Chapter 6

Barcode Based
FeatureMarking Scheme

6.1 Introduction

Aimed at multimedia data, a wide range of effective watermarking algorithms have been proposed and implemented. As far the audio watermarking schemes are concerned, the number of algorithms suggested is very less compared to other multimedia watermarking schemes. The reason behind this is that, the human auditory system (HAS) is far more complex and sensitive than the human visual system (HVS). With electronic communications in this digital era various kinds of disputes may arise in digital audio communication and these disputes may be the denial of authorship of the speech signal, denial of sending or receiving the signal, denial of the time of occurrence etc. The significance of a non-repudiation service arises in these circumstances which guarantee the evidence of an occurrence of a particular event, the time of occurrence and the integrity of the parties
Developing a non-repudiate voice authentication scheme is a challenging task in the context of audio watermarking and this research aims to suggest a digital audio watermarking scheme that ensures authorized and legal use of digital communication, copyright protection, copy protection etc. that helps to prevent such disputes. This chapter introduces a voice signal authentication scheme that employs the FFT towards the embedding and detection schemes and a signal dependent feature for its watermark generation.

6.2 Fourier Analysis

Acoustic signals that are time domain in nature need to get converted into transform domain for retrieving the spectral information. Transforming a signal implies converting time domain information into its frequency domain which demonstrates the details of amplitude and phase components that constitute the signal. Furthermore, there exists an inverse Fourier transform that helps to retrieve the original time domain from its complex frequency domain. The conversion does not result in any loss of its information. It can be concluded that, both time domain and transform domain data are equivalent with different view for a given signal. FFT is computationally efficient method for evaluating the Fourier transform of a digital (acoustic) signal. In order to work with this, the time domain signals must be segmented into finite length blocks of sampled data called frames. Then length of FFT frames is evaluated in terms of sampling rate. In units of time, the duration on which each FFT frame observes the continuous input signal is evaluated. As the duration increases, window size, time between
two consecutive FFT spectrums and FFT data also increase [Bastani and Behbahani 2011; Encyclopedia 2013; Mathworks 1984].

Following figures [figure 6.1, figure 6.2] show the time domain representation of a voice signal plotted using praat5350 win64 tool with a sampling rate of 44100.

![Figure 6.1: Amplitude-time plot](image1)

![Figure 6.2: Spectrum of the signal](image2)

Above spectrum information reveals the following facts:

- Frequency domain:
Lowest frequency: 0 Hz
Highest frequency: 44100 Hz
Total bandwidth: 44100 Hz

Frequency sampling:
- Number of frequency bands (bins): 65537
- Frequency step (bin width): 0.336456298828125 Hz
- First frequency band around (bin center at): 0 Hz

Total energy: 0.18738784 $Pa^2$ sec

The signal sampled at 1000Hz can be plotted as figure 6.3:

- Figure 6.3: Amplitude and frequency plots
6.2. Fourier Analysis

Frequency domain data generated with FFT is complex and is represented in terms of amplitude and phase and linearly spaced in terms of frequency. Time constant which evaluates the length of FFT frames in units of time is inversely proportional to its frequency resolution. FFT spectrum displays the complex frequency data spaced on uniform intervals of its frequency resolution. From this, it is easy to identify the inverse relation between time and frequency domains. Longer FFT size offers higher resolution on spectral data but with slow time response and shorter FFT size offers lower spectral resolution but with faster time response.

Analysis of FFT reveals that the frequency domain spectral data is distributed on a linearly spaced constant frequency interval. A graphical representation employs a frequency axis with equal frequency intervals, denoted in terms of Hz. And it is eminent that, the low-frequency resolution can be maximized by increasing the size of FFT frames but it increases the time duration and thus slower time response. Human auditory system perceives the acoustic pitch or frequency in equal frequency ratios and can be plotted by taking its logarithmic frequency in one axis. While distributing this FFT data in logarithmic frequency axis, it can be observed that the apparent distribution is not constant. Obviously, we require additional methods for viewing the FFT data in a perceptually significant manner to better correlate the graphical data with human hearing [Aeroflex 2005; Hyperphysics 2006].

