Robust Spread Spectrum Based Patchwork Audio Watermarking

Avni Gupta[1] Rakesh Sharma[2]
[1,2]M.Tech
[1,2]Department of Computer Engineering
[1,2]RCEW, Jaipur, Rajsthan

Abstract— Digital Watermarking refers to hiding digital data in digital content with intent to enable copyright protection and illegal distribution. Digital audio watermarking deals with embedding digital information in an imperceptible manner in the host audio signal. Audio watermarking involves much more complex techniques to embed watermark in view of superiority of Human Auditory System (HAS) over Human Visual System (HVS). Spread Spectrum watermarking embeds the watermark in frequency domain, thereby providing a much more robust and invisible watermarking scheme as compared to time domain watermarking, which is comparatively fragile. It turns out that digital watermark embedding in audio signals in any domain results in audible disturbances in the original audio signal. To compensate for these unwanted disturbances in audio files, audio watermarking techniques are to be augmented with Frequency Masking (FM) techniques. It turns out that PAFM creates an imbalance in the number of positive and negative chips that are used to modulate the watermark, and consequently, it introduces error in correlation test. This paper proposes a robust audio watermarking technique based on the patchwork algorithm. Peak Signal to Noise Ratio (PSNR) is evaluated as a function of embedding capacity and robustness. PAFM is analyzed as a function of the threshold limit for embedding of watermark bits. The proposed method provides an optimized way to embed watermark using spread spectrum technique, with a detector of minimal complexity and maximum achievable efficiency. MABLAB is used to simulate the technique and results are derived. PSNR is used as a quality metric for evaluation of the results.

Key words: Digital Audio Watermarking, Spread Spectrum, PN sequences, Correlation Analysis, HAS, PAFM

I. INTRODUCTION

Watermarking Digital content forms the basis of Intellectual Property Rights (IPR) as millions of unauthorized copies of digital content can be created with existing systems. Watermark refers to a specific information that can be added in a digital content to later provide a proof of ownership of the digital document. Watermarking algorithms were primarily developed for digital images and video data and the research in the field of audio watermarking started slightly later. This work primarily deals with audio watermarking in frequency domain using Spread Spectrum (SS) Technique. SS spreads and modulates the watermark bits which are later embedded in the frequency domain of the audio signal, to give watermarked audio signal.

With the growth of the Internet, unauthorized copying and illegal distribution of digital media has been easier to the extent such that this can be done even with the cheapest commodity computer. As a result, the music industry claims a multibillion dollar annual revenue loss due to piracy, which is likely to increase due to peer-to-peer file sharing Web communities. One source of hope for copyrighted content distribution on the Internet lies in technological advances that would provide ways of enforcing copyright in client-server scenarios. Traditional data protection methods such as scrambling or encryption cannot be used since the content must be played back in the original form, at which point, it can always be rerecorded and then freely distributed. A promising solution to this problem is marking the media signal with a secret, robust, and imperceptible watermark (WM). The media player at the client side can detect this mark and consequently enforce a corresponding e-commerce policy.

In this paper, Direct-Sequence Spread Spectrum (DSSS) technique is used to embed Watermarks and techniques are developed to improve the effectiveness of their embedding and detection in audio. WM robustness is enabled using repetition coding for prevention against de-synchronization attacks and psycho-acoustic frequency masking (PAFM). A digital watermark is modulated through PN sequences and embedded in frequency component of the audio file using Patchwork Algorithm. Sound Fidelity is evaluated as a function of embedding capacity and robustness. Repetition embedding in the QIM samples of audio is done to prevent de-synchronization attacks. PSNR is used as quality metric for the evaluation of the results.

Digital Data can be shared by multiple users, over a distributed network, and managed for long period of time without any damage. Consequently, the copyright protection problem arises as unauthorized copying and distribution of digital data are simple even with a low cost commodity computer. As a result, a technique called digital watermarking is introduced to protect the ownership of these contents. Digital Audio Watermarking is a technology to hide copyright information in an audio file without making the information being audible to the listener, and by least affecting the audio quality of the original file. Among all the techniques of audio watermarking which are developed to date, the spread spectrum audio watermarking technique is best in its class by being robust to any kind of intentional or non-intentional attack. Also, it provides a technique of correlation based watermark detection, thereby providing blind watermarking technique in which the original file is not required at the detector for watermark detection or extraction. In the proposed work, a watermark embedding technique is proposed based on spread spectrum and patchwork algorithm which provides a blind watermark technique.

II. RESEARCH APPROACH

In this paper, text data is used as watermark. This data is first converted into decimal numbers through ASCII codes which are then transformed into 7 bit binary numbers. Thus, a binary string of zero and one is obtained. Corresponding to each one or zero in the watermark string, a PN sequence generated by LFSR is embedded to the audio signal using Patchwork algorithm, in frequency domain. A special class of PN sequence, called m sequence is used for the reason that these have perfect two level correlation values (corresponding to auto and cross correlation), for the embedding of zero or one. PSNR is evaluated as a function of tradeoff between embedding capacity and robustness. Results are compared with the benchmark standards and conclusions are drawn.

