Hindawi
Security and Communication Networks
Volume 2018, Article ID 1781384, 21 pages
https://doi.org/10.1155/2018/1781384

Research Article

Blind Key Based Attack Resistant Audio Steganography Using Cocktail Party Effect

Barnali Gupta Banik [1] and Samir Kumar Bandyopadhyay [2]

[1] *St. Thomas' College of Engineering & Technology, Kolkata 700023, India*
[2] *University of Calcutta, Kolkata 700098, India*

Correspondence should be addressed to Barnali Gupta Banik; gupta.barnali@gmail.com

Received 19 October 2017; Revised 13 January 2018; Accepted 5 February 2018; Published 16 April 2018

Academic Editor: Emanuele Maiorana

Steganography is a popular technique of digital data security. Among all digital steganography methods, audio steganography is very delicate as human auditory system is highly sensitive to noise; hence small modification in audio can make significant audible impact. In this paper, a key based blind audio steganography method has been proposed which is built on discrete wavelet transform (DWT) as well as discrete cosine transform (DCT) and adheres to Kerckhoff's principle. Here image has been used as secret message which is preprocessed using Arnold's Transform. To make the system more robust and undetectable, a well-known problem of audio analysis has been explored here, known as Cocktail Party Problem, for wrapping stego audio. The robustness of the proposed method has been tested against Steganalysis attacks like noise addition, random cropping, resampling, requantization, pitch shifting, and mp3 compression. The quality of resultant stego audio and retrieved secret image has been measured by various metrics, namely, "peak signal-to-noise ratio"; "correlation coefficient"; "perceptual evaluation of audio quality"; "bit error rate"; and "structural similarity index." The embedding capacity has also been evaluated and, as seen from the comparison result, the proposed method has outperformed other existing DCT-DWT based technique.

1. Introduction

In the present era, communicating through Internet has become vulnerable as there may be several intruders who can eavesdrop for secret messages to capture and disburse them for unlawful misconducts. Henceforth nowadays it is most necessary to camouflage secret message in such a way that stego cannot be identified as carrier of secret message. Camouflaging secret message through carrier objects introduces the age-old technique of steganography. However, with the current enormous use of Internet and elevation of various Steganalysis attacks, it is required to have an extra shield to protect steganography techniques. This is the reason cocktail party effect in audio steganography has been explored to ensure enhanced security during data transmission.

2. Related Work

2.1. Audio Steganography Techniques. In audio steganography, audio is used as cover media. In [1], authors have described different spatial and frequency domain techniques of audio steganography. The popular spatial domain techniques are as follows.

Least Significant Bit (LSB) Encoding. This is the simplest method of audio steganography where Least Significant Bit of each audio sample is modified with bits of secret message vector. With the extensive use of this method it becomes more prone to attack and its embedding capacity is poor compared to others. To cope up with the necessity of increasing capacity, authors of [2] have proposed an enhanced method of LSB technique where it has been proved that 2nd and 3rd LSB modification does not make audible difference in audio sample. In [3], authors have suggested another enhancement over LSB technique by shifting LSB modification from 3rd bit to 4th bit which incur more embedding capacity compared to previous methods of LSB encoding.

Parity Encoding. In this approach, audio signal is broken into number of samples [4]. Depending on sample's parity bit,

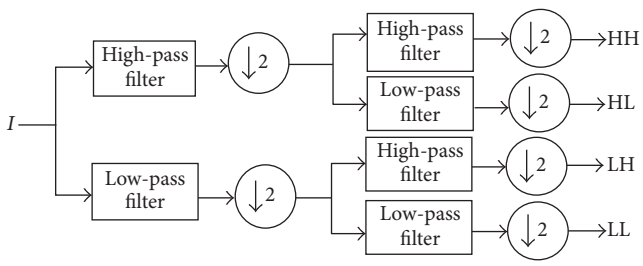


Figure 1: Block diagram of 1 level 2D DWT.

secret message is embedded in the LSB of the sample byte stream.

Echo Hiding. In this method, a short echo signal is introduced as part of cover audio where secret message is hidden [5]. Study shows that the echo signal is inaudible provided the delay between cover audio and echo signal is up to 1 ms.

The widespread frequency domain techniques are as follows.

Phase Coding. As human auditory system cannot percept phase component modulation, hence, in this technique, secret data is embedded by modification of selected phase component of cover audio signal. Using psychoacoustic model, a threshold is calculated which can be used as masking threshold [6]. In [7], authors have used difference between the phase values of the selected component frequencies and their adjacent frequencies of the cover signal as a medium to hide secret data bits. This method provides more robustness than the previous approaches.

Spread Spectrum. The basic principle of spread spectrum is to spread the secret message over the frequency spectrum of cover audio signal. In [8], Direct Sequence Spread Spectrum is used to hide text data in an audio. Here a key is used to embed message to the noise. In [9], authors have discovered that low spreading rate improves performance of spread spectrum audio steganography. Therefore, authors have proposed a technique which decreases correlation between original signal and spread data signal by having phase shift in each subband signal of original audio.

Discrete Wavelet Transforms (DWT). DWT decomposes a signal in four frequency components, popularly known as subbands. These sub bands are Low-Low (LL), Low-High (LH), High-Low (HL), and High-High (HH), as shown in Figure 1. The LL subband describes approximation details. The HL band demonstrates variation along the x-axis or horizontal details and the LH band demonstrates the y-axis variation or vertical details [10]. In other words, the low frequency subband is a low-pass approximation of the original signal and contains most energy of the signal. The other subbands include mainly detailed components which have low energy level. This is the reason LH subband is very popular for data hiding.

In [11], authors have proposed a method to create DWT of cover audio and select higher frequency to embed image data using low bit encoding technique. In [12], authors have decomposed the cover audio signal using Haar DWT and then choose coefficient to embed data. This is done using a precalculated threshold value to flip data. In [13], secret audio is embedded using synchronizing code in the low frequency part of DWT of cover audio.

Discrete Cosine Transforms (DCT). DCT is used to convert a signal from spatial domain into frequency domain. DCT decomposes a signal into a series of cosine functions. The two-dimensional DCT can be performed by executing one-dimensional DCT twice, initially in the x direction, next by y direction. The formulation of the 2D DCT for an input signal S with i rows and j columns and the output signal T has been given in

$$T_{x,y} = a_x a_y \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} S_{ij} \cos \frac{\pi (2i+1) x}{2M} \cos \frac{\pi (2j+1) y}{2N}, \quad (1)$$

where $0 \le x \le M-1$ and $0 \le y \le N-1$ and

$$a_x = \begin{cases} \dfrac{1}{\sqrt{M}}, & \text{where } x = 0 \\ \sqrt{\dfrac{2}{M}}, & \text{where } 1 \le x \le M-1, \end{cases}$$
$$a_y = \begin{cases} \dfrac{1}{\sqrt{N}}, & \text{where } y = 0 \\ \sqrt{\dfrac{2}{N}}, & \text{where } 1 \le y \le N-1. \end{cases} \quad (2)$$

Inverse 2D DCT is also available to transform a frequency domain coefficient to spatial domain signal, as specified in

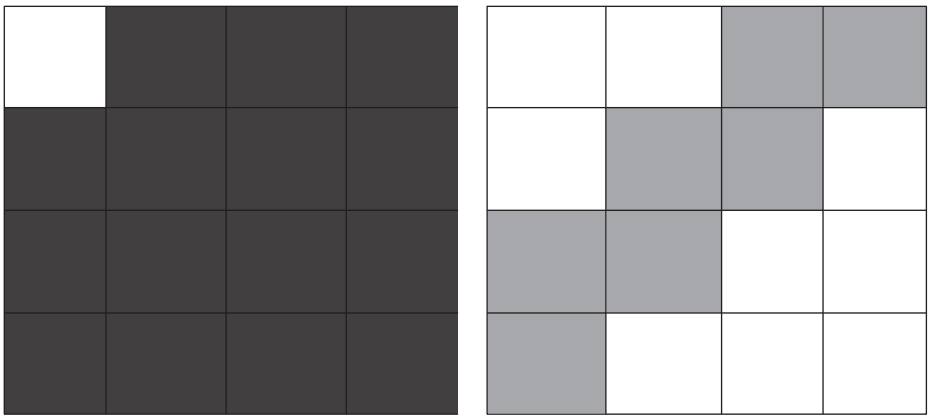
$$S_{ij} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} a_x a_y T_{xy} \cos \frac{\pi (2i+1) x}{2M} \cos \frac{\pi (2j+1) y}{2N}, \quad (3)$$

where $0 \le i \le M-1$ and $0 \le j \le N-1$.