Intensity of the signal Vs time is represented in the following figure 6.4:
Spectrogram representation is also shown in figure 6.5.

Inverse Fourier transform on frequency domain data recovers the original time domain signal. IFFT is the function employed towards this process. And the ‘symmetric flag’ helps to nullify the numerical inaccuracies or in
other words to zero out the small imaginary components and to recover the original signal. Original time domain signal and the recovered time domain signal behave almost identically and does not reveal any significant difference while playing the audio. In an ideal or real world scenario, it might be able to evaluate the FFT of an acoustic signal for a finite duration or constant time window. And a generalization will be deduced towards the entire signal and may lead to irregularities in the frequency domain and thus to errors in the observed frequency spectrum.

The effect of Fourier transform on an infinite time sequence, on a rectangular window or on a hamming window is described in the above figure 6.5. FFT is able to accurately determine the spectrum and could identify unique steady state frequency of the signal. Infinite time domain signal can be reduced to a finite length frame by using rectangular windowing function and is observed that the energy is dispersed into a single main lobe and a number of side lobes. But the rectangular windowing results in abrupt changes towards its edges or the sidelobes and it can be reduced by applying a gentle time windowing function such as Hamming windowing which in effect reduces the level of side lobes to zero at the edges and also masks the spectral details from neighboring frequency components.

Fourier analysis can be viewed as representing input data using orthogonal basis. Basic element here is the sine function. Fourier decomposition thus represents the input data as a superposition of vibrations on the basis elements. Often, there are a few principal frequencies that account for most of the variability in original data.
6.3 One Dimensional and Two Dimensional Data Codes

Encoding data as its 1-D and 2-D form can be achieved with barcodes, data matrix codes or QR codes. Each of these representations is composed of two separate parts: the finder pattern, which is used by the scanner to locate the symbol and the encoded data itself. The finder pattern defines shape (square or rectangle), size, X-dimension and number of rows and columns in the symbol. The data is then encoded in a matrix within the finder pattern. A data carrier represents data in a machine readable form used to enable automatic reading of the element strings [Scandit 2011; Reinhardt 2011; Encyclopedia 2013]. In this scheme, the data carrier employed is barcode and is discussed below.

A barcode is an optical machine readable representation of data relating to the object to which it is attached. Originally barcodes systematically represent data by varying widths and spacings of parallel lines and may be referred to as linear or one-dimensional (1D). Later they evolved into rectangles, dots, hexagons and other geometric patterns in two dimensions (2D). Although 2D systems use a variety of symbols, they are generally referred to as barcodes as well. Barcodes originally were scanned by special optical scanners called barcode readers. Later, scanners and interpretive software became available on devices including desktop printers and smartphones.

6.4 Proposed Scheme

As described in Chapter 4, original speech signal is divided into frames with length \( l \). Applying FFT on these frames generates its magnitude spectrum.
6.4. Proposed Scheme

The magnitude values obtained are arranged in a matrix which is then transformed into its frequency domain in order to embed the watermark bits.

Let \( V = v(i), 0 \leq i < \text{Length} \) represent a host digital audio signal with \( \text{Length} \) samples. \( FM = FM(i,j), 0 \leq i < M, 0 \leq j < N \) is a binary image to be embedded within the host audio signal and \( FM(i,j) \in 0,1 \) is the pixel value at \( (i,j) \).

6.4.1 Watermark Preparation

Feature extraction module as stated in chapter 4, extracts the signal dependent physical features including the MFCC. This scheme employs MFCC values in the generation of barcode which will be treated as its signal dependent watermark.

Quantized MFCC values are streamed in desired format and submit as input to an online barcode generator. Obtained barcode holds all quantized feature vectors of the recorded voice signal. An example of the generated barcode is shown in figure 6.6. Watermark in the proposed scheme is also stated as the FeatureMark.

![Figure 6.6: Sample barcode](image)

6.4.2 FeatureMark Embedding

Obtained FeatureMark bits are first scrambled using Arnold transform which dissipate pixel space relationship of the FeatureMark image and thus helps to improve robustness criteria of the watermarked signal. Scrambled
FeatureMark image will then be converted into a 1-dimensional sequence of 1s and 0s (binary digits).