III. PROPOSED WORK

A. *Algorithm for watermark embedding:*

The embedding of watermark message of length M is done by diving the audio signal into L frames. with each frame consisting of N PCM samples, which is usually a power of two. Using the Fast Fourier Transformation (FFT), the spectrum of Fourier magnitude coefficients e(n,t) is calculated, where n =1,2,...N/2 denotes the band index and t = 1,2,....L.

In the process of embedding, the embedder selects N pairs of values of the digital cover signal pseudo randomly. The name pseudo-random is given as the sequences appears to be random sequences but actually are deterministic and can be generated using a key. These pseudorandom sequence can be generated using characteristic equation or can be generated using Linear Feedback Shift Register (LFSR) circuit. In the former case, the key decides the structure of the equation, and in the latter, the key decides the circuit as well as initial content of the registers.

For every message bit $m_i \epsilon\{0,1\}$, two subsets A_i and B_i (i = 1,2,.....M) containing R coefficients, each are selected from the complete set of coefficients. This selection is done pseudo randomly depending upon the secret key. It is assumed that for unmarked cover data, the FFT coefficients from different time steps and different band index are sufficiently independent and identically distributed that the subsets contain sufficiently many elements. Then the partial sums

$$S_A = \frac{1}{R}\sum_{e\epsilon A_i} e(n,t) \qquad i = 1 \ldots\ldots M \qquad \text{equation 3.2}$$

$$S_B = \frac{1}{R}\sum_{e\epsilon B_i} e(n,t) \qquad i = 1 \ldots\ldots M \qquad \textit{equation } 3.2$$

are approximately identical. As a consequence, the random variable

$$S_i = S_A - S_B \qquad \textit{equation 3.2}$$

and is close to zero.

For embedding a message bit 1, the coefficients A_i are slightly increased while the coefficients in B_i are slightly decreased, and vice versa for embedding zero. The degree of modification is controlled by psycho acoustic frequency masking.

Typically 8000 samples are taken every second thus, each sample is of length .000004 sec in a typical pulse code modulation scheme of an audio signal.

Consider the following hypothetical example for watermark embedding scheme illustration. In this example, L = 4 frames are considered, with N =8 samples per frame. Also R = 8 magnitude coefficients are considered for watermark embedding purpose.

In the table given below, PN1 and PN2 are considered for a single bit embedding. Total 16 values are considered for each of PN1 and PN2 as 16 in place of 8 magnitude coefficients needs to be changed so as to keep the symmetry of Fourier Transform.

The following table will illustrate the procedure.

Time	**Amplitude**	**PCM samples**	**Frame**	**FFT**	**Magnitude**	**N-1**	**N-2**
0	55	110111	Frame #1	897	897		
0.000125	57	111001	Frame #1	-3.39155300195564-2.25618918841821i	4.073453255		
0.00025	59	111011	Frame #1	-0.949747468305841-7.53553390593273i	7.595149209		2
0.000375	58	111010	Frame #1	-0.667833362043576-14.0654866308133i	14.08133217		
0.0005	52	110100	Frame #1	-7.99999999999999+11i	13.60147051		
0.000625	57	111001	Frame #1	-10.9890208874488-6.89391375555946i	12.97245647	1	
0.00075	55	110111	Frame #1	8.94974746830583+0.464466094067254i	8.96179159		2
0.000875	59	111011	Frame #1	3.04840725144799-15.0846163131644i	15.38955607	1	
0.001	58	111010	Frame #1	7	7		
0.00112	51	110011	Frame #1	3.04840725144803+15.0846163131644i	15.38955607	1	

5							
0.00125	58	111010	Frame #1	8.94974746830583-0.464466094067267i	8.96179159		2
0.001375	59	111011	Frame #1	-10.9890208874488+6.89391375555947i	12.97245647	1	
0.0015	57	111001	Frame #1	-8.00000000000001-11i	13.60147051		
0.001625	52	110100	Frame #1	-0.667833362043543+14.0654866308133i	14.08133217		
0.00175	58	111010	Frame #1	-0.949747468305822+7.53553390593275i	7.595149209		2
0.001875	52	110100	Frame #1	-3.39155300195562+2.25618918841823i	4.073453255		
0.002	60	111100	Frame #2	885	885		
0.002125	59	111011	Frame #2	14.584143415306-4.36063241053785i	15.22210085	1	
0.00225	58	111010	Frame #2	5.46446609406725-12.5355339059327i	13.67479433		2
0.002375	56	111000	Frame #2	21.0477692648187-3.68812011678592i	21.36845388		
0.0025	52	110100	Frame #2	-5.99999999999999+9i	10.81665383	1	
0.002625	58	111010	Frame #2	-8.46198282719178-5.10233367915898i	9.881242955		
0.00275	53	110101	Frame #2	12.5355339059327+5.46446609406726i	13.67479433		2
0.0	5	11	Fra	0.830070147067	5.83		