DCT can be performed in block-by-block basis like 4×4, 8×8, and 16×16 blocks.

As shown in Figure 2(a), the top left coefficient is called DC coefficient holding the approximate value of the whole signal; normally it has coefficients with zero frequency and the remaining 15 coefficients are called AC coefficients holding most detailed parameters of the signal, having coefficients with nonzero frequency. There are some DCT coefficients which hold quite similar values. Human brains are less sensitive to detect changes where all the elements hold more or less the same value. Therefore, this region of similar values can be selected for data hiding purpose. This region is known as midband region, as shown in Figure 2(b).

In [14], authors have used speech signal as cover, where voiced and nonvoiced part of the speech are separated by zero crossing count and short time energy. The secret data is embedded by modifying DCT coefficient of nonvoiced part.



(a) DC and AC coefficients in 4 × 4 block (b) Midband region of 4 × 4 block

Figure 2

In [15], authors have decomposed the cover audio in 8 × 8 nonoverlapping block and secret data is hidden in the DC coefficient and 4th AC coefficient in line. In [16], authors have embedded secret data in the low frequency component of DCT quantization. In [17], authors have decomposed the cover audio into 8×8 block and then each of those blocks was decomposed further into 4 × 4 frames. Embedding of secret message depends on the difference between first or last two frames.

2.2. Correlation Coefficient (CC). A correlation coefficient is a measure of linear relationship between two random variables. This term was first coined by Karl Pearson in 1896. The value of correlation coefficient can vary from −1 to 1. If the value is perfect −1 or 1 that indicates both variables are linearly related. If the value is 0 that indicates there is no relation between the said variables. Moreover, the sign indicates that the variables are positively related or negatively related [18]. There are three types of correlation coefficients: Pearson's coefficient (r), Spearman's rho coefficient (r_s), and Kendall's tau coefficient (τ). Pearson's coefficient, which is also known as product-moment correlation coefficient, is the most widely used popular correlation coefficient. It is given by paired measurements $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$ as mentioned in

$$Corr_p = \frac{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{\sqrt{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2 \sum_{i=1}^{n}\left(Y_i - \overline{Y}\right)^2}}, \tag{4}$$

where $\overline{X}$ and $\overline{Y}$ are the mean of $(X_1, X_2, \ldots, X_n)$ and $(Y_1, Y_2, \ldots, Y_n)$, respectively. Correlation coefficient can also be used as quality metrics to measure similarity between two signals.

2.3. Arnold Transform. Arnold's Transform is a chaotic bidirectional map proposed by Vladimir Arnold in 1960. A chaotic map is an evaluation function which demonstrates some sort of chaotic nature, as seen in the following transformation function:

$$\begin{aligned} \Gamma &: \mathbb{T}^2 \longrightarrow \mathbb{T}^2 \text{ given by,} \\ \Gamma &: (a, b) \longrightarrow (2a + b, a + b) \bmod 1. \end{aligned} \tag{5}$$

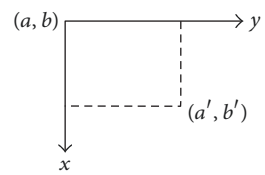


Figure 3: Representation of point (a, b) sheared to point (a', b').

An image is collection of pixels in row and column arrangement, which can be organized in square or nonsquare shape. If Arnold transform is applied to an image, it scrambles the image by "N" times iteration (e.g., iteration 1 will scramble less and iteration 10 will scramble more), which makes the image imperceptible. This undetectable image format can be used for data hiding securely as it is unable to reveal any existence of secret data. Hence scrambling an image can be a preprocessing step of data hiding technique.

Traditionally Arnold transform can be applied only for square matrices; however later it has been improvised to apply on any matrix, by

$$\begin{pmatrix} a' \\ b' \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} \bmod M, \quad \text{where } a, b \in \{0, 1, 2, \ldots, M-1\}, \tag{6}$$

where (a, b) is the element of original matrix and (a', b') is the element of transformed matrix and M is the order of the matrix; as shown in Figure 3, the point (a, b) is sheared through x- and y-axis to get (a', b').

The function $\bmod M$ is important to regenerate the original $M \times M$ image. The functions to shear in x-axis, y-axis, and modulo function is represented in

$$\begin{bmatrix} a \\ b \end{bmatrix} \longrightarrow \begin{bmatrix} a+b \\ b \end{bmatrix}$$

(a) Function to shear in x axis

$$\begin{bmatrix} a \\ b \end{bmatrix} \longrightarrow \begin{bmatrix} a \\ a+b \end{bmatrix} \tag{7}$$

(b) Function to shear in y axis

$$\begin{bmatrix} a \\ b \end{bmatrix} \longrightarrow \begin{bmatrix} a \\ b \end{bmatrix}$$

(c) Modulo function.

Arnold transformation is reversible [19]. To recover original image from scrambled image there are two ways, the traditional way is periodicity, and the better approach is to use inverse matrix, which is also known as Reverse Arnold Transformation [20] and expressed by

$$\begin{pmatrix} a' \\ b' \end{pmatrix} = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} \bmod M. \tag{8}$$

In [21], authors have used Arnold's transformation to scramble the image before embedding into the DWT coefficient of cover audio. In [22], authors have embedded scrambled image in "Redundant Discrete Wavelet Transform" coefficient using Singular Value Decomposition (SVD) technique. In [23], authors have proposed data hiding in DWT and DCT domain using SVD where the secret image is scrambled before embedding.

2.4. Cocktail Party Problem. Cocktail Party Problem is a classic example of source separation which is very popular in digital signal processing. In this problem, several people are talking to each other in a banquet room and a listener is trying to recognize one specific speech from that crowd of partying guests. Human brain can distinguish one explicit signal component from a mixed signal combination in real time which is popularly known as "Auditory Scene Analysis." However, in digital signal processing, it is difficult to extract only one speaker's voice from the rest in cocktail party situation.

In [24], Colin Cherry first revealed the ability of human auditory system to separate a single speech or audio from a combination of voices, which may turn into noise through properties like pitch, gender, rate of speech, and/or direction of speech. This task of separating single source audio from a noise is known as dichotic listening task [25]. In [26], authors have reviewed the same techniques to train machine to segregate signals. In [27], Broadbent has concluded that simultaneous listening can be performed for small messages, not for long ones. Human ability to identify audio from a mixed signal can be improved by listening by two ears [28]. It has been seen that, in ideal circumstances, the signal detection threshold of binaural listening is 25 dB more than monaural listening. In [29], it has been stated that cocktail party effect can be explained by Binaural Masking Level Difference (BMLD). As per BMLD, for binaural listening the desired signal coming from one direction is ineffectively masked by the noise generated in different direction. In [30], Kassebaum et al. discussed two methods for signal separation—Back Propagation (BP) and Self-Organizing Neural Network (SONN). That experiment was carried out through 4 kHz channel using a modem data signal and a male speech signal. It has been concluded that BP requires more inputs and training time than SONN.

In [31] authors have discussed 3 types of approach to solve Cocktail Party Problem:

(i) Temporal binding and oscillatory correlation

(ii) Cortronic network

(iii) Blind source separation.

In [32], von der Malsburg explained the temporal binding technique. He stated that neuron carries two distinct signals and the binding is accomplished by correlation. The synchronization allows neuron to create topological network. In [33], von der Malsburg and Schneider proposed a cocktail party processor enhancing this idea—the Oscillatory Correlation which is the basis of Computational Auditory Scene Analysis. In [34, 35], multistage neural model has been proposed to separate speech from interfering sounds using oscillatory correlation.

In [36], authors have proposed a biological approach to solve Cocktail Party Problem using artificial neural network named as cortronic network. A cortronic neural network describes connection among neurons in several regions which demonstrates the output links of each neuron and the strength of the connections.

The Blind Source Separation (BSS) is the technique of separating signal from a mixed source without having knowledge of source signals and the process of mixing. There are different methods of BSS among which Principal Component Analysis (PCA), Independent Component Analysis (ICA), and Time and Frequency domain approaches are significant. PCA and ICA are both statistical approaches which are better than Time or Frequency domain approach, since Fourier components of data segments are fixed in frequency domain whereas in statistical domain the transformation depends on the data to be analyzed [37].

PCA is a mathematical technique of transforming large correlated dataset into a small number of major components known as principal components [38]. It is moderately related to mathematical theory of Singular Value Decomposition (SVD), which is used to implement PCA [39]. Independent Component Analysis can also be implemented with SVD, though there are subtle differences between PCA and ICA. The aim of PCA is to find decorrelated variables whereas the aim of ICA is to find independent variables. PCA and ICA both perform matrix factorization for linear transformation, though PCA perform low rank matrix factorization whereas ICA performs full-rank matrix factorization. The

Table 1: Advantage and disadvantage of different approaches for solving Cocktail Party Problem.

