A Novel Transcode Resistant Audio Watermark Technique

Suvra Sankar Bandyopadhyay
Department of Computer Science Engineering
Maulana Abul Kalam Azad University of Technology
Kolkata – 700064, India

Mihir Sing
Department of Computer Science Engineering
Maulana Abul Kalam Azad University of Technology
Kolkata – 700064, India

ABSTRACT

In today's world audio files are mostly used in compressed forms transcoded from one to another by freely available software tools that can degrade audio quality and damage any hidden watermark embedded in the file. In this paper, we present a novel audio watermarking approach by embedding two step secured watermark data in the audio file and then extracting from various converted transcoded formats with different bit rates and sampling frequencies resulting in lower Bit Error Rate.

Keywords

Watermarking, Hamming, Arnold, Bit Error Rate.

1. INTRODUCTION

The process in which watermark is embedded imperceptibly into digital media as persistent signs and then obtained from the media ensuring its authenticity is known as Digital watermarking. The watermark is always attached with the digital media to be owner protected. This means that each digital media has its own watermark or each owner has his/her exclusive watermark. Music distribution around the Internet has acquired an easy way to get audio recordings of popular singers and bands. In this framework digital audio watermarking provides effective protection against illegitimate copying or distribution of audio information and further enables the robust and imperceptible copy of extra right information within the audio file.

Audio Transcoding is used for conversion from one encoded form to another encoded form of audio data files for various purposes. This is a compression technique with some loss of data and can be found in many areas of content adaptation primarily for mobile devices. In these conversions as data may be lost due to the nature of the compression, ownership information hidden in the audio data for identification and copy control purpose may also be lost if the watermark information is not properly hidden and secured. In our algorithm we demonstrated how the watermark information changes or can be distorted after different types of audio conversion (compression and decompression) process. We also analyzed the outcome (i.e., the extracted watermark information) in terms of bit error rate and showed that using Hamming code to secure the watermark information before embedding in the audio file results in substantial increase or decrease in bit error rate.

2. REQUIREMENTS FOR AUDIO WATERMARKING

1. Robustness: Watermarks must be able to cope with all problematic situations such as compression, conversion, noise or any other form of attack. [1, 2].
2. Capacity: The data stored in the audio file should be large enough to give adequate information about the owner of the digital media.
3. Perceptual transparency: Watermarks which are enclosed in audio files can be of two types. They can either be perceptible or imperceptible. Imperceptible watermarks are preferred in case of audio files because of their robustness. [1, 2].

However, achieving all the three requirements simultaneously is not feasible. With approximation for obtaining an optimal solution, it is possible to achieve a compromise among these requirements. Such an optimal solution is dependent on the algorithm which is being used for watermarking.

3. THEORY BEHIND PROPOSED ALGORITHM

Watermarking in wav audio has the advantage of hiding the information in the redundant portion to minimize the distortion caused by the watermark data. But in modern days audio file is most often used in different compressed and other usable formats not in the raw format. In this process raw audio file is converted to other formats by audio converter tool and those converter processes are mostly prone to losses and remove the redundant data from the audio. This way, most of the watermark data also lost and it is a very difficult to reconstruct the watermark bits as the data is already lost. Several audio watermarking algorithms [3-6] were studied and in most of the papers the authors focused primarily on watermarking on a particular type of audio format in wav and in compressed domain mp3 format and tested the robustness of the watermark and watermarked audio quality by their proposed algorithms. In one of the papers[3], the authors designed an algorithm scheme for mp3 to securely embed the watermark information directly in the compressed domain with altered time and sub-band domain. In another paper[4], a lossless and secure watermarking scheme is presented on mp3 audio file by changing the private bit of mp3 header in the time domain. In our proposed method the focus was not only on watermark insertion algorithm but also on ensuring that the watermark data is secured in such a way that after extracting from the audio file the watermark data could be reconstructed that signifies minimizing the Bit Error Rate properly.