028755	4	0110	me #2	089-5.77484597291094i	4197671		
0.003	53	110101	Frame #2	-5	5		
0.003125	50	110010	Frame #2	0.830070147067097+5.77484597291095i	5.834197671		
0.00325	58	111010	Frame #2	12.5355339059327-5.46446609406727i	13.67479433		2
0.003375	60	111100	Frame #2	-8.46198282719178+5.10233367915902i	9.881242955	1	
0.0035	52	110100	Frame #2	-6.00000000000001-9i	10.81665383		
0.003625	51	110011	Frame #2	21.0477692648187+3.68812011678588i	21.36845388		
0.00375	54	110110	Frame #2	5.46446609406728+12.5355339059327i	13.67479433		2
0.003875	57	111001	Frame #2	14.584143415306+4.36063241053784i	15.22210085	1	
0.004	50	110010	Frame #3	869	869		
0.004125	58	111010	Frame #3	4.93076080390519-5.77484597291095i	7.593500386		2
0.00425	53	110101	Frame #3	-24.778174593052-9.84924240491749i	26.66393655	1	
0.004375	59	111011	Frame #3	-1.46779471420834-5.102333679159i	5.309258922		
0.004	60	1111	Frame	12+5i	13		

5		00	#3				
0.004625	54	110110	Frame #3	-11.6032730976571-3.6881201167859i	12.17531012	1	
0.00475	53	110101	Frame #3	-9.22182540694801-19.8492424049175i	21.88685651		2
0.004875	51	110011	Frame #3	-3.85969299203972-4.36063241053785i	5.823430691		
0.005	53	110101	Frame #3	-1	1		
0.005125	52	110100	Frame #3	-3.85969299203971+4.36063241053785i	5.823430691		
0.00525	50	110010	Frame #3	-9.22182540694796+19.8492424049175i	21.88685651		2
0.005375	59	111011	Frame #3	-11.6032730976571+3.68812011678591i	12.17531012	1	
0.0055	60	111100	Frame #3	12-5i	13		
0.005625	51	110011	Frame #3	-1.46779471420834+5.10233367915899i	5.309258922		
0.00575	55	110111	Frame #3	-24.778174593052+9.84924240491751i	26.66393655	1	
0.005875	51	110011	Frame #3	4.9307608039052+5.77484597291094i	7.593500386		2
0.006	55	110111	Frame #4	892	892		
0.00612	58	1110	Frame	9.36396103067893-5.912433899320	11.07432		2

5		10	#4	38i	349		
0.00625	56	111000	Frame #4	-14.3137084989848+5.00000000000001i	15.16186832	1	
0.006375	60	111100	Frame #4	-3.36396103067894-14.8760533716144i	15.25166213		
0.0065	54	110110	Frame #4	-2+4i	4.472135955		2
0.006625	59	111011	Frame #4	-3.36396103067894-9.46183980924129i	10.04204393	1	
0.00675	54	110110	Frame #4	8.31370849898475-5.00000000000002i	9.701430256		
0.006875	59	111011	Frame #4	9.36396103067891-8.4982203369473i	12.64529616		
0.007	52	110100	Frame #4	-20	20		
0.007125	51	110011	Frame #4	9.36396103067894+8.49822033694727i	12.64529616		
0.00725	51	110011	Frame #4	8.31370849898477+4.99999999999999i	9.701430256		
0.007375	59	111011	Frame #4	-3.36396103067891+9.46183980924129i	10.04204393	1	
0.0075	56	111000	Frame #4	-2-4i	4.472135955		2
0.007625	58	111010	Frame #4	-3.36396103067891+14.8760533716144i	15.25166213		
0.00775	58	111010	Frame #4	-14.3137084989848-4.999999999999	15.16186832	1	

				98i			
0.007875	52	110100	Frame #4	9.36396103067894+5.91243389932038i	11.07432349		2

Table 3.1 Illustration Of Watermark Embedding Procedure

The partial sums obtained are A_i = 14.80549, and B_i = 11.11667. The difference between these two is A_i - B_i = 3.688823. To make it embed a bit 1, this difference must be greater than a certain threshold. To make it embed a bit 0, this difference must be less than the same given threshold.

To embed the bit 1, the N pairs (a_i, b_i) are modified as per the following equation:

$$a'_i = a_i + \delta$$
$$b_i = b_i - \delta$$

Thus, using pseudorandom keys, value δ is added to first value and subtracted from the second value of all the coefficients. In this scheme, 4 frames are used to embed a single bit as per the Pseudo Noise sequence. However, considering frames of substantial length, a single frame can be used to embed a single bit. Subsequent bit are embedded in the same way in the Fourier Coefficients.

In the detection process, the N pairs of values are retrieved again using the same key and the result

$$S = \sum_{i \in key}^{R} a'_i - \sum_{i \in key}^{R} b'_i$$

is computed. If the signal values actually contains the watermark bit, then it is expected that the difference of the values would be greater than a threshold (or less than the same threshold), otherwise it is approximately zero. This is because the randomly selected signal values are independent and identically distributed (i.i.d) and it can be expected that the summation of the differences is approximately zero. This statistical assumption can be written as

$$\sum_{i=0}^{N} (a_i - b_i) \sim 0$$

If several pairs are selected from the cover signal, these will be independent and identically distributed.

B. Algorithm for Spread Spectrum Audio Watermarking in Magnitude of Fourier Coefficients.:

For embedding watermark in audio file in robust way, patchwork watermarking in the domain of Fourier magnitude coefficients is used.
The steps for watermark embedding can be described as shown:

(1) Digitize the given audio file into PCM samples with requisite sampling rate.
(2) After obtaining the PCM samples, covert the samples into frequency domain using Discrete Fourier Transform.
(3) The Fourier coefficients are complex numbers of the form a + ib, where a,b ϵ R and i= $\sqrt{1}$.
(4) Convert the given coefficients into Polar Form a + ib = r.$e^{i\theta}$, where
(5) $r = \sqrt{a^2 + b^2}$ and $\theta = \tan^{-1}\frac{a}{b}$.
(6) The proposed algorithm change magnitude (r) of the Fourier Coefficients but keep the phase (θ) remains unchanged.
(7) The magnitude of the Fourier Coefficients is modulated corresponding to the spread sequence and the watermark bits while keeping the symmetry unchanged.
(8) Inverse Fourier Transform is then computed to get the watermarked PCM samples.