Approach for solving Cocktail Party Problem	Advantages of the method	Disadvantages of the method
Temporal binding	As shown in [44], this strategy helps √ robustness against loss of network elements √ richness of representation √ processing speed enhancement	As stated in [45], this strategy results ✘ inflexible refocusing of system onto events rapidly occurring in sequence
Cortronic network	As mentioned in [36], in this method √ there is no requirement for having knowledge of background sounds such as static, traffic, and music	As shown in [36], this technique is ✘ costly to implement as it requires a separate artificial neural network
Blind source separation	As shown in [46], in this technique √ there is no need for having knowledge of source signals or the process of mixing √ no need for defining a cut-off frequency for separation √ low computational complexity √ helps signal enhancement	As reported in [47], in this method ✘ convergence speed is slow

advantage of ICA over PCA is that PCA just removes correlations whereas ICA removes correlations and higher order dependencies [40]. ICA has extensive use in biomedical imaging and audio processing [41]. ICA can also be used for transformation to independent variable using multiplication of observed data and for demixing matrix [42]. It depends on the fact that there are as many sources as channels of data available, which are to be separated as independent sources—by utilizing this fact, ICA is used in Blind Source Separation. In [43], author described a fast method for ICA using fixed point iteration. This algorithm is popularly known as FastICA.

In Table 1, comparison of the existing techniques for solving Cocktail Party Problem has been discussed. It can be noted that each of these techniques has its own advantage and disadvantages. However, as blind steganographic approach is considered more robust and secure than the nonblind steganography techniques, hence, in this proposed method, "Blind Source Separation" approach has been chosen for solving cocktail party effect.

3. Proposed Method

3.1. In a Nutshell. Steganography can be broadly grouped into two types: blind and nonblind techniques. The technique where cover object is not required to retrieve the secret is called blind steganography. The method where cover object is required to regain secret is called nonblind or cover escrow technique of steganography. To create a most robust method of steganography, here a blind steganography technique has been proposed.

In this proposed method, image has been used as secret message. This secret image is scrambled using Arnold transform. Then Haar filter is applied for two-dimensional DWT on the cover source audio. Since audio is one-dimensional signal, hence it must be reshaped into two-dimensional matrix to perform 2D DWT. Haar is simple, fast, and memory efficient compared to other available DWT filters like Daubechies and Coiflets. After DWT application, LH subband has been chosen for further decomposition into 4×4 blocks where two-dimensional DCT has been applied. As shown in Figure 2(b), in Section 2.1, midband region of those 4×4 blocks has been chosen and embedding has been performed by the following equation:

$$\text{mid}\left(\dot{F}\left(C_a\right)\right) = \text{mid}\left(\dot{F}\left(C_a\right)\right) + \propto \times \text{PN}, \tag{9}$$

where $\text{mid}(\dot{F}(C_a))$ indicates midband frequency region; $\propto$ is the embedding factor; and PN is the pseudorandom number. Equation (9) has been further explained in Section 3.6; embedding factor ($\propto$) has been discussed in Section 3.4 and pseudorandom number (PN) has been discussed in Section 3.5.

After embedding, the resultant cover becomes stego audio. To increase security of the proposed method, this stego audio is blended with other audio signals to produce cocktail party effect—afterwards this has been securely transmitted through the web to reach the intended recipient. Even if any intruder is able to break the communication channel and get access to the transmitted media, neither he would decipher the cocktail party effect to identify stego audio nor he would able to decode the stego audio to recognize the secret message without knowing the key required for extraction, whereas the intended receiver knowing the key as well as the entire algorithms is able to easily extract the secret message implanted without any loss of data. The proposed method is also tested against well-known Steganalysis attacks and the outcomes are quite impressive (discussed in Section 4.3)—hence this technique provides complete security.

Once the intended recipient receives the cocktail effect, using the demixing algorithm (discussed in Section 3.8) s/he

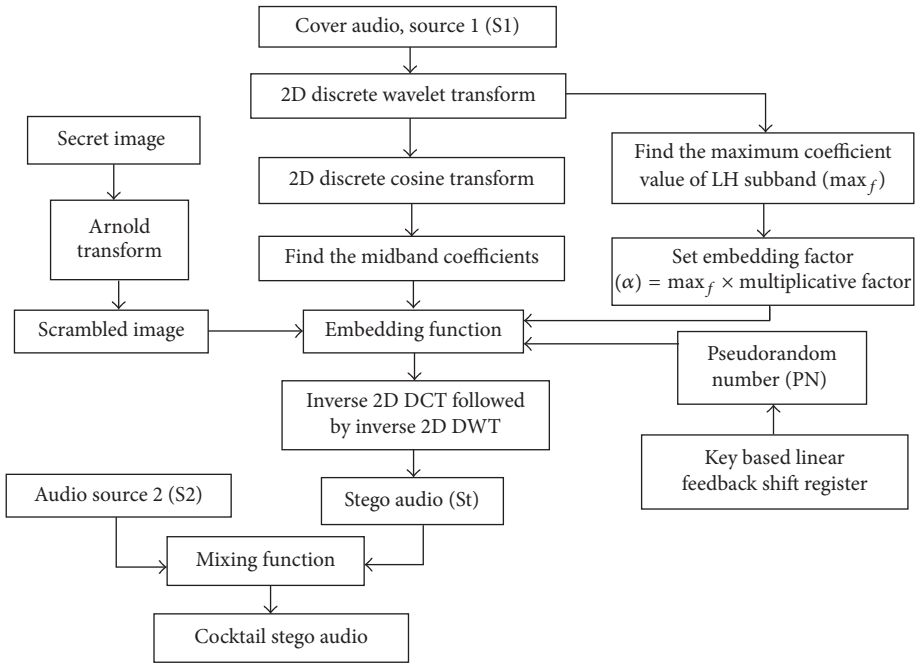


Figure 4: Flowchart of embedding procedure.

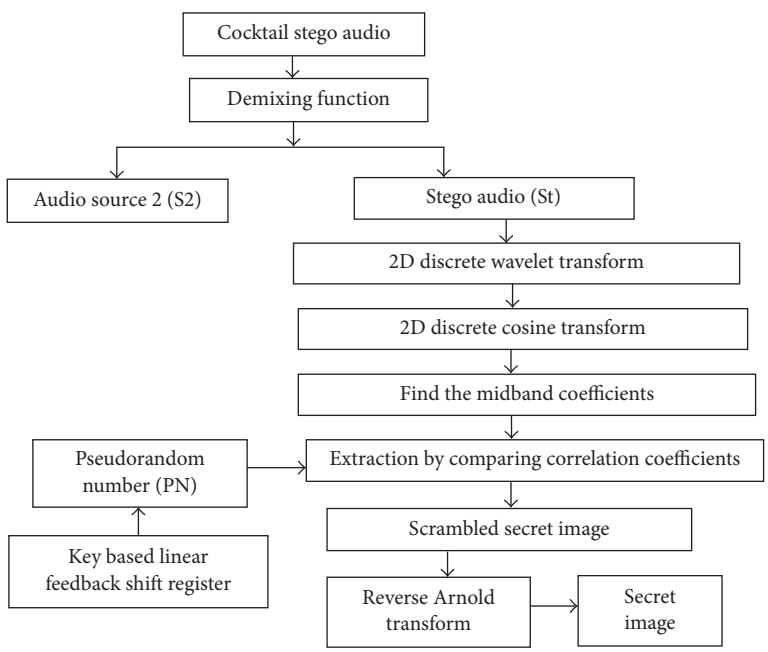


Figure 5: Flowchart of extraction procedure.

can separate the audios and can also apply the extraction procedure on them, as the recipient is aware of the key. The extraction algorithm performs correlation between the coefficients and extracts the secret bits, from which the scrambled secret image can be generated. Finally, by applying inverse Arnold transform, the secret image can be reconstructed. The flowcharts for embedding and extraction procedure have been shown in Figures 4 and 5, respectively.

3.2. Input Preparation

Cover Audio Source. Any speech or music can be used here as cover audio sources. For this demonstration, popular English songs have been chosen—as mentioned below. All the audio sources have been sampled at 44100 kHz in monochannel with 16-bit depth, cut to 26 seconds' duration for optimizing embedding capacity calculation, and finally saved as .wav file.

The following are the audio sources used for this research experiment:

(1) "My Heart Will Go On" by Celine Dion from film "Titanic" → saved as tt.wav

(2) "Beat It" by Michael Jackson from album "Thriller" → saved as mj.wav

(3) "Like a Rolling Stone" by Bob Dylan from album "Highway 61 Revisited" → saved as bob.wav

(4) Title song from film "Mamma Mia!" by Meryl Streep → saved as mm.wav

(5) Title song from film "High School Musical" by chorus → saved as hsm.wav.

