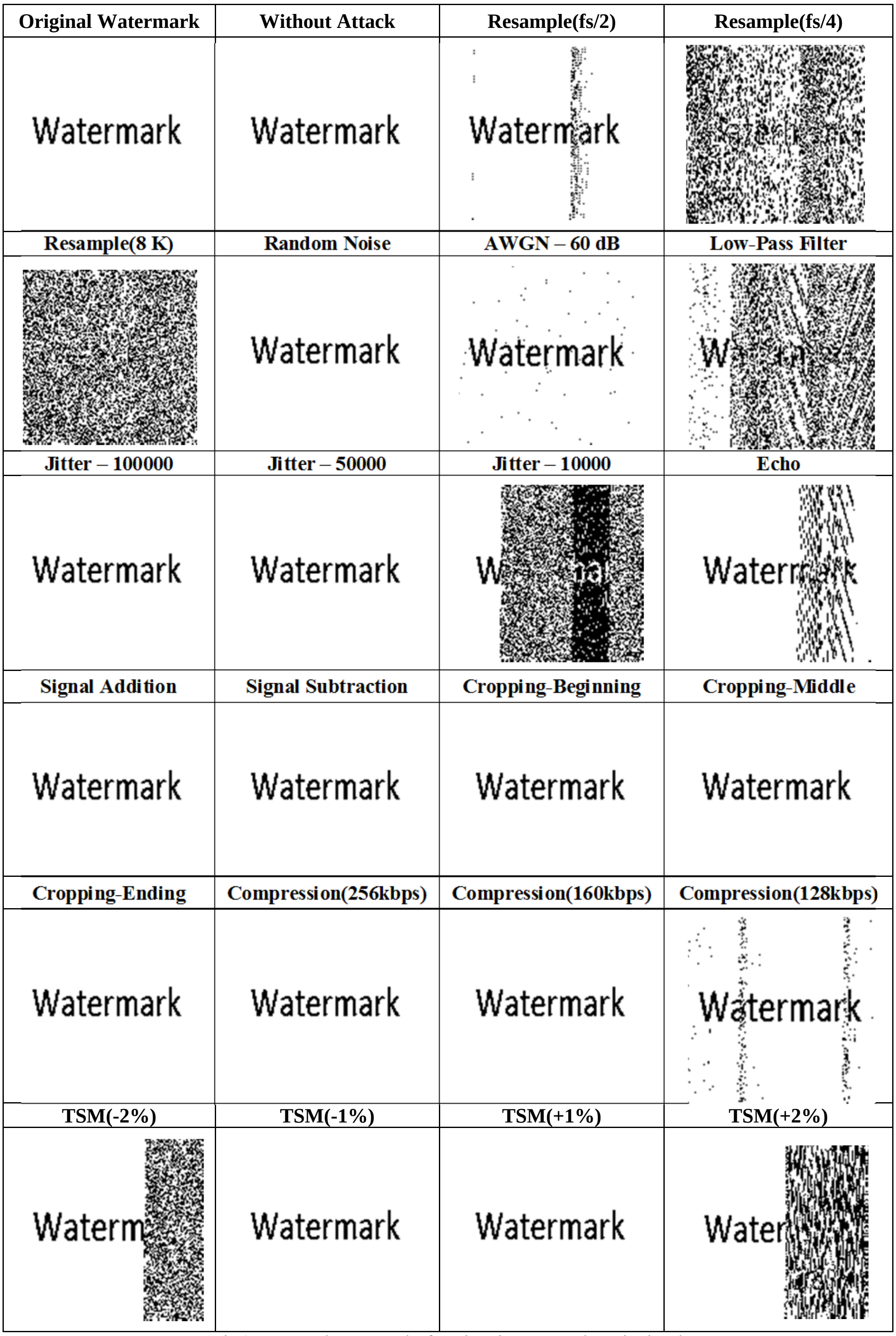

Fig.9. Extracted Watermarks for a jazz instrumental music signal

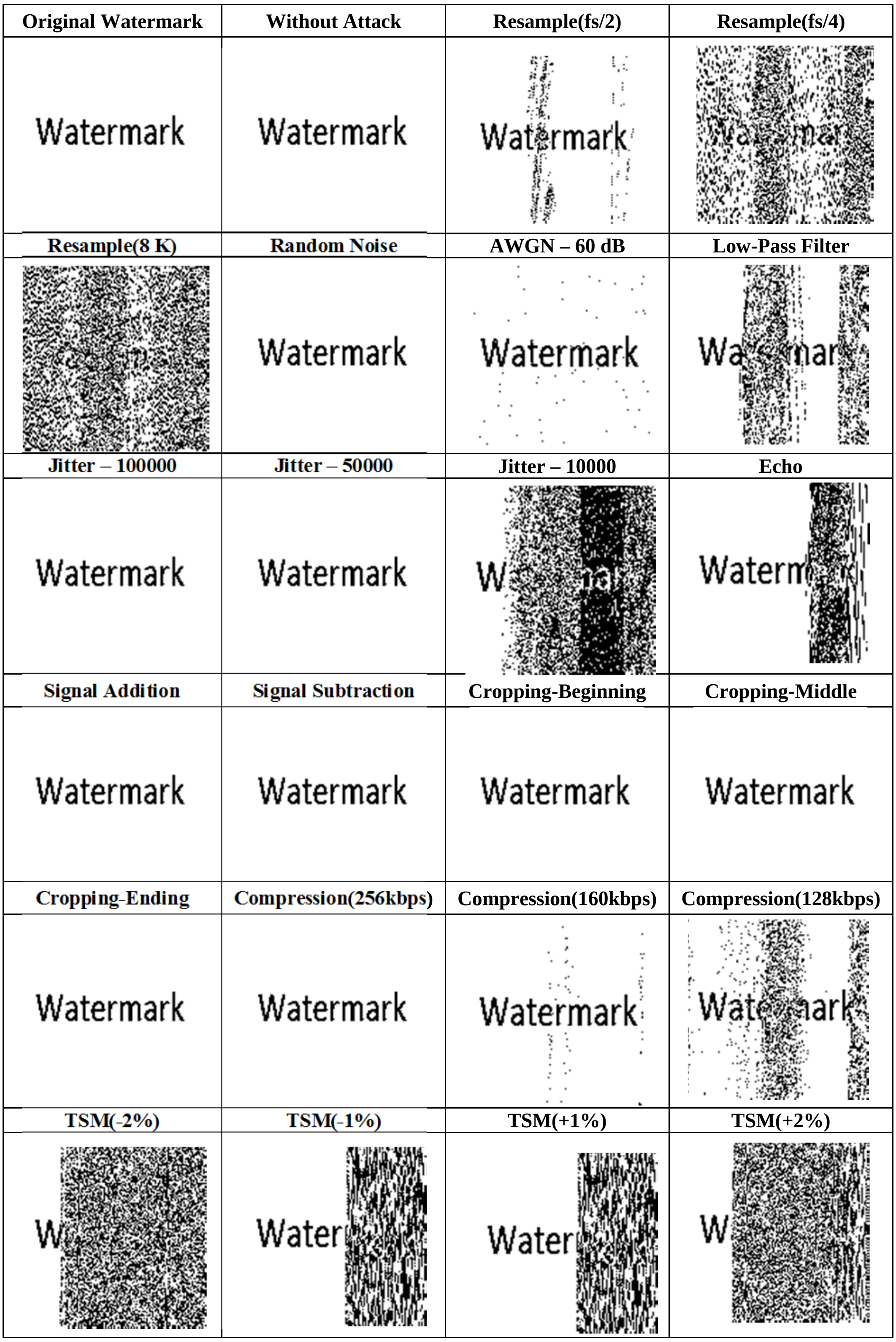

Fig. 10. Extracted Watermarks for a speech signal

The performance metrics, i.e., BER, Precision and Correlation coefficient for the above said attacks are computed for the four classes of audio signals (pop, rock, jazz and speech) are summarized in Table IV.