Let $FM = FM(i, j), 0 \leq i \leq M, 0 \leq j \leq N$ represents the original FeatureMark image.

Applying Arnold transform results in a scrambled structure which can be represented as figure 6.7

$FM1 = FM1(i, j), 0 \leq i \leq M, 0 \leq j \leq N.$

Figure 6.7: Arnold transformed barcode

Number of iterations selected to generate the Arnold transform is 2. Scrambled structure is then converted into a sequence of 1s and 0s as follows:

$FM2 = fm2(k) = FM1(i, j), 0 \leq i \leq M, 0 \leq j \leq N,$  

$k = i \times N + j, fm2(k) \in 0, 1$ Finally, each bit of the FeatureMark data is mapped into the signal frames by transforming it using FFT. Embedding scheme employs the following condition towards mapping of each FeatureMark bits.

Let $f_1, f_2...f_n$ be the Fourier coefficients obtained. These values are sorted in an ascending order, let it be $f'_1, f'_2...f'_n$ and take first 8 peak coefficients such as $f'_{n}, f'_{n-1}...f'_{n-7}$ towards FeatureMark embedding. Thus, each frame can hold 8 FeatureMark bits.
6.4. Proposed Scheme

FeatureMark Embedding Scheme (represented in figure 6.8) is presented in the following algorithm:

**Algorithm 1 FeatureMark embedding**

**Inputs**
1. Original speech signal
2. FeatureMark

**Output**
1. FeatureMarked speech signal

**Steps**
1. Prepare the barcode as FeatureMark using an online barcode generator
2. Generated FeatureMark is a two-dimensional sequence represented as:
   - \( FM = FM(i, j), 0 \leq i < M, 0 \leq j < N \) where \( FM(i, j) \in 0, 1 \) is the pixel value at \((i, j)\)
3. Arnold transform is applied to scramble the FeatureMark image denoted as:
   - \( FM1 = FM1(i, j), 0 \leq i \leq M, 0 \leq j \leq N \)
4. Scrambled structure will then be converted into a one-dimensional sequence of 1s and 0s represented as:
   - \( FM2 = fm2(k) = f1(i, j), 0 \leq i \leq M, 0 \leq j \leq N, k = i \times N + j, fm2(k) \in 0, 1 \)
5. Convert the speech signal into speech samples with a sampling rate of 44100
6. Apply FFT to the original speech signal and the FFT frames are segmented into non-overlapping subsegments

Sampling rate in context of signal processing refers to the average number of samples per second to represent the event digitally. In most digital audio, common sampling rate/frequency is 44,100 Hz as it allows for a 2.05 kHz transition band which is used in compact discs. Sampling rate
44,100 Hz was originated in the late 1970s with PCM adaptors where digital audio was recorded on video cassettes (the Sony PCM-1600 (1979) and subsequent models in this series). Nyquist-Shannon sampling theorem suggest ideal sampling rate to be greater than twice the maximum frequency one wishes to reproduce. As hearing range of human ears is roughly 20 Hz to 20,000 Hz the ideal sampling rate therefore had to be greater than 40 kHz [Encyclopedia 2013].

Algorithm 1 Featuremark embedding (continued)

7: Do for each subsegment

- Calculate the magnitude and phase spectrum of each subsegment using the FFT: Let $f_1, f_2, \ldots, f_n$ be the magnitude coefficients obtained
- Sort the magnitude coefficients in ascending order of their values: Let it be $f'_1, f'_2, \ldots, f'_n$
- Energy entropy of each frame is then calculated using the following equation 6.1
  \[ I_j = - \sum_{i=1}^{k} \sigma_i^2 \log_2 \sigma_i^2 \] (6.1)
- Obtain the first 8 peaks of these magnitude coefficients, let it be $f''_1, f''_2, \ldots, f''_{n-7}$
- Insert watermark bits into these coefficients by using the following condition:
  1. For embedding a 1 to $f''_i$, make $f''_i = f''_i + \alpha$
  2. For embedding a 0 to $f''_i$, make $f''_i = f''_i - \alpha$, where $\alpha = 0.0001$
  3. Repeat embedding
- Perform the same sequence on the subsequent segments till all the watermark bits are embedded

8: Inverse FFT is applied to convert frequency domain back to original time domain to form the FeatureMarked speech signal
Here, $f_i$ is a magnitude coefficient into which a watermark is embedded, $f_{mi}$ is a watermark bit to be inserted into $f_i$, $\alpha$ is a scaling factor, $f''_i$ is an adjusted magnitude coefficient. Energy entropy of each frame is evaluated to confirm that the embedding does not destroy energy distribution of the signal.