Secret Image. Though any types of grayscale image (.jpg or .bmp) can be used here as secret, however for this experiment binary images (.pbm) have been chosen for better quality extraction. For this proposed method, secret images need to transform to binary, which is lossy conversion; hence any true-color RGB images cannot be applied here as, after extraction, the retrieved image will only have two colors—black and white. Secret image size here is taken as 128 × 128, which can be further increased if the length of input cover audio source is more than 26 seconds. For this experiment, secret images have been either downloaded from Internet (these do not have any copyright restriction) or drawn by Microsoft Paint software.

3.3. Scrambling and Descrambling Algorithm for Secret Image. The "Arnold transform" algorithm randomizes the input image by number of iterations to create scrambled image.

Input: Any binary Image ($I_{m\times n}$), number of iteration (t)
Output: Scrambled Image (I_{out})
Algorithm: written as function Arnold ($I_{m\times n}, t$)
Step 1: Find out the size of I and store in m and n
Step 2:
for $j = 1$ to t
for $y = 0$ to n
for $x = 0$ to n
Find out $P = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}$;
$I_{\text{out}}(\text{mod}(P(2), m) + 1, \text{mod}(P(1), m) + 1)$
$\leftarrow I(y + 1, x + 1)$;
end;
end;
$I = I_{\text{out}}$;
end;

Once applied to the scrambled image, the "Reverse Arnold Transform" algorithm returns the original secret image after specified iterations.

Input: Any scrambled binary Image ($I_{m\times n}$), number of iteration (t)
Output: Descrambled Image (I_{out})
Algorithm: written as function iArnold ($I_{m\times n}, t$)
Step 1: Find out the size of I and store in m and n
Step 2:
for $j = 1$ to t
for $y = 0$ to n
for $x = 0$ to n
Find out $P = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}$;
$I_{\text{out}}(\text{mod}(P(2), m) + 1, \text{mod}(P(1), m) + 1)$
$\leftarrow I(y + 1, x + 1)$;
end;
end;
$I = I_{\text{out}}$;
end;

3.4. Embedding and Multiplicative Factors. As shown in (9) in Section 3.1, embedding factor (α) has been multiplied with PN to offset the increment of DCT coefficient value such that, after embedding, stego audio will not have any audible noise. Hence the value of α must be between 0 and 1. After repeated experiments, it has been observed that when value of embedding factor nears 1, then the extracted message is having very high PSNR and SSIM—which tends to high robustness—however simultaneously, in stego audio, there are audible artifacts identified, which is differentiating with the cover audio. This signifies value of α near to 1 compromise imperceptibility. On the other hand, if the value of α approaches 0, the stego audio would be just like the original cover audio (the PSNR between these two audios reaches around 100 dB), whereas then the secret image extracted is completely corrupted. These test results indicate that, to get an optimum outcome, the tradeoff must be done between robustness and imperceptibility.

While experimenting with several cover audios along with various secret images, it has been also noticed that keeping a constant value of embedding factor (α) cannot ensure similar quality outcome, after extraction. Henceforth it is decided to set α depending on the cover to generate the optimal result. As the data hiding takes place in the LH subband of DWT, hence, to formularize α, maximum coefficient value of the LH subband has been chosen as one of the aspects of the following formula:

$$\begin{aligned} &\text{Embedding Factor } (\alpha) \\ &\quad = \text{Multiplicative Factor} \\ &\qquad \times \text{Max (coefficients of LH)}. \end{aligned} \tag{10}$$

Finally, for this proposed method, the value of Multiplicative Factor has been universally set as 0.2, based on the experimental outcome, as shown in Table 2.

3.5. Pseudorandom Number. For embedding secret into cover, in this proposed method "pseudorandom number" (PN) has been used; PN is generated using Linear Feedback

Table 2: Experimental Results with different embedding and multiplicative factors.

Original secret	Extracted secret image	Embedding factor	PSNR of extracted secret	SSIM of extracted secret	PSNR of stego audio
		0.1x Maximum Coefficient Value of LH	62.0078	0.9990	88.1656
		0.2x Maximum Coefficient Value of LH	72.8714	0.9999	82.1450
		0.3x Maximum Coefficient Value of LH	84.2544	1.0000	78.6231

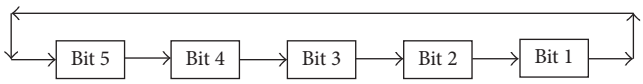


Figure 6: Simplified block diagram of LFSR.

Shift Register (LFSR), as shown in Figure 6. Here LFSR has been designed using only right shift operator and the operation of this shift register is completely deterministic. It must be initialized with a set of numbers and, at any given point, the value of LFSR can be determined by its present state.

In this proposed method, two simple algorithms have been designed to generate two different sets of PN values for a given key with the same initial sequence of numbers. This initial sequence can be altered any time. Here, for easy illustration purpose, "0 0 0 0 1" has been chosen as initial sequence.

Description: The below algorithm(s) generates endless non-sequential lists of numbers in binary base using Linear Feedback Shift Register.
Input: A number as Key
Output: Pseudo-random Numbers, PN1[] and PN2[] respectively.
Algorithm 1: written as function SRPN1 (Key)

```
Step 1: set N = Key;
Step 2: set initial state of shift register as
state = [0 0 0 0 1]
Step 3: set PN1 = [];
Step 4:
  for j = 1 to N
    PN1 = [PN1 state(5)]
    if state(1) == state(4)
      then set temp = 0;
      else set temp = 1;
    end;
    set state(1) = state(2);
```

```
        set state(2) = state(3);
        set state(3) = state(4);
        set state(4) = state(5);
        set state(5) = temp;
      end;
```

Algorithm 2: written as function SRPN2 (Key)

Step 1: set N = Key;

Step 2: set initial state of shift register as state = $[0\ \ 0\ \ 0\ \ 0\ \ 1]$

Step 3: set PN2 = [];

Step 4:

```
  for j = 1 to N
   PN2 = [PN2 state(5)]
   if state(1) == state(2)
      then set temp 1 = 0;
      else set temp 1 = 1;
   end;
   if state(4) == temp 1
      then set temp 2 = 0;
      else set temp 2 = 1;
   end;
   if state(5) == temp 2
      then set temp 3 = 0;
      else set temp 3 = 1;
   end;
   set state(1) = state(2);
   set state(2) = state(3);
   set state(3) = state(4);
   set state(4) = state(5);
   set state(5) = temp 3;
  end;
```

3.6. Embedding Algorithm. To ensure more security and imperceptibility, in this proposed method, the secret message is embedded in the transform domain using discrete wavelet transform (DWT) as well as by discrete cosine transform (DCT).

Description: algorithm for embedding secret data.

Input: a Cover Audio (C_a), Secret message as an image (S_I)

Output: a Stego Audio (Steg_Aud).

Algorithm:

Step 1: read cover audio (C_a)

Step 2: read secret message (S_I)

Step 3: set iteration as a number = t

Step 4: call function Arnold(S_I, t) which returns scrambled image (SS_I)

Step 5: set Key as a number = N

Step 6: call function SRPN1(N) which returns PN1[];

Step 7: call function SRPN2(N) which returns PN2[];

Step 8: apply 2D DWT on C_a to decompose in LL, LH, HL and HH;

Step 9: find $\max_f$ = max(value of coefficients in LH);

Step 10: set embedding factor (α) = Multiplicative Factor × $\max_f$

Step 11: apply 2D DCT over LH and get $\dot{F}(C_a)$.

Step 12: find mid-band coefficient region of $\dot{F}(C_a)$ and term it as mid($\dot{F}(C_a)$);

Step 13: if $SS_I(x, y) == 0$

then set mid($\dot{F}(C_a)$) = mid($\dot{F}(C_a)$) + $\alpha\times$ PN1[];
else set mid($\dot{F}(C_a)$) = mid($\dot{F}(C_a)$) + $\alpha\times$ PN2[]; end;

Step 14: perform inverse DCT to get new(LH).

Step 15: perform inverse DWT using LL, new(LH), HL, HH and get Stego

Step 16: write Stego in Steg_Aud

3.7. Mixing Algorithm. This algorithm mixes two audio sources from two different channels to create cocktail effect of two audio signals.