C. Algorithm for Watermark Detection:

The watermark detection can be done as per the following algorithm:

(1) Digitize the given audio file into PCM samples with specified sampling rate.
(2) After obtaining the PCM samples, covert the samples into frequency domain using Discrete Fourier Transform.
(3) The Fourier coefficients are complex numbers of the form a + ib, where a,b ϵ R and i= √-1.
(4) Convert the given coefficients into Polar Form a + ib = r.$e^{i\theta}$, where $r = \sqrt{a^2 + b^2}$ and $\theta = \tan^{-1}\frac{a}{b}$.
(5) Magnitude of Fourier coefficients corresponding to the key values are considered and the difference is computed.
(6) If the difference is greater than a certain threshold, then the embedded bit is approximated as 1, if it is less than same threshold, it is zero. if it lies between the threshold, then it indicates that data is corrupted or erased.

The detection algorithm requires a perfect synchronization at the receiver otherwise the embedded watermark cannot be extracted.

An important parameter to be investigated is the extent of variation in the Fourier Coefficients so as to make watermark robust enough to survive against attacks while at the same time, imperceptible to human ears.

D. Psycho Acoustic Frequency Masking (PAFM):

As the magnitude of Fourier Coefficients is changes in accordance with embedding the bit '1' or '0', it is possible that some inaudible frequency components rises to audible level, and vice versa.

One parameter to be adjusted in order to make the relative level or audibility or inaudibility of the components is to make the step δ, as small as possible to make the watermarking being close to imperceptible. However, small δ corresponds to smaller threshold which gives less robust (or fragile) watermarking. One way to overcome this is to use a long PN sequence which in turn requires more frames leading to high computational complexity and less embedding capacity.

Section 4 discusses the simulation of the model and discusses the results. MATLAB is used as simulation tool and results are investigated and conclusions are drawn.

IV. Results and Conclusion

A. Time Domain Analysis of mp3 file:

A music file "xperia.mp3" is analyzed for watermark embedding. The information of the audio file can be obtained using the info command in matlab console.

COMMAND

```
file = 'xperia.mp3';
info = audioinfo(file)
```

OUTPUT :

```
Filename: 'C:\avni\xperia.mp3'
CompressionMethod: 'MP3'
NumChannels: 2
SampleRate: 24000
TotalSamples: 711360
Duration: 29.6400
Title: []
Comment: []
Artist: []
BitRate: 96
```

The info array shows the filename along with the location as its first attribute. The next attribute is the compression method which is mp3. A compression method is an encoding of the digital file so as to make it compatible for certain operations. The third parameter is the number of channels in the audio file which is generally 2 or 4. A channel depicts a layer of music or voice so as to improve the quality of music. The file under consideration is a dual channel audio file. Sampling rate is one of the most important parameter of the audio file. It depicts the number of PCM samples that are played every second, under normal mode. However, sampling rate is adjustable in certain audio players. The sampling rate of audio file under consideration is 24000 samples per second. The next parameter is total number of samples. For the file investigated, it is 711360. The length (in seconds) of the audio file can be obtained by dividing the total number of samples by the sampling rate, i.e. 711360/24000 = 29.64 which is the next element of the array info. Title, Comment and artist are the elements of the array which hold general information of audio file and can be set and reset using MATLAB commands. Compression efficiency of encoders is typically defined by the bit rate. Bit rate of 96 means that there are 96 kilobits of data stored in every second of the audio file. The time domain analysis of the audio file "xperia.mp3", (also included in the CD ROM) is shown in figure 4.1.

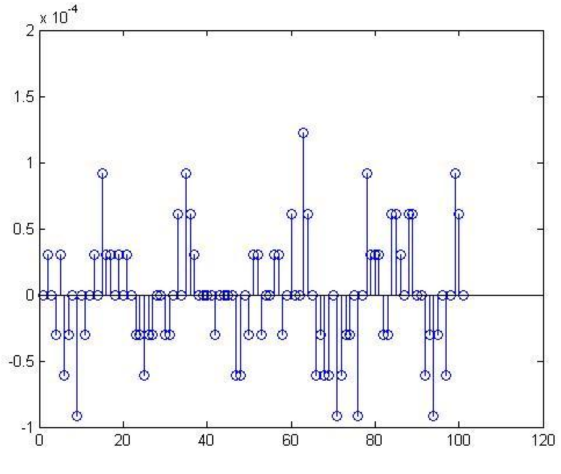

Fig. 4.1 Time Domain Plot of xperia.mp3

It illustrates 100 PCM samples of channel 1 of the audio file through the code shown below:

CODE:

```
file = 'xperia.mp3';
[y,Fs] = audioread(file);
ys = y(2048:2058,1);
stem(ys);
```

B. Fourier Analysis of the audio samples:

The Fourier Analysis of the audi samples can be done using the following code.