![Watermark embedding scheme](image)

Figure 6.8: Watermark embedding scheme

### 6.4.3 Repeat Embedding

Embedding a single FeatureMark for an entire acoustic signal does not guarantee any robustness against common signal manipulations as well as desynchronization attacks. Excellent robustness is a key criterion of every
watermarking scheme. Robustness against these attacks can be improved by embedding the FeatureMark more than once in a particular signal. The term coined to demonstrate it is the ‘repeat embedding’ and is employed to reduce the number of ‘bit errors’ aroused due to these attacks. Let ‘r’ denote the number of repetitions and $FM_l$ denote the number of bits of the FeatureMark that need to be embedded in the signal. For example, if r is 1, only $FM_l$ bits are embedded and if r is 5, five times $FM_l$ bits are embedded. In other words, a total of $5 \times FM_l$ bits are embedded. That is, each FeatureMark bit sequence is repeatedly embedded r times in an acoustic signal. The average number of repetitions used in our case is three. Thus, the precision or accuracy of the extracted FeatureMark can be improved to a great extent.

Let the generated watermark bits hold 2000 bits for signal duration of 1 minute then theoretical evaluation reveals the need for 5 seconds to embed the watermark and gives a prospect to embed the watermark 12 times. From the above equation deriving capacity, actual capacity of the proposed watermarking scheme is 33.33 bps.

### 6.4.4 Signal Reconstruction

After embedding r times the FeatureMark bits on the acoustic signal, an inverse FFT is applied to produce the speech signal. That is, modified spectra of the time domain signal is firstly converted using inverse FFT. Then, combining each of the non-overlapping or overlapping window-frame series respectively helps to reconstruct the FeatureMarked time domain signal.

Signal reconstruction can be described as follows:
6.4. Proposed Scheme

- Embedding process preceded by the computation of non-overlapping Hamming windowed FFT. The original speech signal denoted as \( v_1(n) \) and the results obtained is a spectra denoted as \( V_1(\omega) \).

- Obtained spectra \( V_1(\omega) \) is then manipulated to embed the FeatureMark bits which in turn results in the modified spectra denoted as \( V'_1(\omega) \).

- At the time of speech signal reconstruction convert the \( V'_1(\omega) \) to \( v'_1(n) \) using inverse FFT. Then combine \( v'_1(n) \), the non-overlapping Hamming window \( w_1(n) \) and the overlapping Hamming window \( w_2(n) \) together to reconstruct the original speech domain signal.

From this, it is understood that FFT transform on the overlap regions is not ideal for embedding the FeatureMark bits since it may be thrown away during signal reconstruction procedure. It can be summarized as follows: Let \( \text{fft}(m) \) provides spectrum of the frame ‘m’ which is modified for embedding the FeatureMark bits; its subsequent frame ‘m+1’ is also get modified to satisfy the embedding criteria because m and m+1 are overlapping frames. That is, spectral modification applied to a particular frame will get distorted by spectral modifications applied to its subsequent overlapping frame. This will result in reduced accuracy of the extracted FeatureMark bits as the modification on each frame represents the embedded FeatureMark bits. In short, it can be stated that, for an FFT based FeatureMarking scheme the spectrum of each frame of the original speech signal is computed using FFT and the magnitudes of this spectrum is arranged into a matrix of order \( M \times N \).
6.4.5 Watermark Detection

FeatureMark detection procedure employed in this scheme does not need the original speech signal but it needs the length or number of embedded mark bits. The detection process mainly involves identification of the presence of embedded mark bits.

In order to detect the FeatureMark bits, spectrum of each frame of the FeatureMarked signal is evaluated and the obtained magnitudes are arranged into matrix of order $M' \times N'$.