Input: two monochannel .wav files (S_1 and S_2) having same duration and sampling rate of 44100 Hz

Output: .wav files having cocktail sound effect (S_3 and S_4)

Algorithm: written as function Mixing (S_1, S_2)

Step 1: set Gain Factor (g) as decimal ($0 < g < 1$)

Step 2: read S_1 and S_2 in sig_1 & sig_2 while keeping their respective sampling frequencies stored in Fs_1 and Fs_2

Step 3: set $\text{Mixed}_1 = \text{sig}_1 + (g \times \text{sig}_2)$ and $\text{Mixed}_2 = \text{sig}_2 + (g \times \text{sig}_1)$;

Step 4: write Mixed_1 in audio file S_3 with Fs_1 and write Mixed_2 in audio file S_4 with Fs_2

3.8. Demixing Algorithm. Here, for demixing, FastICA MATLAB package (ver. 2.5) has been used which estimates the independent components from given multidimensional signals using Blind Source Separation technique.

Input: two .wav files (S_3 and S_4) containing mixed signals from different channels

Output: two unmixed source.wav files (S_1, S_2)

Algorithm: written as function Demixing (S_3, S_4)

Step 1: read S_3 and S_4 in Y & Z while keeping their respective sampling frequencies stored in Fs_1 and Fs_2

Step 2: find complex conjugate transpose of Y and Z, store them in A and B

Step 3: create one matrix from A and B, store it in X

Step 4: set S = FastICA(X);

Step 5: extract two sources from S as source_1 and source_2

Step 6: write source_1 in S_1 with Fs_1 and source_2 in S_2 with Fs_2

3.9. Extraction Algorithm

Input: stego audio (Steg_Aud)

Output: secret image (S_I)

Algorithm:

Step 1: read Stego audio (Steg_Aud) in S_a

Step 2: set Key as a number = N

Step 3: call function SRPN1(N) which returns PN1[];

Step 4: call function SRPN2(N) which returns PN2[];

Step 5: apply 2D DWT on S_a to decompose it in LL, LH, HL and HH;

Step 6: apply 2D DCT over LH and get $\dot{F}(S_a)$

Step 7: find mid-band coefficient region of $\dot{F}(S_a)$ and term it as $\text{mid}(\dot{F}(S_a))$

Step 8: if Correlation($\text{mid}(\dot{F}(S_a))$, PN1[]) >= Correlation($\text{mid}(\dot{F}(S_a))$, PN2[])

then $SS_I(x, y) = 0$ else $SS_I(x, y) = 1$; end;

Step 9: reshape the image bits stored in SS_I to get secret scrambled image

Step 10: set iteration as a number = t

Step 11: call function iArnold (SS_I, t) which returns secret image (S_I)

4. Experimental Results and Analysis

This proposed method has been applied on several sets of cover audio and secret images, though, for efficient use of space, here only 2 sets of robustness test results have been presented for Steganalysis attacks.

4.1. Adherence to Kerckhoff's Principle. In this research article, a key based steganography technique has been proposed. Hence it should follow Kerckhoff's principle of cryptography [48], which says an exemplary method should be secure even if the public is aware of all the details of that method except the key. As mentioned in Section 3.5, here LFSR has been used both at sender's end and at receiver's end. It requires a unique key to generate the same set of pseudorandom numbers [49] which are used in embedding equation (9) and again in Step8 of the extraction algorithm for comparing correlation coefficients. If the exact same key is not used during embedding and extraction, then LFSR will generate different set of pseudorandom numbers using which secret image cannot be extracted from the stego audio. Henceforth it is proved that the proposed method complies with Kerckhoff's principle.

4.2. Outcome of Quality Metrics

Embedding Capacity (EC). EC is measured by the ratio between size of hidden message (in bits) and size of cover (in bits), as shown in (11) below. In this research experiment, it has been observed that, to hide 128 × 128 size of a secret image, it requires cover audio size of 1048576 bits—which implies embedding capacity value of 1.5625%. Similarly, to implant a 64 × 64 secret image, 262144 bits of cover audio is needed—this again confirms the proportion of embedding capacity as 1.5625%.

$$\text{capacity} = \frac{\text{size of hidden data}}{\text{size of cover data}} \times 100\%. \tag{11}$$

Peak Signal-to-Noise Ratio (PSNR). PSNR represents the ratio between maximum power of test signal and the power of reference signal. The mathematical representation for PSNR is as follows:

$$\text{PSNR} = 10\log_{10}\left(\frac{{\text{Max}_{\text{sf}}}^2}{\text{MSE}}\right), \tag{12}$$

where Max_{sf} is maximum signal value or maximum fluctuation in the input image data type (e.g., for 8-bit unsigned integer data type, Max_{sf} is 255) and MSE is the Mean Squared Error, which is given by

$$\text{MSE} = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left[S_{\text{Ref}} - S_{\text{Test}}\right]^2, \tag{13}$$

where S_{Ref} represents original signal; S_{Test} represents degraded signal; m and n represent numbers of rows and columns of the signal matrix, respectively; i represents index of row and j represents index of column.

Structural Similarity Index (SSIM). SSIM is a measurement of similarity, calculated through luminance, contrast, and structural differences between two images as given below.

$$\text{SSIM (S, E)} = \frac{\left(2\mu_{\text{S}}\mu_{\text{E}} + c_1\right)\left(2\sigma_{\text{SE}} + c_2\right)}{\left(\mu_{\text{S}}^2 + \mu_{\text{E}}^2 + c_1\right)\left(\sigma_{\text{S}}^2 + \sigma_{\text{E}}^2 + c_2\right)}, \tag{14}$$

where μ_{S} and μ_{E} are the mean of secret image S and extracted image E, respectively; σ_{S} and σ_{E} are the standard deviation of S and E; σ_{SE} is correlation of S and E.

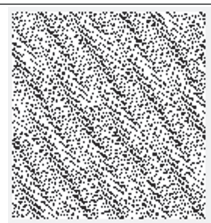
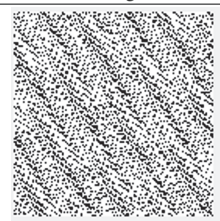
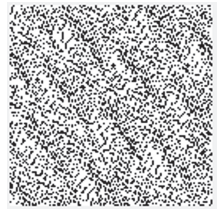
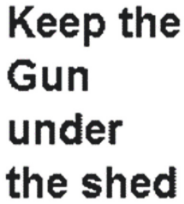
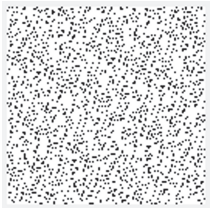
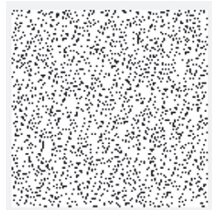
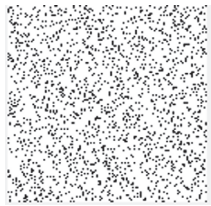
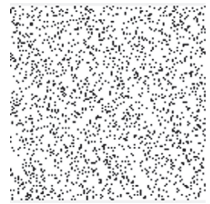
Bit Error Rate (BER). BER is defined by number of error bits divided by total number of transmitted bits, as shown in the following equation:

$$\text{BER} = \frac{N_{\text{ErrorBit}}}{N_{\text{BitsTransmitted}}} \times 100. \tag{15}$$

Here the BER is calculated between original secret image and extracted secret image.

Table 3 shows the quality outcome of the secret and extracted images with respect to PSNR, SSIM, BER, and correlation coefficient (CC, discussed in Section 2.2).

Table 3: Quality analysis of secret and extracted image.

Secret image (I_S)	Scrambled secret image	Extracted scrambled image	Extracted image (I_E)	PSNR (I_S, I_E)	SSIM (I_S, I_E)	BER (I_S, I_E)	CC (I_S, I_E)
				72.8714	0.9999	1.0925	0.9717
				70.9007	0.9999	1.0620	0.9729
				68.3717	0.9998	0.9460	0.9597
				72.3511	0.9999	0.3174	0.9852

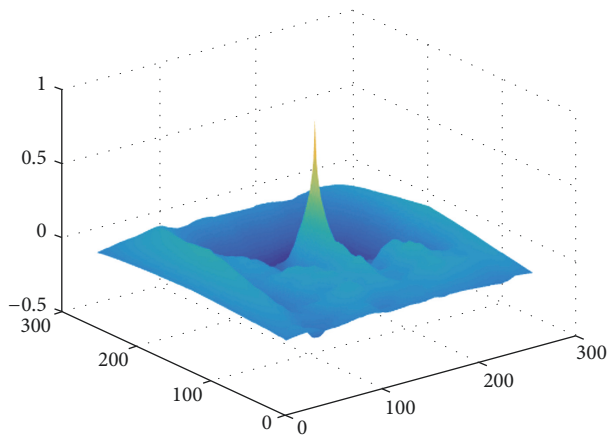


Figure 7: Surface plot of NCC between secret and extracted image.