CODE

```
[y,Fs] = audioread(file);
ys = y(2048:2058,1);
ys_fft = fft(ys);
stem(ys_fft);
```

The output of the code is illustrated in figure 4.2 shown below.

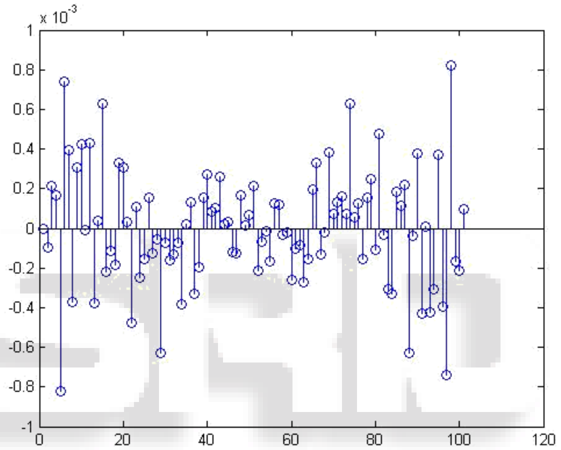

Fig 4.2 Fourier Coefficients corresponding to 100 samples illustrated in figure 4.1

It can be verified from the above figure, that corresponding to any number of samples, the Fourier Transform keeps the symmetry. Half of the Fourier Coefficients are complex conjugates of the other half of the coefficients in symmetrical way as illustrated in figure 4.2. The two parameters, apart from symmetry, associated with Fourier coefficients is amplitude and phase. The amplitude is the absolute value or modulus of the complex number. The tangent of the phase is the ratio of the real and imaginary part. In the proposed algorithm, watermark bits are embedded in the amplitude of the Fourier coefficients while keeping the phase unchanged. However, the conjugates are correspondingly changed so as to maintain the symmetry of the Fourier Coefficients.

C. Analysis of the digital watermark:

In the proposed work, text string is used as a watermark. This text string is converted into ASCII code and finally into binary code to yield a binary watermark. This binary watermark is then modulated using m sequence and embedded in the magnitude of the Fourier Coefficients of the watermark. The text string to be used as watermark is operated as per the following code:

CODE:

```
message = 'RCEW MUSIC';
message_ascii = uint8(message);    % vector corresponding to message bits
[r,c] = size(message_ascii)
message_bin = zeros(0,0);
for i = 1:c
tmp = de2bi(message_ascii(1,i));
[r1,c1] = size(tmp);
if c1<7
t= zeros(1,7-c1);
tmp = horzcat(t,tmp);
end
message_bin = horzcat(message_bin,tmp);
end
wvtool(message_bin);
```

OUTPUT

```
r =1      % 1 row
c = 10    % 10 columns
```

Text	**R**	**C**	**E**	**W**	**[space]**	**M**	**U**	**S**	**I**	**C**
ASCII (7 bit) (Decimal)	82	67	69	87	32	77	85	83	73	67
Binary	1010010	1000011	1000101	1010111	100000	1001101	1010101	1010011	1001001	1000011

Table 4.1: illustration Of Watermark And The Embedding Process

Thus, the size of the binary watermark vector is 10*7 = 70 bits. For each of the watermark bit, an entire 7 bit length m sequence is embedded in the audio samples, hence given the name- spread spectrum. Thus a total of 70*15 = 1050 bits needs to be embedded to watermark the audio. In the proposed watermark embedding algorithm, 1024 samples of audio are modulated to embed a bit. Thus, the number of samples need to be modified to embed entire spread sequence is 70*7*1024 = 501760. As the total number of samples in the original audio is 711360, the watermark can be embedded into the audio file.

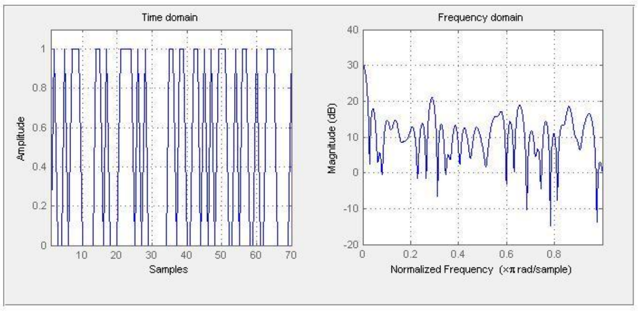

Fig. 4.3: Plot of watermark samples and Frequency Domain Analysis

The index values of the samples shown in figure 4.2 , after watermarking processing are depicted in Figure 4.4

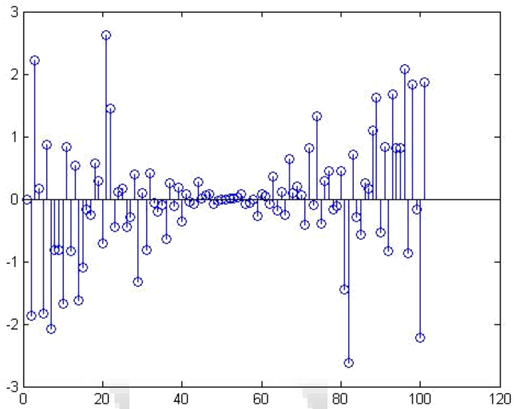

Fig 4.4: Fourier Coefficients corresponding to 100 watermarked samples for the sample indexes as illustrated in figure 4.1

It is clear from figure 4.2 and 4.4 that spread spectrum based watermarking in the frequency domain creates a disturbance in the sample values. This results in perceptibility of the watermark as some of the samples which lies in the audible range becomes inaudible and vice versa. This can be reduced by keeping the threshold value for watermark embedding as low as possible and number of samples to embed a single bit as high as possible., thereby reducing the embedding capacity.