TABLE IV.
BER, Precision, CC for the four classes of audio signals

Type of Audio	Type of attack	BER	Precision	CC
POP	Resample (fs/2)	0	1	1
	Resample (fs/4)	0.1165	0.8835	0.4370
	Resample (8kHz)	0.2360	0.7640	0.2395
	Random Noise (40dB)	0	1	1
	AWGN(60dB)	0.0037	0.9963	0.9595
	Lowpass filter (16k)	0.1827	0.8173	0.3141
	Jittering (100000)	0	1	1
	Jittering (50000)	0	1	1
	Jittering (10000)	0.4824	0.5176	0.0152
	Echo	0.2914	0.7086	0.1771
ROCK	Resample (fs/2)	0.1744	0.8256	0.3352
	Resample (fs/4)	0.4514	0.5486	0.0400
	Resample(8kHz)	0.4775	0.5225	0.0280
	Random Noise(40dB)	0	1	1
	Lowpass filter(16k)	0.3870	0.6130	0.0951
	Jittering(100000)	0	1	1
	Jittering(50000)	0	1	1
	Jittering(10000)	0.4799	0.5201	0.0258
	Echo	0.1445	0.8555	0.3811
	AWGN(60dB)	0.0045	0.9955	0.9511
JAZZ	Resample (fs/2)	0.0204	0.9796	0.8145
	Resample (fs/4)	0.3531	0.6469	0.1197
	Resample (8kHz)	0.4656	0.5344	0.0145
	Random Noise (40dB)	0	1	1
	AWGN (60dB)	0.0037	0.9963	0.9592
	Lowpass filter (16k)	0.2980	0.7020	0.1857
	Jittering (100000)	0	1	1
	Jittering (50000)	0	1	1
	Jittering (10000)	0.4756	0.5244	0.0272
	Echo	0.0828	0.9172	0.5231
SPEECH	Resample (fs/2)	0.0302	0.9698	0.7532
	Resample (fs/4)	0.3013	0.6987	0.1881
	Resample (8kHz)	0.4166	0.5834	0.0874
	Random Noise(40dB)	0	1	1
	AWGN(60dB)	0.0038	0.9962	0.9582
	Lowpass filter(16k)	0.1660	0.8340	0.3534
	Jittering(100000)	0	1	1
	Jittering(50000)	0	1	1
	Jittering(10000)	0.4705	0.5295	0.0444
	Echo	0.1697	0.8303	0.3488

In the proposed method, the insertion of synchronization code improves the detection of watermarks. The synchronization code is inserted and the watermark is embedded in the cover audio. The robustness is evaluated by extracting the watermark from the watermarked audio which is disturbed by de-synchronization attacks. A similar procedure is adopted to measure the robustness parameters without synchronization code. The improvement of robustness with the insertion of synchronization code is shown in Table V.

TABLE V.
Robustness for Desynchronization Attacks

Type of Audio	Type of attack	Without Synchronization			With Synchronization		
		BER	**Precision**	**CC**	**BER**	**Precisio n**	**CC**
POP	Signal addition	0.0199	0.9801	0.8248	0	1	1
	Signal subtraction	0.0199	0.9801	0.8248	0	1	1
	Start cropping	0.0088	0.9912	0.9109	0	1	1
	Middle cropping	0	1	1	0	1	1
	End cropping	0	1	1	0	1	1
	MP3 Compression(256)	0	1	1	0	1	1
	MP3 Compression(160)	0.0005	0.9995	0.9937	0.0005	0.9995	0.9937

	MP3 Compression(128)	0.0045	0.9955	0.9517	0.0045	0.9955	0.9517
	TSM_99	0.4935	0.5065	0.0028	0	1	1
	TSM_98	0.4826	0.5174	0.0231	0.2039	0.7961	0.3052
	TSM_101	0.4993	0.5007	0.0002	0	1	1
	TSM_102	0.4990	0.5010	-0.0006	0.2516	0.7484	0.2326
ROCK	Signal addition	0.0192	0.9808	0.8298	0	1	1
	Signal subtraction	0.0192	0.9808	0.8298	0	1	1
	Start cropping	0.0088	0.9912	0.9109	0	1	1
	Middle cropping	0	1	1	0	1	1
	End cropping	0	1	1	0	1	1
	MP3 Compression(256)	0.0427	0.9573	0.6882	0.0216	0.9784	0.8063
	MP3 Compression(160)	0.2026	0.7974	0.3030	0.2075	0.7925	0.2925
	MP3 Compression(128)	0.3123	0.6877	0.1748	0.3307	0.6693	0.1663
	TSM_99	0.5091	0.4909	0.0078	0.1598	0.8402	0.3723
	TSM_98	0.4984	0.5016	0.0162	0.1083	0.8917	0.4799
	TSM_101	0.4979	0.5021	0.0105	0	1	1
	TSM_102	0.5027	0.4973	-0.0113	0.1572	0.8428	0.3686
JAZZ	Signal addition	0.0180	0.9820	0.8379	0	1	1
	Signal subtraction	0.0180	0.9820	0.8379	0	1	1
	Start cropping	0.0089	0.9911	0.9104	0	1	1
	Middle cropping	0	1	1	0	1	1
	End cropping	0	1	1	0	1	1
	MP3 Compression(256)	0.0002	0.9998	0.9972	0	1	1
	MP3 Compression(160)	0.0277	0.9723	0.7671	0.0277	0.9723	0.7671
	MP3 Compression(128)	0.0713	0.9287	0.5741	0.0190	0.9810	0.8260
	TSM_99	0.4993	0.5007	-0.0050	0	1	1
	TSM_98	0.4836	0.5164	-0.0057	0.1588	0.8412	0.3671
	TSM_101	0.4978	0.5022	0.0004	0	1	1
	TSM_102	0.4930	0.5070	0.0025	0.2358	0.7642	0.2625
SPEECH	Signal addition	0.0173	0.9827	0.8427	0	1	1
	Signal subtraction	0.0174	0.9826	0.8422	0	1	1
	Start cropping	0.0089	0.9911	0.9104	0	1	1
	Middle cropping	0	1	1	0	1	1
	End cropping	0	1	1	0	1	1
	MP3 Compression(256)	0.0510	0.9490	0.6464	0.0044	0.9956	0.9523
	MP3 Compression(160)	0.1592	0.8408	0.3596	0.1321	0.8679	0.4141
	MP3 Compression(128)	0.1919	0.8081	0.3149	0.1919	0.8081	0.3149
	TSM_99	0.4875	0.5125	0.0166	0.2238	0.7762	0.2732
	TSM_98	0.4870	0.5130	0.0115	0.4008	0.5992	0.1013
	TSM_101	0.4929	0.5071	0.0149	0.4619	0.5381	0.0129
	TSM_102	0.4767	0.5233	0.0144	0.3671	0.6329	0.1071