Perceptual Evaluation of Audio Quality (PEAQ). PEAQ is a standardized metric to evaluate audio quality utilizing human perceptual properties, output of which is given in a scale of 1 to 5 (where 1 signifies poor and 5 implies excellent) depending on the Mean Opinion Score (MOS) of all listeners. The quality of output audio is measured by comparing with a reference audio.

Normalized Cross-Correlation (NCC). NCC quantifies degree of similarity between two signals. NCC computes normalized two-dimensional cross-correlation values between two image metrics. The values of correlation coefficients lie between −1 and 1, where 1 signifies identical images and −1 denotes totally different image. It is formulated as

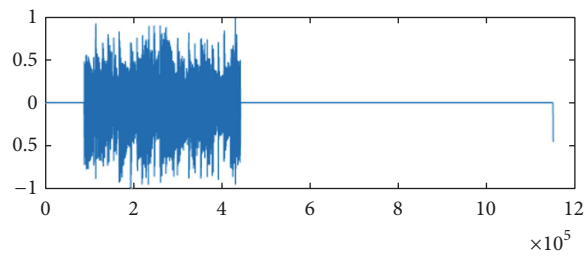
$$\mathrm{NCC} = \frac{\sum_{p=1}^{A}\sum_{q=1}^{B} X(p,q)\,\overline{X}(p,q)}{\sqrt{\sum_{p=1}^{A}\sum_{q=1}^{B} X(p,q)^2}\sqrt{\sum_{p=1}^{A}\sum_{q=1}^{B} \overline{X}(p,q)^2}}, \quad (16)$$

where $X(p,q)$ is the extracted image and $\overline{X}(p,q)$ is the reference image. NCC is used to produce surface plot, which depicts functional relationship between two independent variables and map to a plane which is parallel to X-Y plane. Here, in Figure 7, the surface plot of NCC between secret and extracted image has been shown.

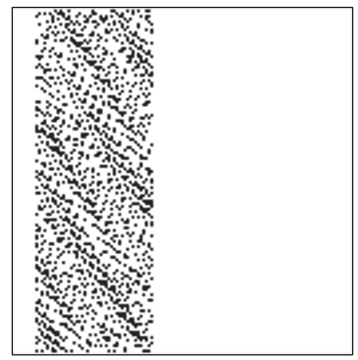
In Table 4, quality analysis of the cover and stego audio has been shown in PSNR, PEAQ, and CC.

4.3. Robustness Tests by Steganalysis Attacks

By Random Cropping. On average, English music or a full song has duration of over 5 minutes, that is, more than 300 seconds. In this proposed method, only 25 seconds of audio is required to hide a secret image having size of 128 × 128. This secret can be kept anywhere within the stego, that is, at the start or at the end or after nth seconds—in short, the secret can be moved throughout the cover and the exact place of hiding is not predetermined. That is why 9 out of 10 attempts of random cropping leave the secret image intact, as stego has been cropped elsewhere. For the remaining 1 out of 10 attempts, that is, when the stego audio has been cropped in such a place where secret image was embedded, Figures 8(a), 8(b), and 8(c) provide the results.



(a) Graphical plot of audio during cropping



(b) Scrambled secret

(c) Extracted secret

Figure 8

As shown in Figure 8(a), from a stego audio of 26 seconds' duration, 8-second-long window (from 2nd to 10th second) has been chosen and the remaining audio signal has been replaced with zero. When the intended recipient applies the extraction mechanism on such modified stego audio, it generates only a portion of scrambled secret image as shown in Figure 8(b). However, when "Reverse Arnold Transform" has been applied on such partially scrambled secret image, it still recovers the extracted secret as shown in Figure 8(c). Quality analysis of the extracted secret image has revealed PSNR value of 55.7633 and SSIM value of 0.9867, when compared with the original secret image which was embedded.

By Adding White Gaussian Noise. In this type of attack, "Additive White Gaussian Noise" (AWGN) is added to the stego audio to distort the hidden message. AWGN can be added to any signal, and it has uniform power and is

Table 4: Quality analysis of cover and stego audio.

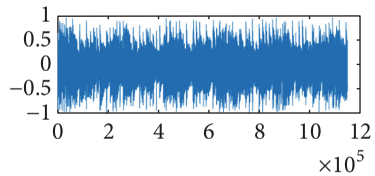
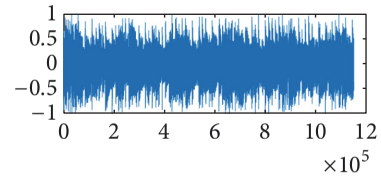
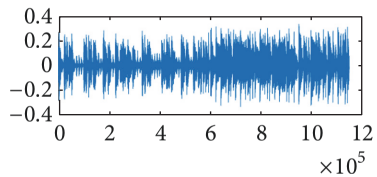
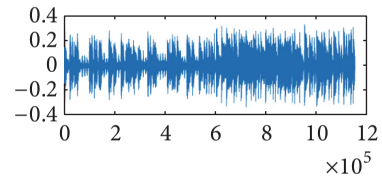
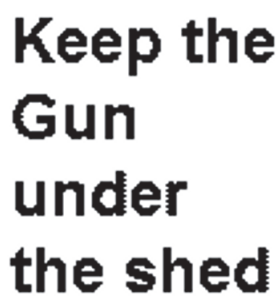
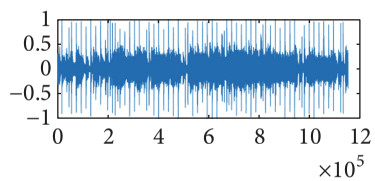
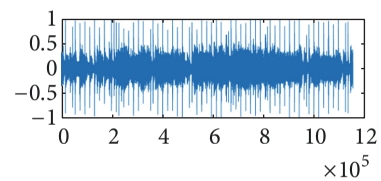
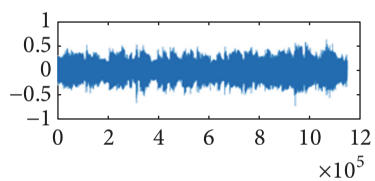
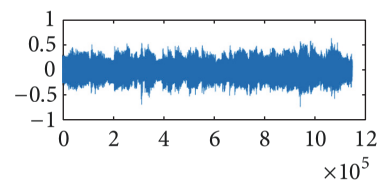
Secret image and cover audio	Graphic plot of cover audio	Graphic plot of stego audio	PSNR (in dB)	PEAQ (in MOS)	CC
 embedded in bob.wav			86.1450	3.99	0.9854
 embedded in mm.wav			90.8836	4.02	0.9854
 embedded in mj.wav			82.1531	3.98	0.9858
 embedded in tt.wav			90.5024	4.04	0.9986

Table 5: Experimental results of AWGN stego analysis attack.

Original secret	Extracted secret after adding noise at SNR 20 dB	Extracted secret after adding noise at SNR 30 dB	Extracted secret after adding noise at SNR 40 dB
	 **PSNR:** 69.4114 **SSIM:** 0.9998	 **PSNR:** 72.1459 **SSIM:** 0.9999	 **PSNR:** 72.7163 **SSIM:** 0.9999
	 **PSNR:** 67.5566 **SSIM:** 0.9997	 **PSNR:** 69.5195 **SSIM:** 0.9998	 **PSNR:** 69.6304 **SSIM:** 0.9998

Table 6: Outcome of resampling attack.

Original secret embedded in cover audio of sampling rate 44100 Hz	Extracted secret after changing sampling rate to 320000 Hz	Extracted secret after changing sampling rate to 480000 Hz	Extracted secret after changing sampling rate to 960000 Hz
	 **PSNR**: 55.9038 **SSIM**: 0.9933	 **PSNR**: 68.8762 **SSIM**: 0.9998	 **PSNR**: 68.8762 **SSIM**: 0.9998
	 **PSNR:** 55.0401 **SSIM:** 0.9915	 **PSNR:** 66.0920 **SSIM:** 0.9996	 **PSNR:** 66.1086 **SSIM:** 0.9996

Table 7: Results of requantization attack.

Original secret embedded in cover audio of 16-bit depth	Extracted secret after changing audio bit depth to 8 bits	Extracted secret after changing audio bit depth to 24 bits	Extracted secret after changing audio bit depth to 32 bits
			
	PSNR: 72.2816 **SSIM**: 0.9999	**PSNR**: 72.1459 **SSIM**: 0.9999	**PSNR**: 72.1459 **SSIM**: 0.9999
			
	PSNR: 67.7953 **SSIM:** 0.9998	**PSNR:** 67.7465 **SSIM:** 0.9998	**PSNR:** 67.7465 **SSIM:** 0.9998

Table 8: Experimental results of pitch shifting attack.