Hamming Code (1 bit error detection and correction) is a very popular single bit error detection and error correction method in data communication. It works with an idea that if many error-correcting bits are included with data and if those bits can be arranged in such a way that different incorrect bits produce different error results, then error bits could be identified. With k parity bits, it can cover bits from 1 up to 2k-1 of binary data. Among this the data part would be 2k-k-1 bits. In our algorithm to minimize the bit error rate, consider k = 3. So the total hamming length will be 7. So the data part will be 7-3 = 4 bits and 3 parity bits. Hamming Code is only able to correct one bit of error from a data stream. Thus, the minimal data block size of 4 bits of the watermark data would minimize the probability of having error more than one bit in each block. The Hamming

Code algorithm in encoding the watermark data was used with the purpose of solving two issues. Firstly the watermark data got secured and the bit error correction could also be done in the extraction process while decoding. And second by taking block size of length 4, the hamming coded watermark data length also increased by 3 in each data block and resulting in enhanced robustness.

4. PROPOSED ALGORITHM

Step A: Watermark Creation

1) The watermark image (black and white) is read and preprocessed by Arnold Transformation [7] to encrypt the secret information and the private key (let KI) is stored for decryption and then converted into single dimension array. Let the length of the array be n.

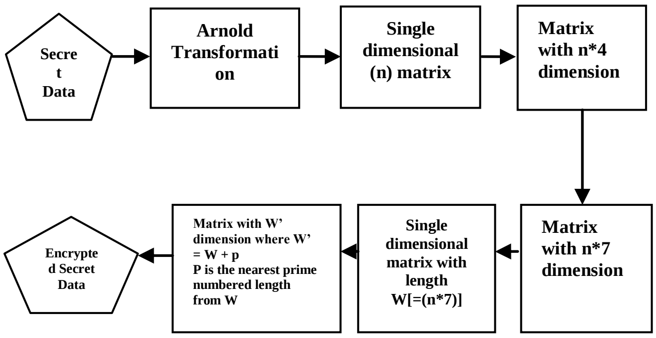


Fig 1.Watermark Creation Diagram

2) In order to increase the robustness of the watermark data extra security is added to it with the following steps:

2.1) An array of m x 4 dimension is made where m = n/4. If n is not divisible by 4, it is made divisible by 4 by appending extra bits to the array and the count of extra zeros is stored for extraction procedure in future.

2.2) Now the hamming code (error detection and correction) is applied to each row of the newly constructed matrix(each row having length 4) and Hamming Code word(7,4) is built where no. of parity bits is 3(7-4) and as a result hamming coded matrix with dimension m x 7 is obtained.

2.3) Now it is made into a one dimensional array and thus the encoded watermark data is created. Let the length of the array be W where W = m x 7.

2.4) To make the watermark data more secure extra bits are appended to the array to make W a prime number. W' = W + p, where p is the extra length of bits of for the nearest prime number and p bits are stored for future extraction procedure. It will be used as a private key for the users.

Step B: Watermark insertion in the audio data

1) The no of blocks of the audio file is selected by selecting size of each block. Let the size of each block is B and total length of the audio data stream be L, so the no of block is B/L = A.

2) It should be ensured that length A should be greater than length W so that length of watermark bits does not exceed the number of audio blocks.

3) The starting block is selected according to convenience. Also a watermark peak value (let it be Wp) is selected such that it corresponds to a low frequency peak for watermark embedding and all the frequencies below the present spectrums are considered to be inaudible.

4) One bit at a time is read from the watermark array and iterated through the blocks of the audio data until all the bits of the watermark array are completed, that is up to the length W'.

For each iteration, let X be the audio block data with size B.

5) X is transformed into frequency domain (let it be Y) by Fourier transformation as in the frequency domain, information is hardly detectable without the knowledge of the embedding procedure or an analysis of the spectrum, which may show suspicious frequency peaks from watermarking.

6) Now beginning at number Wp(watermark peak value) peaks with different absolute values within the each audio data block of Y are searched except the middle portion of the block.

7) Let the two different consecutive peak values we get beY(n) and Y(n+1); then the following is done:

7.1) If $Y(n) < Y(n+1)$ and the watermark bit is 0 then swap the values of the two is swapped and the lower value that is Y(n) is amplified with a constant amplifying factor.

7.2) If $Y(n) > Y(n+1)$ and the watermark bit is 1 then the values of the two are swapped and the lower value that is Y(n+1) is amplified with a constant amplifying factor.

8) The inverse Fourier transform of the Y block is taken and replaced in the main audio data array.

Steps 5 through 8 are repeated for both the channels (if it is a stereo type audio) or single channel (for mono) audio file.