Table VI shows the comparison of the performance of various audio watermarking techniques. It indicates that the proposed algorithm obtained a relatively high payload with respect to [7], [19], [13], [6], [18], [3] except [14]. The algorithm is capable of achieving moderately high SNR indicates better imperceptibility even at high payload.

TABLE VI.
SUMMARY OF ALGORITHM COMPARISON

Reference	**Method**	**Payload (bps)**	**Blind**	**SNR(dB)**	**Secret key used**	**Synchronization**	**Subjective test**
Bhat et al [7]	DWT-SVD	45.9	Yes	24.37	Yes	Yes	Yes
Wang et al [19]	FFT-RSVD	187	Yes	27.23	No	No	Yes
Lei et al [13]	SVD-DCT	43	Yes	32.53	Yes	Yes	Yes
B.Lei at al [6]	LWT-SVD	170.67	Yes	40	Yes	Yes	Yes
Khaldi et al [18]	EMD	50.3	Yes	26.38	No	Yes	Yes
P K Dhar et al [3]	CPT-SVD	689.56	Yes	36.86	Yes	No	Yes
Mehdi Fallahpour et al [14]	Log-FFT	7000	Yes	36	Yes	No	Yes
Ours	**FFT-QIM**	**1638**	**Yes**	**45.598**	**Yes**	**Yes**	**Yes**

V. CONCLUSION

The proposed scheme is a blind audio watermarking in frequency domain based on QIM. The choice of FFT makes the scheme to work with less computational effort. This algorithm uses the synchronization code to make the scheme to withstand de-synchronization attacks like signal addition, signal subtraction, cropping, MP3 compression and time-scale modification. The subjective and objective analysis demonstrates that this watermarking scheme offers better robustness against most of the attacks like resampling, cropping, random noise, additive noise, low-pass filtering, jittering and echo addition. The Gauss map chaotic sequence scrambles the watermark image in the host audio which increases the robustness as well as security. The performance of the algorithm is compared with other state-of-art algorithms indicates that the proposed method is superior in terms of imperceptibility, robustness and payload.

REFERENCES

[1] M.Arnold, "Audio Watermarking: Features, Applications and Algorithms", *in Proc, IEEE Int. Conf. Multimedia and Expo.*, New York, 2000, vol. 2, pp. 1013-1016.

[2] Katzenbesisser, S., Petitcolas, F.A.P. "Information hiding techniques for steganography and digital watermarking. Artech. House, Boston, 2000.

[3] P.K.Dhar, T. Shimamura, "Audio watermarking in transform domain based on singular value decomposition and Cartesian-polar transformation", *International Journal of Speech Technology*, Vol.17, pp-133-144, June 2014.

[4] P.Bassia, I. Pitas, N. Nikolaidis, "Robust audio watermarking in the time domain", *IEEE Transactions on Multimedia,* vol.3, No.2, pp. 232-241, 2001.