In this embedding process, the first PN sequence consists of 20, 40, 60, 80 and 100th sample of the audio file and the absolute value of the coefficients is considered and then summation is computed.

For a separate PN sequence (PN-2), consisting of 10, 30, 50, 70 and 90th sample of the file, the absolute value of the coefficients is considered and then summation is computed.

For embedding a bit 1, the difference between the above two values is checked and if it lies well above a certain threshold, then 1 is implicitly considered to be embedded. Otherwise, the coefficients are modified keeping the phase unchanged as illustrated in the above table. In the above table, a constant 0.5 is added repeatedly to the coefficients keeping the phase unchanged.

D. Analysis of the watermarking Process under various parameters:

The analysis of the watermarking process under various parameters of embedding for 'xperia.mp3' are illustrated as shown in table 4.2.

Watermark Message Bits	**Total bits which needs to be embedded**		
	m sequence length = 7	**m sequence length = 15**	**m sequence length = 31**
10	70	150	310
20	140	300	620
30	210	450	930
40	280	600	1240
50	350	750	1550
60	420	900	1860
70	490	1050	2170
80	560	1200	2480
90	630	1350	2790
100	700	1500	3100
110	770	1650	3410
120	840	1800	3720
130	910	1950	4030
140	980	2100	4340

Table 4.2 Spreaded Message Length As A Function Of Length Of M Sequence

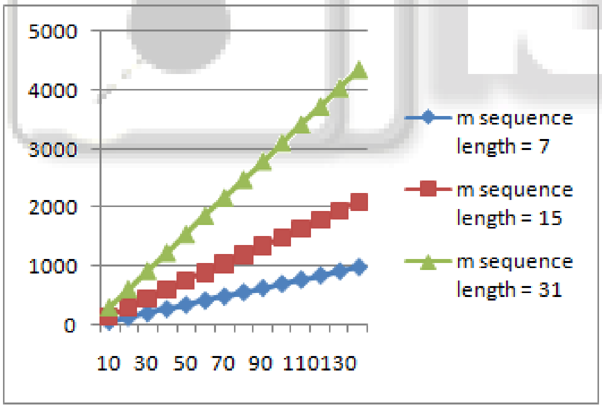

Fig. 4.5: Plot corresponding to table 4,2, corresponding to the length of spreaded message as a function of length of spread sequence. Horizontal axes shows bits in the original binary message and vertical axes shows bits in the spreaded message.

Spreaded Message Bits	**Total number of samples required for embedding of message**						
	Samples needed for 1 bit = 4	**Samples needed for 1 bit = 5**	**Samples needed for 1 bit = 6**	**Samples needed for 1 bit = 7**	**Samples needed for 1 bit = 8**	**Samples needed for 1 bit = 9**	**Samples needed for 1 bit = 10**
70	280	350	420	490	560	630	700
140	560	700	840	980	1120	1260	1400
210	840	1050	1260	1470	1680	1890	2100
280	1120	1400	1680	1960	2240	2520	2800
350	1400	1750	2100	2450	2800	3150	3500
420	1680	2100	2520	2940	3360	3780	4200
490	1960	2450	2940	3430	3920	4410	4900
560	2240	2800	3360	3920	4480	5040	5600
630	2520	3150	3780	4410	5040	5670	6300
700	2800	3500	4200	4900	5600	6300	7000
770	3080	3850	4620	5390	6160	6930	7700
840	3360	4200	5040	5880	6720	7560	8400
910	3640	4550	5460	6370	7280	8190	9100
980	3920	4900	5880	6860	7840	8820	9800

Table 4.3: Number Of Samples Modified As A Function Of Number Of Samples Per Bit Of Spreaded Message

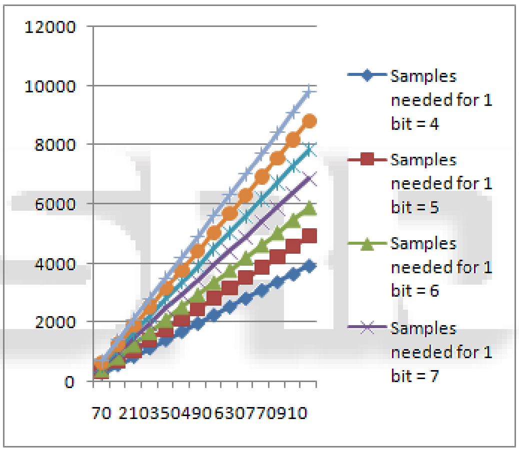

Fig. 4.6: Plot corresponding to table 4.3, corresponding to the number of PCM samples as a function of number of samples per bit. Horizontal axes shows bits in the spreaded message and vertical axes shows samples required for watermarking.