The process continued until all the watermark bits are complete.

After insertion of watermark data it is saved to a wav file and the watermarked audio file is created.

Step C: Watermark extraction

In our proposed algorithm we need three keys for watermark extraction procedure

- Arnold Transform Key (KI)
- Prime numbered length key (P)
- Dimensions of final watermark array (W')

We do not need the original file for watermark extraction process.

1. A one dimensional array of 1's with length W' is created.
2. The watermark insertion process is applied in the reverse way to create a one dimensional array of length W'
3. The last P bits from W' are removed to get W (= W' - P).
4. The array dimension would be 1 x W where W = n x 7
5. The one dimensional matrix is converted to n x 7 matrix.
6. Then apply the hamming decode process is applied to each row of the matrix and get the n x 4 matrix.
7. Then the matrix is converted to the corresponding image matrix dimension and then finally decrypted by Inverse Arnold Transformation with the key KI to reconstruct the image.

Watermark creation Algorithm with an example.

Consider an image matrix $\begin{pmatrix}1 & 0\\0 & 1\\1 & 1\end{pmatrix}$ with dimension (3 x 2)

After Arnold transformation the matrix becomes $\begin{pmatrix}0 & 1\\1 & 1\\1 & 0\end{pmatrix}$

It is then convert into an one dimensional matrix

$$\begin{pmatrix}0 & 1 & 1 & 1 & 1 & 0\end{pmatrix}$$

In our algorithm the matrix is converted into m x 4 where

m = n / 4, here n = 6.

As here n(=6) is not divisible by 4 so we append 2 extra random bits in the array to make it divisible by 4.So the new matrix becomes $\begin{pmatrix}0 & 1 & 1 & 1\\1 & 0 & 0 & 0\end{pmatrix}$ with dimension (2x 4).

Now hamming code(error detection and correction) is applied to each row of this matrix. When hamming code is applied on 4 elements it will give a codeword with 7 elements of which 4 bits are the original data part and the extra three bits are the parity bits. They are placed in between the data bits but not consecutively.

So the new matrix after applying hamming code becomes $\begin{pmatrix}0 & 0 & 0 & 1 & 1 & 1 & 1\\1 & 1 & 1 & 0 & 0 & 0 & 0\end{pmatrix}$ with dimension (2 x 7).

Now converting it into a single dimensional matrix becomes $\begin{pmatrix}0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0\end{pmatrix}$ with dimension 1 x 14 and length 14.

For giving extra security a random sequence is appended of bits of length 3 with this matrix to make it a length of 17 so that it becomes an

array of prime length. This number 3 is stored as private key for detection of the watermark data.

So the final watermark data becomes

$$\begin{pmatrix}0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1\end{pmatrix}$$

Now applying the watermark insertion algorithm the resulting watermark data is inserted in the audio data. In this case Wp = 2 is taken as 2 corresponds to 5Hz.

For watermark detection procedure for this example we assume that after applying the watermark detection algorithm on the audio data the extracted matrix is

$\begin{pmatrix}0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1\end{pmatrix}$ with length 17.This is the hamming coded data and needs to be decoded for getting the image matrix.

First the extra padded bits are removed from the matrix with the private key that is 3 in this example.

So the matrix becomes

$\begin{pmatrix}0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0\end{pmatrix}$ with dimension (1 x 14).

Now as it is known that the length of the hamming code word is 7 so after converting the matrix to (2 x 7) dimension it becomes $\begin{pmatrix}0 & 0 & 0 & 1 & 1 & 1 & 1\\1 & 1 & 1 & 0 & 0 & 0 & 0\end{pmatrix}$

Now applying the hamming decoding process on this matrix the matrix $\begin{pmatrix}0 & 1 & 1 & 1\\1 & 0 & 0 & 0\end{pmatrix}$ with dimension (2 x 4) is obtained.

To make it a multiple of foursome bits were appended to the original image matrix andby removing the bits it becomes $\begin{pmatrix}0 & 1\\1 & 1\\1 & 0\end{pmatrix}$ with dimension (3 x 2) which is the Arnold encrypted matrix. Now applying the Inverse Arnold transformation with the key KI to the matrix the actual matrix that is $\begin{pmatrix}1 & 0\\0 & 1\\1 & 1\end{pmatrix}$ is obtained.After reconstructing the image from the resultant matrix we get the watermark extracted image.