6.4.6 Digital Watermark Extraction

Once the presence of the FeatureMark is detected, next step is to extract (shown in figure 6.9) the FeatureMark bits by employing the conditions mentioned in the embedding module.

But in this scheme, entire signal should be traversed to identify the presence of FeatureMark as well to extract the mark bits. According to this condition, it extracts the FeatureMark bits and the number of bits extracted should match with that of the embedded FeatureMark bits.
Algorithm 2 FeatureMark extraction

Inputs
1: FeatureMarked speech signal

Output
1: FeatureMark

Steps
1: The FeatureMarked speech signal is transformed into FFT domain.

Figure 6.9: Watermark extraction scheme
Algorithm 2 FeatureMark extraction (continued)

2: Do for each subsegment
   • Calculate the magnitude and phase spectrum of each subsegment using the FFT:
   • Let $f_1, f_2, \ldots, f_n$ be the magnitude coefficients obtained
   • Sort the magnitude coefficients in ascending order of their values: Let it be $f_1', f_2', \ldots, f_n'$
   • Energy entropy of each frame is then calculated using the following equation 6.2
     \[
     I_j = - \sum_{i=1}^{k} \sigma_i^2 \log_2 \sigma_i^2
     \] (6.2)
   • Obtain the first 8 peaks of these magnitude coefficients, let it be $f_n'-7, \ldots, f_n'-2, f_n'-1, f_n'$
   • Extract the watermark bits from these coefficients by using the following condition:
     1. Extract a ‘1’ for $f_n'' = f_n' + \alpha$
     2. Extract a ‘0’ $f_n'' = f_n' - \alpha$, where $\alpha = 0.0001$
     3. Perform the same sequence on the subsequent segments till all the watermark bits are extracted

3: Convert the one-dimensional sequence into a two-dimensional scrambled structure
4: Apply the anti-Arnold transformation on this scrambled structure to obtain the original FeatureMark image
5: Repeat Steps 2, 3 and 4 upto $r$ times
6: Inverse FFT is applied to convert the frequency domain back to its time domain (if needed)
6.5 Experimental Results

In order to carry out the watermarking as well as to evaluate the performance of this scheme, various tests were conducted. These tests include imperceptibility tests, robustness tests and capacity tests. Experimental set up for this proposed scheme is illustrated as follows: 50 audio signals of around 10 members with a sampling rate of 44,100 times per second. The Matlab version R2009b is used in embedding and detection schemes. Music editor sound recorder is employed for inducing some of the common signal manipulations and desynchronization attacks in the signal.

Audio signals in the test are based on Malayalam speech signals with 16 bits/sample, 44.1 kHz sample rates. Signal duration vary for each signal and is in the range of 2 - 300s. Number of frames employed towards signal processing depends on the length of the signal with a frame rate of 100. From the experiments conducted it is obvious that the embedding scheme works fine with around 50 samples. Prepared FeatureMarks: Barcodes (figures 6.10 & 6.11)

Figure 6.10: Sample 1
Experiments were conducted on both single channel and multi-channel speech signals and the results are plotted in following figures - figure 6.12 & figure 6.13:

Figure 6.12: Single channel - original and FeatureMarked speech signal

Figure 6.13: Multi-channel - original and FeatureMarked speech signal
6.5. Experimental Results

Transparency Tests

Transparency of this proposed method is evaluated by employing the subjective listening tests. Selection of listening panel is based on a set of expert as well as non-expert listeners. In this scheme a set of non-expert listeners and used and the panel includes representative of the general population - fellow research scholars and colleagues. The final decision is based on a 5-point grade scale [ITU-R 2002] presented in table 6.1.