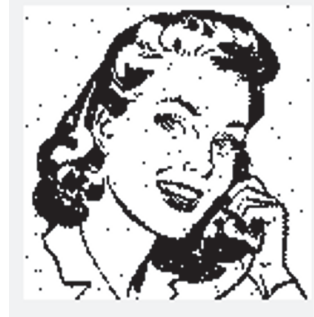
Original secret in cover audio	Extracted secret after 20% reduction of audio pitch	Extracted secret after 20% increment of audio pitch	Extracted secret after 40% increment of audio pitch
	 **PSNR**: 68.9396 **SSIM**: 0.9998	 **PSNR**: 69.6680 **SSIM**: 0.9998	 **PSNR**: 70.1890 **SSIM**: 0.9999
	 **PSNR:** 63.4898 **SSIM**: 0.9993	 **PSNR:** 65.0246 **SSIM:** 0.9995	 **PSNR:** 65.9614 **SSIM:** 0.9996

Table 9: Experimental results of MP3 compression attack.

Secret embedded in cover audio .wav file	Extracted secret from MP3 of bitrate 128 kbps	Extracted secret from MP3 of bitrate 192 kbps	Extracted secret from MP3 of bitrate 320 kbps
	 **PSNR**: 62.4718 **SSIM**: 0.9991	 **PSNR**: 63.1317 **SSIM**: 0.9992	 **PSNR**: 63.2335 **SSIM**: 0.9992
	 **PSNR:** 60.7422 **SSIM:** 0.9985	 **PSNR:** 61.1528 **SSIM:** 0.9987	 **PSNR:** 61.2550 **SSIM:** 0.9987

distributed with respect to time. As shown in Table 5, to test robustness of the proposed method, here 20, 30, and 40 dB of SNR (Signal-to-Noise Ratio) per sample is added to the stego audio signal, assuming the power of stego signal is 0 dBW (decibel-watt is a unit of power in decibel scale, relative to 1 watt).

By Resampling. While writing audio data into a file, sampling rate of the audio is generally mentioned as Fs. In the resampling attack, at first this sampling rate has been changed to a higher or lower frequency while saving the same audio in a new file. As resampling causes impact on audio file length, hence, to maintain the same length as of original cover, modified audio has been cut or filled with zeros. Once saved, resampling has been performed again on the modified audio to revert it back to the original sampling frequency—by this, audibly no differences will be noted; however it will distort the embedded secret message (if any). In Table 6, result of such resampling attack has been shown.

By Requantization. The number of bits required to express each audio sample is known as bit depth. It is a measurement of sound accuracy: the higher the bit depth is, the more it would be precise. In the requantization attack, this bit depth of stego audio has been changed to pervert the embedded secret image. Table 7 illustrates the outcome of the extraction process after requantization attack.

By Pitch Shifting. Pitch means tone of a signal; it describes the quality of a sound by the rate of vibrations. In pitch shifting attack, original pitch of an audio is lifted or dropped without modifying its length to destroy the hidden message embedded in a stego audio. Here pitch shifting has been done by utilizing time-scale modification algorithm called "Phase Vocoder" [50], the result of which is shown in Table 8.

By MP3 Compression. In this Steganalysis attack, stego.wav file has been compressed to MP3 format to eliminate redundant data, by which embedded secret message would be completely removed. Here mp3write MATLAB function has been used to convert the stego.wav file into mp3 format and mp3read MATLAB function has been applied to read from the mp3 file during extraction process.

Table 9 reflects the extraction outcome from three different mp3 files of the same stego audio which has been encoded with bitrates 128 kbps, 192 kbps, and 320 kbps, respectively.

4.4. Comparison with Existing Method. For comparison with the proposed method, research articles published in SCI indexed journal have been searched—where data hiding in audio has been performed by DWT along with DCT and extraction mechanism is blind. Authors of [51] have proposed DCT-DWT based data hiding technique using 16-bit Barker code as synchronizing code to accommodate 64 × 64 binary image as secret message. From the comparison results presented in Table 10, this can be proved that the proposed method has outperformed the existing one in terms of quality and robustness test against Steganalysis attacks.

Table 10: Comparison results.

Features based comparison and robustness tests	Proposed method	Existing method [51]
Secret message size	128 × 128	64 × 64
Adherence to Kerckhoff's principle	√	✘
Peak signal to noise ratio	72.8714	-
Structural similarity index	0.99	-
Perceptual evaluation of audio quality	√	-
Addition of white Gaussian noise	√	√
Random cropping Steganalysis attack	√	✘
Resampling Steganalysis attack	√	√
Requantization Steganalysis attack	√	√
Pitch shifting Steganalysis attack	√	✘
MP3 compression Steganalysis attack	√	√

In Table 10, "√" signifies "satisfactory result obtained"; "✘" signifies "unsatisfactory result or method does not comply"; and "-" implies "details not mentioned."

5. Conclusion

Secret communication using age-old steganography techniques often increases chances of detectability through the perceivable noise. Hence, in this article, the cocktail party effect has been considered which has effectively reduced the probability of detectability. This has also been proved by the help of different Steganalysis techniques. Additionally, PSNR, CC, and PEAQ values are also analyzed to determine the perceptual noise recorded due to secret message embedding and extraction. Since all the above results verify the undetectability and robustness of the system, hence it can be concluded that this audio steganography technique is successful in secret communication with very high robustness.

In future, this proposed method can be further improvised by utilizing speaker diarization technique, which determines "who spoke when." Application of speaker diarization along with speech recognition would identify a speaker's voice and this concept will permit segregating secret audio stream into multiple speech segments, ensuring another novel approach of data hiding.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] B. G. Banik and S. K. Bandyopadhyay, "Review on steganography in digital media," *International Journal of Science and Research*, vol. 4, no. 2, pp. 265–274, 2015.

[2] M. Asad, J. Gilani, and A. Khalid, "An enhanced least significant bit modification technique for audio steganography," in *Proceedings of the 1st International Conference on Computer Networks and Information Technology (ICCNIT '11)*, pp. 143–147, IEEE, Pakistan, July 2011.

[3] N. Cvejic and T. Seppanen, "Increasing the capacity of LSB-based audio steganography," in *Proceedings of the 2002 5th IEEE Workshop on Multimedia Signal Processing (MMSP '02)*, pp. 336–338, IEEE, USA, December 2002.

[4] Jayaram, Ranganatha, and Anupama, "Information Hiding Using Audio Steganography - A Survey," *The International Journal of Multimedia & Its Applications*, vol. 3, no. 3, pp. 86–96, 2011.

[5] D. Gruhl, A. Lu, and W. Bender, "Echo hiding," in *Information Hiding*, vol. 1174 of *Lecture Notes in Computer Science*, pp. 295–315, Springer Berlin Heidelberg, Berlin, Heidelberg, 1996.

[6] D. Xiaoxiao, M. F. Bocko, and Z. Ignjatovic, "Robustness analysis of a digital audio steganographic method based on phase manipulation," in *Proceedings of the 7th International Conference on Signal Processing (ICSP '04)*, vol. 3, pp. 2375–2378, IEEE, Beijing, China, 2004.

[7] N. Parab, M. Nathan, and K. T. Talele, "Audio Steganography Using Differential Phase Encoding," in *Technology Systems and Management*, vol. 145 of *Communications in Computer and Information Science*, pp. 146–151, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[8] R. M. Nugraha, "Implementation of Direct Sequence Spread Spectrum steganography on audio data," in *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics (ICEEI '11)*, pp. 1–6, IEEE, Indonesia, July 2011.

[9] H. Matsuoka, "Spread spectrum audio steganography using sub-band phase shifting," in *Proceedings of the 2006 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP '06)*, pp. 3–6, IEEE, USA, December 2006.

[10] G. Prabakaran and R. Bhavani, "A modified secure digital image steganography based on discrete wavelet transform," in *Proceedings of the 2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET '12)*, pp. 1096–1100, IEEE, India, March 2012.

[11] N. Gupta and N. Sharma, "Dwt and LSB based Audio Steganography," in *Proceedings of the 2014 International Conference on Reliability, Optimization and Information Technology (ICROIT '14)*, pp. 428–431, IEEE, India, February 2014.

[12] S. S. Verma, R. Gupta, and G. Shrivastava, "A novel technique for data hiding in audio carrier by using sample comparison in DWT domain," in *Proceedings of the 2014 4th International Conference on Communication Systems and Network Technologies (CSNT '14)*, pp. 639–643, IEEE, India, April 2014.