It is evident from the results presented in section 3 that imperceptibility increases as the step change in magnitude of PCM samples increases which makes the watermarking scheme fragile. However, robustness increases with the increase in the step size but makes watermark audible at times. If the step size is to be kept small, while making the minimum threshold between zero and one bit large, more number of samples needs to be changed giving low embedding capacity. Thus, audio watermarking using spread spectrum techniques gives a tradeoff between robustness, imperceptibility and embedding capacity.

PSNR value is used as a quality metric for watermark embedding purpose. The result of PSNR values of three different type of music categories under parameter settings described are depicted in table 4.4 and figure 4.7.

Parameter Specification	Valu

	e
Total Duration of Audio	30 Sec.
Sampling Rate	2400 0
Total Samples	7200 00
Watermark Bits	70
m Sequence Chip Rate	7
Number of Samples for 1 bit	1024
Total Samples	5017 60
Bit Rate (number of bits processed per second when audio is played)	96 kb
Bits per sample = Bit Rate/sampling rate	~4

Table 4.4: Peak Signal To Noise Ratio Evaluation For Different Type Of Music

	MSE	**PSNR**
POP	7.54	19.84652
ROCK	8.25	17.24993
JAZZ	8.61	16.01755

Table 4.5: MSE & PSNR

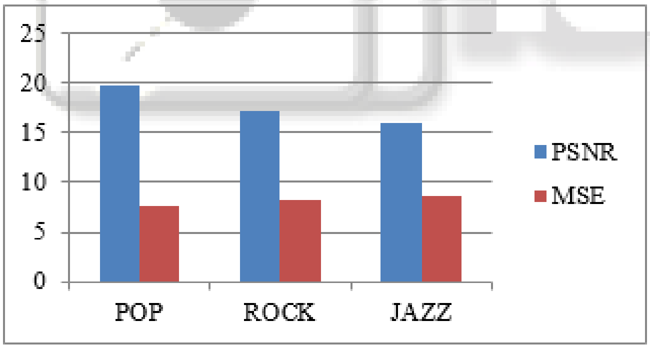

Fig. 4.7 Plot corresponding to table 4.4, corresponding to PSNR and MSE for different music types

It is evident from the above figure that maximum embedding quality (imperceptual) for same watermark is obtained in case of POP as compared to jazz or rock possibly due to its high scale. Low scale sounds are comparatively more distorted in watermark embedding process as compared high scale sound files. Section 5 discusses the results and draws the conclusion.

E. Future Scope:

An audio Watermarking Algorithm is a watermarking process applied on Audio signal. Watermarking of audio can be done in temporal and frequency domain. The temporal domain is very sensible to all the forms of echo, noise addition, down sampling, encoding. That is why the Watermark audio algorithm are mostly applied in the frequency domain. The embedded content should be statically undetectable, inaudible, robust again manipulation and secure. Thus to be efficient for monitoring a file, fingerprinting, and to indicate if a content has been manipulated.

In this paper, Patchwork Based watermark embedding is improved using spread spectrum techniques in frequency domain. M sequence produced by linear feedback shift register is used to modulate the watermark bits which are then embedded into the Fourier coefficients using patchwork algorithm. Also, PAFM is investigated as a function of embedding capacity, robustness and imperceptibility. It is concluded that PAFM as a function of watermark step size and threshold, depends upon the message length and therefore, the total number of samples under consideration. It also depends upon the type of music file under consideration. The scaling factor, frame size and the selected frequency band are the three adjustable parameters of this method which regulate the capacity, the perceptual distortion and the robustness trade-off of the scheme accurately. Furthermore, the suggested scheme is blind, since it does not need the original signal for extracting the hidden bits. The experimental results show that this scheme has a high capacity (0.7 to 2.8 kbps) without significant perceptual distortion.

An open problem which is not often considered is modern attacks, such as compression, channel fading, jitter and packet drop. These problems are severe for the case of watermark embedding algorithm presented. However, these issues can be resolved by embedding bits using repetition in the cover media. However, this reduces the embedding capacity. Moreover, the synchronization is still a challenge. The detector should be able to extract watermark from the marked data without the need of accurate synchronization. Recently, with rapid production of digital media, these modern attacks are becoming more important. These attacks are particularly relevant in various networks such as GSM and the Internet. For future work, this problem will be considered and techniques will be worked out to make the proposed method robust against these issues and attacks.

REFERENCES

[1] M. Davarynejad, S. Sedghi, M. Bahrepour, C.W. Ahn, M. Akbarzadeh, C. A. Coello Coello,Detecting Hidden Information from Watermarked Signal using Granulation Based Fitness Approximation, Applications of Soft Computing: From Theory to Praxis, Springer, Series: Advances in Intelligent and Soft Computing, Volume 58/2009, ISBN 978-3-540-89618-0, pp. 463-472, 2009.

[2] S. Katzenbeisser and F. A. P. Petitcolas, *Hiding Techniques for Steganography and Digital Watermarking*, S. Katzenbeisser and F. A. P. Petitcolas, Eds. Boston, MA: Artech House, 2000.