Table 6.1: 5-Point grade scale

<table>
<thead>
<tr>
<th>Quality</th>
<th>Impairment</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 Excellent</td>
<td>5 Imperceptible</td>
</tr>
<tr>
<td>4 Good</td>
<td>4 Perceptible, but not annoying</td>
</tr>
<tr>
<td>3 Fair</td>
<td>3 Slightly annoying</td>
</tr>
<tr>
<td>2 Poor</td>
<td>2 Annoying</td>
</tr>
<tr>
<td>1 Bad</td>
<td>1 Very annoying</td>
</tr>
</tbody>
</table>

And the result obtained for this scheme is given below:

Table 6.2: Imperceptibility criteria

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Imperceptibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFT &amp; Barcode</td>
<td>Good</td>
</tr>
</tbody>
</table>

Robustness Tests

Robustness of this scheme is evaluated by performing common signal manipulations or desynchronization attacks on the FeatureMarked signals.

- Measure of strength of watermarking scheme against common signal processing functions is described below:
Table 6.3: Common signal manipulations

<table>
<thead>
<tr>
<th>Manipulation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise Addition:</td>
<td>Added white Gaussian noise with an SNR of 50</td>
</tr>
<tr>
<td>Silence Addition:</td>
<td>A silence of 100 ms duration is inserted at the beginning of the FeatureMarked signal</td>
</tr>
<tr>
<td>Echo Addition:</td>
<td>Added an echo with a delay of 200 ms</td>
</tr>
<tr>
<td>Re-sampling:</td>
<td>Down sampled to frequencies 22.05 kHz and then up sampled to its original 44.1 kHz</td>
</tr>
<tr>
<td>Re-quantization:</td>
<td>Signal with 16-bit quantized to 8-bit and then back to its original 16-bit</td>
</tr>
<tr>
<td>Low-pass Filtering:</td>
<td>Done with cut off frequencies 10 kHz and 200 Hz</td>
</tr>
<tr>
<td>Band-pass Filtering:</td>
<td>Done with cut off frequencies 10 kHz and 200 Hz</td>
</tr>
</tbody>
</table>

The procedure used for these common signal manipulations such as silence addition, echo addition, low-pass and band-pass filtering are performed using the freely available music editor tool. Other signal manipulations are performed in Matlab.

- Measure of strength of watermarking scheme against desynchronization attacks is shown below:
6.5. Experimental Results

<table>
<thead>
<tr>
<th>Table 6.4: Desynchronization attacks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Amplitude Variation:</strong> Signal is amplified to its double as well as half</td>
</tr>
<tr>
<td><strong>Pitch Shifting:</strong> Involve frequency fluctuation to the signal</td>
</tr>
<tr>
<td><strong>Cropping:</strong> Performed randomly at different positions of the signal</td>
</tr>
<tr>
<td><strong>Time-scale Modification:</strong> Watermarked voice signal was lengthened (slow-down) and shortened (double speed)</td>
</tr>
</tbody>
</table>

The procedure used for desynchronization attacks such as amplification to double volume and half volume, pitch change, speed changes and random cropping are done using the Music editor software.

Performance of the proposed FeatureMarking scheme is analyzed by evaluating bit error rate (BER) of the extracted FeatureMark to the actual or embedded FeatureMark. BER is a unitless performance measure which is often expressed in %. Evaluation is performed on both single channel and multi-channel audio signals. In case of multi-channel signals, both channels are not employed for embedding the watermark bits. Instead, the two channels are added up so as to compromise the properties of the original audio signal. Results obtained are as follows.

Experiment # 1: Performed on single channel sounds with around 27 signals and was found successful on all the signals.
Table 6.5: Robustness test for signal manipulations (in BER×100%)

<table>
<thead>
<tr>
<th>Original Signal</th>
<th>Attacks Free</th>
<th>Noise Addition</th>
<th>Silence Addition</th>
<th>Echo Addition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Audio 1</td>
<td>0.0000</td>
<td>0.2000</td>
<td>0.1000</td>
<td>0.4200</td>
</tr>
<tr>
<td>Host Audio 2</td>
<td>0.0000</td>
<td>0.1000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>Host Audio 3</td>
<td>0.0000</td>
<td>0.1100</td>
<td>0.1000</td>
<td>0.2100</td>
</tr>
<tr>
<td>Host Audio 4</td>
<td>0.0000</td>
<td>0.1100</td>
<td>0.2000</td>
<td>0.4000</td>
</tr>
</tbody>
</table>

Table 6.6: Robustness test for signal manipulations (in BER×100%)