[13] W. Junjie, M. Qian, M. Dongxia, and Y. Jun, "Research for synchronic audio information hiding approach based on DWT domain," in *Proceedings of the 2009 International Conference on E-Business and Information System Security (EBISS '09)*, pp. 1–5, IEEE, China, May 2009.

[14] A. Kanhe and G. Aghila, "DCT based audio steganography in voiced and un-voiced frames," in *Proceedings of the 1st International Conference on Informatics and Analytics (ICIA '16)*, pp. 1–4, ACM Press, India, August 2016.

[15] Z. Zhou and L. Zhou, "A novel algorithm for robust audio watermarking based on quantification DCT domain," in *Proceedings of the 3rd International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP '07)*, pp. 441–444, IEEE, Taiwan, November 2007.

[16] W. Yongqi and Y. Yang, "A synchronous audio watermarking algorithm based on chaotic encryption in DCT domain," in *Proceedings of the 2008 International Symposium on Information Science and Engineering (ISISE '08)*, pp. 371–374, IEEE, China, December 2008.

[17] S. Roy, N. Sarkar, A. K. Chowdhury, and S. M. A. Iqbal, "An efficient and blind audio watermarking technique in DCT domain," in *Proceedings of the 18th International Conference on Computer and Information Technology (ICCIT '15)*, pp. 362–367, IEEE, Bangladesh, December 2015.

[18] B. Ratner, "The correlation coefficient: Its values range between +1/−1, or do they?" *Journal of Targeting, Measurement and Analysis for Marketing*, vol. 17, no. 2, pp. 139–142, 2009.

[19] L. Min, L. Ting, and H. Yu-jie, "Arnold Transform Based Image Scrambling Method," in *Proceedings of the 3rd International Conference on Multimedia Technology (ICMT '13)*, pp. 1309–1316, Publisher Atlantis Press, Guangzhou, China, November 2013.

[20] L. Wu, J. Zhang, W. Deng, and D. He, "Arnold transformation algorithm and anti-Arnold transformation algorithm," in *Proceedings of the 1st International Conference on Information Science and Engineering (ICISE '09)*, pp. 1164–1167, IEEE, China, December 2009.

[21] N. V. Lalitha, S. Rao, and P. V. JayaSree, "DWT - Arnold Transform based audio watermarking," in *Proceedings of the 2013 IEEE Postgraduate Research in Microelectronics and Electronics Asia (PrimeAsia)*, pp. 196–199, IEEE, Visakhapatnam, India, December 2013.

[22] S. Gaur and V. K. Srivastava, "Robust embedding of improved arnold transformed watermark in digital images using RDWT-SVD," in *Proceedings of the 4th IEEE International Conference on Parallel, Distributed and Grid Computing (PDGC '16)*, pp. 563–568, IEEE, India, December 2016.

[23] Z. Zhang, C. Wang, and X. Zhou, "Image watermarking scheme based on Arnold transform and DWT-DCT-SVD," in *Proceedings of the 13th IEEE International Conference on Signal Processing (ICSP '16)*, pp. 805–810, IEEE, China, November 2016.

[24] E. C. Cherry, "Some experiments on the recognition of speech, with one and with two ears," *The Journal of the Acoustical Society of America*, vol. 25, no. 5, pp. 975–979, 1953.

[25] R. Russell, *Cognition: Theory and Practice*, Worth Publishers, 2013.

[26] B. Arons, "A Review of The Cocktail Party Effect," *Journal of The American Voice I/O Society*, vol. 12, pp. 35–50, 1992.

[27] D. E. Broadbent, "Selective listening to speech," in *Perception and Communication*, pp. 11–35, Pergamon Press, 1958.

[28] N. I. Durlach and H. S. Colburn, "Binaural Phenomena," in *Hearing*, pp. 365–466, Elsevier, 1978.

[29] J. Blauert and R. A. Butler, "Spatial hearing: the psychophysics of human sound localization," *The Journal of the Acoustical Society of America*, vol. 77, no. 1, pp. 334-335, 1985.

[30] J. Kassebaum, M. F. Tenorio, and C. Schaefers, "The Cocktail Party Problem: Speech/Data Signal Separation Comparison between Backpropagation and SONN," in *Proceedings of the 2nd International Conference on Neural Information Processing Systems*, pp. 542–549, MIT Press Cambridge, Cambridge, USA, 1990.

[31] S. Haykin and Z. Chen, "The cocktail party problem," *Neural Computation*, vol. 17, no. 9, pp. 1875–1902, 2005.

[32] C. von der Malsburg, "The correlation theory of brain function," in *Models of Neural Networks*, Temporal Aspects of Coding and Information Processing in Biological Systems, pp. 95–119, Springer New York, New York, NY, USA, 1994.

[33] C. von der Malsburg and W. Schneider, "A neural cocktail-party processor," *Biological Cybernetics*, vol. 54, no. 1, pp. 29–40, 1986.

[34] D. L. Wang and G. J. Brown, "Separation of speech from interfering sounds based on oscillatory correlation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 10, no. 3, pp. 684–697, 1999.

[35] G. J. Brown and D. L. Wang, "An oscillatory correlation framework for computational auditory scene analysis," in *Advances in Neural Information Processing Systems 12*, pp. 747–753, MIT Press, 2000.

[36] B. Sagi, S. C. Nemat-Nasser, R. Kerr, R. Hayek, C. Downing, and R. Hecht-Nielsen, "A biologically motivated solution to the cocktail party problem," *Neural Computation*, vol. 13, no. 7, pp. 1575–1602, 2001.

[37] S. Ao, Z. Luo, N. Zhao, and R. Wang, "Blind source separation based on principal component analysis- independent component analysis for acoustic signal during laser welding process," in *Proceedings of the 2010 International Conference on Digital Manufacturing and Automation (ICDMA '10)*, pp. 336–339, IEEE, China, December 2010.

[38] I. T. Jolliffe, *Principal Component Analysis*, Springer Series in Statistics, Springer, New York, NY, USA, 2nd edition, 2002.

[39] M. E. Wall, A. Rechtsteiner, and L. M. Rocha, "Singular value decomposition and principal component analysis," in *A Practical Approach to Microarray Data Analysis*, pp. 91–109, Kluwer Academic Publishers, 2003.

[40] J. Wellhausen, "Audio signal separation using independent subspace analysis and improved subspace grouping," in *Proceedings of the 7th Nordic Signal Processing Symposium (NORSIG '06)*, pp. 310–313, IEEE, Iceland, June 2006.

[41] Q. Cai and X. Tang, "A digital audio watermarking algorithm based on independent component analysis," in *Proceedings of the 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, CISP-BMEI 2016*, pp. 1053–1058, IEEE, China, October 2016.

[42] A. Hyvärinen, "Independent component analysis: recent advances," *Philosophical Transactions of the Royal Society A: Mathematical, Physical & Engineering Sciences*, vol. 371, no. 1984, 20110534 pages, 2012.

[43] A. Hyvärinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 10, no. 3, pp. 626–634, 1999.

[44] H. Helfrich, *Time and Mind II: Information Processing Perspectives*, Hogrefe & Huber Publishers, Cambridge, MA, USA, 2004.

[45] M. F. Casanova and I. Opris, Eds., *Recent Advances on the Modular Organization of the Cortex*, Springer Netherlands, Dordrecht, 2015.

[46] C. Ionescu and R. De Keyser, "Exploring the advantages of blind source separation in monitoring input respiratory impedance during apneic events," *Journal of Control Engineering and Applied Informatics*, vol. 10, no. 3, pp. 53–59, 2008.

[47] Q. Su, Y. Shen, W. Jian, and P. Xu, "Blind source separation algorithm based on modified bacterial colony chemotaxis," in *Proceedings of the 5th International Conference on Intelligent Control and Information Processing (ICICIP '14)*, pp. 354–359, IEEE, Dalian, China, August 2014.

[48] F. A. P. Petitcolas, "Kerckhoffs' Principle," in *Encyclopedia of Cryptography and Security*, H. C. A. van Tilborg and S. Jajodia, Eds., p. 675, Springer, Boston, MA, USA, 2011.

[49] Paar, Christof, and J. Pelzl, "Stream Ciphers," in *In Understanding Cryptography*, p. 35, Springer, Berlin, Heidelberg, Germany, 2010.

[50] J. Laroche and M. Dolson, "New phase-vocoder techniques for pitch-shifting, harmonizing and other exotic effects," in *Proceedings of the 1999 Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 91–94, IEEE, New Paltz, NY, USA.

[51] X.-Y. Wang and H. Zhao, "A novel synchronization invariant audio watermarking scheme based on DWT and DCT," *IEEE Transactions on Signal Processing*, vol. 54, no. 12, pp. 4835–4840, 2006.