[5] S.Wu, J.Huang, D.Huang, Y.Q.Shi, "Efficiently self_synchronized audio watermarking for assured audio data transmission", *IEEE Trans. Broadcasting,* Vol. 51, No.1, pp 69-76, 2005.

[6] B.Lei, I.Y.Soon, F.Zhou, Z.Li, H.Lei, "A robust audio watermarking scheme based on lifting wavelet transform and singular value decomposition", *Signal Processing,vol.* 92, pp.1985-2001, 2012.

[7] V.Bhat K, I. Sengupta, A. Das, "An adaptive audio watermarking based on the singular value decomposition in the wavelet domain", *Digital Signal Processing*, Vol. 20, pp.1547-1558, 2010a.

[8] B.Chen and G.Wornell, "Quantization Index Modulation: A class of provably good methods for digital watermarking and information embedding*", IEEE Trans. Inform. Theory*, vol. 47, pp. 1423-1443, 2001.

[9] P.Moulin and R. Koetter, "Data-hiding codes*", In Proc. IEEE*, 2005, vol. 93, no. 12, pp. 2083-2126.

[10] X.Y.Wang, H.Zhao, "A novel synchronization invariant audio watermarking scheme based on DWT and DCT*", IEEE Trans. Signal Process*., Vol.54, No.12, pp. 4835-4840, 2006.

[11] Megias D., Sreea-Ruiz J., Fallahpour M., "Efficient self-synchronized blind audio watermarking system based on time domain and FFT amplitude modification", *Signal Processing*, vol. 90, No.12, pp.3078-3092, 2010.

[12] V.Bhat K, I. Sengupta, A. Das, "An audio watermarking scheme using singular value decomposition and dither-modulation quantization", *Multimedia Tools and Applications*, Vol.52, pp.369-383, 2010b.

[13] B.Y.Lei, I.Y.Soon, Z.Li, "Blind and robust audio watermarking scheme based on SVD-DCT", *Signal Processing, vol.* 91, (8) pp. 1973-1984, 2011.

[14] M. Fallahpour, David.M, " Secure logarithmic audio watermarking scheme based on the human auditory system", *Multimedia Systems,* vol. 20, pp. 155–164, 2014.

[15] Acevedo A., "Digital Watermarking for Audio Data in Techniques and Applications of Digital Watermarking and Content Protection", Artech House, USA, 2003.

[16] Christian N, Jurgen H, "Digital watermarking and its influence on audio quality*", In: Proceeding of the 105th AES convention*, 1998, Preprint 4823.

[17] Beerends J.and Stemerdink J., "A Perceptual Audio Quality Measurement Based on a Psychoacoustic Sound Representation", *Journal of the Audio Engineering Society*, vol. 40, no.12, pp.963-978, 1992.

[18] Khaldi K and Boudraa A O, "Audio Watermarking via EMD," *IEEETransactions on Audio, Speech and Language Processing,* vol. 21, no. 3,pp. 675-680, 2013.

[19] J.Wang, R.Healy, J.Timoney, "A Robust audio watermarking scheme based on reduced singular value Decomposition and distortion removal," *Signal Processing*, vol. 91, no. 8, pp. 1693-1708, 2011.

AUTHOR PROFILE

N. V. Lalitha is presently pursuing PhD at GIT, GITAM University. She obtained her M.Tech from Jawaharlal Nehru Technological University, Kakinada and B.Tech from Jawaharlal Nehru Technological University. Presently, she is working as Assistant professor in the Department of Electronics and Communication Engineering at GMR Institute of Technology, Rajam, Srikakulam District. She is having 8 years of teaching experience. Her research interests are Audio and Image Processing. She is a Life Member of IETE.

Srinivasa Rao Ch is currently working as Professor in the Department of ECE, JNTUK University College of Engineering, Vizianagaram, AP, India. He obtained his PhD in Digital Image Processing area from University College of Engineering, JNTUK, Kakinada, AP, India. He received his M. Tech degree from the same institute. He published 36 research papers in international journals and conferences. His research interests are Digital

Speech/Image and Video Processing, Communication Engineering and Evolutionary Algorithms. He is a Member of CSI. Dr Rao is a Fellow of IETE.

P. V. Y. Jayasree is currently working as Associate Professor in the Department of ECE, GIT, GITAM University. She obtained her PhD from University College of Engineering, JNTUK, Kakinada, AP, India. She received M.E. from Andhra University. She published more than 50 research papers in international journals and conferences. Her research interests are Signal Processing, EMI/EMC, RF & Microwaves.