<table>
<thead>
<tr>
<th>Original Signal</th>
<th>Re-Sampling</th>
<th>Re-Quantization</th>
<th>Low-pass filtering</th>
<th>Band-pass Filtering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Audio 1</td>
<td>0.4200</td>
<td>0.2100</td>
<td>0.3200</td>
<td>0.2100</td>
</tr>
<tr>
<td>Host Audio 2</td>
<td>0.4500</td>
<td>0.1100</td>
<td>0.4100</td>
<td>0.1500</td>
</tr>
<tr>
<td>Host Audio 3</td>
<td>0.4800</td>
<td>0.1000</td>
<td>0.1200</td>
<td>0.1300</td>
</tr>
<tr>
<td>Host Audio 4</td>
<td>0.4000</td>
<td>0.4100</td>
<td>0.2200</td>
<td>0.2500</td>
</tr>
</tbody>
</table>

Table 6.7: Robustness test for desynchronization attacks (in BER×100%)

<table>
<thead>
<tr>
<th>Original Signal</th>
<th>Amplitude Variation</th>
<th>Pitch Shifting</th>
<th>Random Cropping</th>
<th>Time-Scale modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Audio 1</td>
<td>0.4000</td>
<td>0.3900</td>
<td>0.2500</td>
<td>0.4200</td>
</tr>
<tr>
<td>Host Audio 2</td>
<td>0.3900</td>
<td>0.3800</td>
<td>0.3600</td>
<td>0.3700</td>
</tr>
<tr>
<td>Host Audio 3</td>
<td>0.4100</td>
<td>0.4100</td>
<td>0.4000</td>
<td>0.3900</td>
</tr>
<tr>
<td>Host Audio 4</td>
<td>0.4200</td>
<td>0.4200</td>
<td>0.3900</td>
<td>0.3500</td>
</tr>
</tbody>
</table>

Experiment # 2: Performed on multi-channel sounds with around 23 signals and was found successful on all the signals.
6.5. Experimental Results

Table 6.8: Robustness test for signal manipulations (in BER \times 100\%)

<table>
<thead>
<tr>
<th>Original Signal</th>
<th>Attacks Free</th>
<th>Noise Addition</th>
<th>Silence Addition</th>
<th>Echo Addition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Audio 5</td>
<td>0.0000</td>
<td>0.1100</td>
<td>0.1200</td>
<td>0.4200</td>
</tr>
<tr>
<td>Host Audio 6</td>
<td>0.0000</td>
<td>0.1100</td>
<td>0.1100</td>
<td>0.4000</td>
</tr>
<tr>
<td>Host Audio 7</td>
<td>0.0000</td>
<td>0.2100</td>
<td>0.1100</td>
<td>0.4000</td>
</tr>
<tr>
<td>Host Audio 8</td>
<td>0.0000</td>
<td>0.2100</td>
<td>0.1200</td>
<td>0.4100</td>
</tr>
</tbody>
</table>

Table 6.9: Robustness test for signal manipulations (in BER \times 100\%)

<table>
<thead>
<tr>
<th>Original Signal</th>
<th>Re-Sampling</th>
<th>Re-Quantization</th>
<th>Low-pass filtering</th>
<th>Band-pass Filtering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Audio 5</td>
<td>0.4100</td>
<td>0.2100</td>
<td>0.3800</td>
<td>0.4000</td>
</tr>
<tr>
<td>Host Audio 6</td>
<td>0.4200</td>
<td>0.2200</td>
<td>0.4100</td>
<td>0.3500</td>
</tr>
<tr>
<td>Host Audio 7</td>
<td>0.3700</td>
<td>0.2100</td>
<td>0.3700</td>
<td>0.3600</td>
</tr>
<tr>
<td>Host Audio 8</td>
<td>0.3800</td>
<td>0.4000</td>
<td>0.3800</td>
<td>0.4500</td>
</tr>
</tbody>
</table>

Table 6.10: Robustness test for desynchronization attacks (in BER \times 100\%)