[3] K. Su, D. Kundur, and D. Hatzinakos, "Statistical Invisibility for Collusion-resistant Digital Video Watermarking," *IEEE Transactions on Multimedia 2004*

[4] K. Su, D. Kundur, and D. Hatzinakos, "Spatially Localized Image-dependent Watermarking for Statistical Invisibility and Collusion Resistance," *IEEE Transactions on Multimedia 2004*

[5] K. Su, "Digital Video Watermarking Principles for Resistance to Collusion and Interpolation Attacks,"

Master of Applied Science thesis, University of Toronto, Sept. 2001.

[6] Cox, M. Miller, and J. Bloom, ”Digital watermarking,” Morgan Kaufmann Publishers, Oct. 2001, ISBN 1-55860- 714-5.

[7] Bender, W., Gruhl, D., & Morimoto, N. (1996). Techniques for data hiding. *IBM Systems Journal*, *35*(3), 313-336.

[8] Cox, I., & Miller, M. (2001). Electronic watermarking: The first 50 years. In *Proceedings of the IEEE Workshop on Multimedia Signal Processing* (pp. 225-230). Cannes, France.

[9] Hartung, F., & Kutter, M. (1999). Multimedia watermarking techniques. In *Proceedings of the IEEE*, *87*(7), 1709-1107.

[10] Swanson, M., Zhu, B., & Tewfik, A. (1999). Current state-of-the-art, challenges and future directions for audio watermarking. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems* (pp. 19-24). Florence, Italy.

[11] Cox, I., Miller, M., & Bloom, J. (2003). *Digital watermarking*. San Francisco: Morgan Kaufmann.

[12] Johnston, J. (1998). Estimation of perceptual entropy using noise masking criteria. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (pp. 2524-2527). New York.

[13] Kirovski, D., & Malvar, H. (2001). Spread-spectrum audio watermarking: Requirements, applications, and limitations. In *Proceedings of the IEEE International Workshop on Multimedia Signal Processing* (pp. 219-224). Cannes, France.

[14] Steinebach, M., Petitcolas, F., Raynal, F., Dittmann, J., Fontaine, C., Seibel, S., et al. (2001).Stirmark benchmark: Audio watermarking attacks. In *Proceedings of the International Conference on Information Technology: Coding and Computing* (pp. 49-54). Las Vegas, Nevada.

[15] Noll, P. (1993). Wideband speech and audio coding. *IEEE Communications Magazine*, *31*(11), 34-44.

[16] Wu, M., Trappe, W., Wang, Z., & Liu, K. (2004). Collusion-resistant fingerprinting for multimedia. *IEEE Signal Processing Magazine*, *21*(2), 15-27.

[17] Chenyu, W., Jie, Z., Zhao, B., & Gang, R. (2003). Robust crease detection in fingerprint images. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern* Recognition (pp. 505-510). Madison, Wisconsin.

[18] Hong, Z., Wu, M., Wang, Z., & Liu, K. (2003). Nonlinear collusion attacks on independent fingerprints for multimedia. In *Proceedings of the IEEE Computer Society Conference on Multimedia and Expo* (613-616). Baltimore.

[19] Trappe, W., Wu, M., Wang, Z., & Liu, K. (2003). Anti-collusion fingerprinting for multimedia. *IEEE Transactions on Signal Processing*, *51*(4), 1069-1087.

[20] Termont, P., De Stycker, L., Vandewege, J., Op de Beeck, M., Haitsma J., Kalker, T., et al. (2000). How to achieve robustness against scaling in a real-time digital watermarking system for broadcast monitoring. In *Proceedings of the IEEE International Conference on Image Processing* (pp. 407-410). Vancouver, British Columbia.

[21] Termont, P., De Strycker, L., Vandewege, J., Haitsma, J., Kalker, T., Maes, M., et al. (1999). Performance measurements of a real-time digital watermarking system for broadcast monitoring. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems* (pp. 220-224). Florence, Italy.

[22] Depovere, G., Kalker, T., Haitsma, J., Maes, M., de Strycker, L., Termont, P., et al. (1999). The viva project: Digital watermarking for broadcast monitoring. In *Proceedings of the IEEE International Conference on Image Processing* (pp. 202-205). Kobe, Japan.

[23] Kalker, T., & Haitsma, J. (2000). Efficient detection of a spatial spread-spectrum watermark in MPEG video streams. In *Proceedings of the IEEE International Conference on Image Processing* (pp. 407-410). Vancouver, British Columbia.

[24] Bloom, J., Cox, I., Kalker, T., Linnartz, J., Miller, M., & Traw, C. (1999). Copy protection for DVD video. *Proceedings of the IEEE*, *87*(7), 1267- 1276.

[25] Craver, S., & Stern, J. (2001). Lessons learned from SDMI. In *Proceedings of the IEEE International Workshop on Multimedia Signal Processing* (pp. 213-218). Cannes, France.

[26] Cox I. J., Kilian J. Leighton F. T. and Shamoon T. 1997, “ Secure Spread Spectrum Watermarking for Multimedia”. *IEEE Trans. On Image Processing*, Vol. 6, No. 12, pp. 1673- 1687.

[27] Qiao L. and Nahrstedt K. 1999, “Non-invertible Watermarking Methods for MPEG Encoded Audio”, *Conf. on Security and Watermarking of Multimedia Contents*, pp. 194-202

[28] Neubauer C. and Herre J. 2000a, “Audio Watermarking MPEG-2 AAC Bitstream”, *108th AES Convention, Audio Engineering Society Preprint 5101*, Paris.