<table>
<thead>
<tr>
<th>Original Signal</th>
<th>Amplitude Variation</th>
<th>Pitch Shifting</th>
<th>Random Cropping</th>
<th>Time-Scale Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Audio 5</td>
<td>0.4100</td>
<td>0.4000</td>
<td>0.3600</td>
<td>0.3700</td>
</tr>
<tr>
<td>Host Audio 6</td>
<td>0.3900</td>
<td>0.3900</td>
<td>0.4100</td>
<td>0.3600</td>
</tr>
<tr>
<td>Host Audio 7</td>
<td>0.4200</td>
<td>0.4100</td>
<td>0.3900</td>
<td>0.4100</td>
</tr>
<tr>
<td>Host Audio 8</td>
<td>0.3900</td>
<td>0.4100</td>
<td>0.3800</td>
<td>0.4200</td>
</tr>
</tbody>
</table>

Watermark recovery rate has been evaluated using the equation \((1 - BER) \times 100\%\). And the above results reveal that watermarked signals are vulnerable to common signal manipulations and desynchronization attacks.
An enhanced scheme that can withstand the common signal processing and desynchronization attacks is suggested in the upcoming chapter.

Average of the recovery rate obtained for some signals are plotted below - figure 6.14:

![Figure 6.14: Average recovery rate](image)

**Capacity Tests**

Capacity or data payload demonstrates the amount of information that can be embedded and recovered in the audio stream. It can be evaluated using the equation 6.3.

\[ C = \frac{M}{L} \text{bps}, \]  

(6.3)

where \( M \) refers to the number of watermark bits and \( L \) the length of the audio signal.
6.5. Experimental Results

Theoretical evaluation demonstrates that this scheme holds 400 watermark bits per second because we are selecting 8 coefficients from each of the frames and a total of 50 frames have been obtained for a second. But the actual calculation provides a capacity of 66.67 bps.

6.5.1 Non-Repudiation Services

Experimental results reveal that objective of the proposed scheme, non-repudiation service [Onieva, Zhou, and Lopez 2004; Zhou and Lam 1999; Zhou and Gollmann 1997b; Zhou and Gollmann 1996a; Coffey and Saidha 1996; Onieva et al. 2004; Steinebach et al. 2001; Zhou and Gollmann 1997a; Zhou and Gollmann 1996b; Schneider 1998] could be achieved to a great extent. Realization of voice authentication scheme that guarantees non-repudiation is done by the following sequence of steps: First a signal dependent FeatureMark is developed using the physical features of signal. Then the FeatureMark is embedded by utilizing suitable FFT coefficients by transforming the signal to its frequency domain spectra. After embedding the FeatureMark bits, inverse transform IFFT is applied to convert the speech signals back to its time domain.

FeatureMarked signal is then transferred to the recipient and at recipient side, the signal is again transformed using FFT transform. Then for identifying the presence of watermark, entire signal is traversed from one end to the other end till a FeatureMark occurrence is detected. When the presence of the FeatureMark is detected, the bits are extracted and then combined to make the original FeatureMark. All the steps performed in the embedding module are reversed to make up the FeatureMark at the recipient side.

FeatureMark at the recipient side can be compared to the original one to confirm the authenticity as well as to guarantee non-repudiation. Feature
extraction module can also be conducted at the receiving side to create the same FeatureMark in case of any disputes.

Thus, it can be assured that the proposed scheme helps in achieving an authentic communication scheme for transferring audio or speech signals without causing any ownership disputes. Important characteristics of a non-repudiation scheme such as creating and storing the evidence for sender and receiver, fairness, timeliness and confidentiality could be accomplished without any difficulty.

### 6.6 Summary

In this scheme, a novel watermarking technique is presented using FFT for authenticating speech signals that can guarantee non-repudiation. Experimental results reveal that this scheme can be employed in authentication or copyright protection of audio sound. Obtained results also indicate that the proposed scheme has not compromised audibility of the signal. Experimental results also guarantees tolerance to various signal processing and desynchronization attacks such as noise addition, silence addition, echo addition, filtering, compression, cropping, amplitude variation and time-scale modification. The only drawback identified in this scheme is that, in order to extract the FeatureMark bits the entire signal should be traversed from one end to the other end which eventually increase the time it takes to detect the presence of the mark.