5. COMPARISON AND ANALYSIS

This algorithm was implemented in MATLAB and tested on randomly selecting wav audio samples of different file size and genre. After watermark insertion the watermarked audio file was converted into other useful formats(mp3,aac,flac,ogg,mp4,wma)using popular and available software tools[8,9] and then watermark from those types were extracted and analyzed the outcomes. Here a sample audio file of size 40.6 MB [10] with sampling frequency 44.1 KHz and bitrate 128Kbps is taken. A grayscale watermark image in .bmp format and of size 10.1 KB [11] is taken. After surveying various available audio converters two popular audio converter software were found 1. Free Audio Converter [8] and 2. Total Audio Converter [9] and the watermarked audio file were tested with both the softwares and the outcomes analyzed. In both these software first wav watermarked file is converted into popular compressed audio formats i.e., mp3, aac, ogg, wma and mp4 with different bit rates and sampling frequencies and then decompressed back to wav format. From all these formats the watermark information that was embedded in the wav format was extracted and the bit error percentage was compared with the proposed algorithm. In all the form of audio files the Signal-to-Noise-Ratio, ODG(Objective Difference Grade) standard values(Table 1) of the audio files were calculated for testing the quality of the files and it was found that the values fell within the normal range. In Table 4 and Table 5 the Bit Error Rates are presented from all the converted watermarked audio formats with and without applying hamming code and their difference along with the above mentioned audio quality evaluation parameters values of these audio files by software1[8] and software[9] respectively. In Table 2 and Table 3 the extracted watermark images are shown from the different types of audio format with varying bitrates and sampling frequency converted by software1 and software2 respectively from the watermarked wav audio file. In both the figures it is clearly seen that except in few cases where the bit rate is low the extracted images were reconstructed properly with the low bit error as we applied hamming 1 bit error detection and correction algorithm.

By applying hamming code to each row of the image watermark data matrix it result into two significant improvements in the audio watermarking procedure.

1. As after applying hamming code the hamming code word length is greater than the data word length so as a whole the length of the matrix increases to an extent. So more number of watermark data can be inserted in the audio main data and the robustness of the watermark data can be increased. But at the same time we have to consider that the increased robustness should have a limit so that it cannot degrade the audio quality. To have an optimal balance between the imperceptibility and robustness we consider the length of the data part to 4 so that after hamming code it becomes 7 bits out of which 4 bits are the data part and three bits are parity bits so for each row 3 extra bits are inserted.

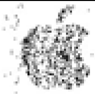
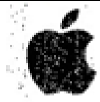
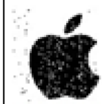
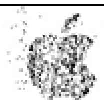
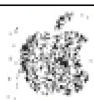
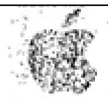
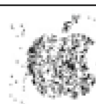
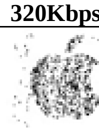
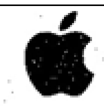
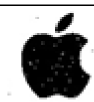
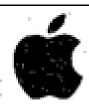
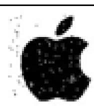
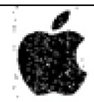
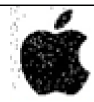
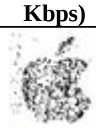
2. As hamming decode process can detect and correct only one bit of data so taking 4 bit at a time can substantially reduce the bit error rate probability. After applying hamming code error

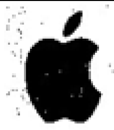
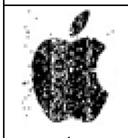
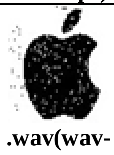
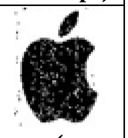
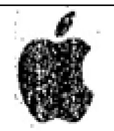
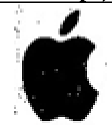
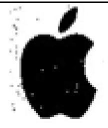
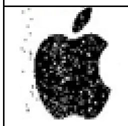
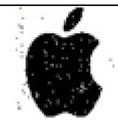
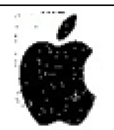
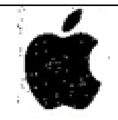
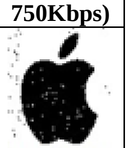
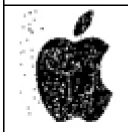
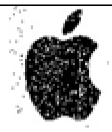
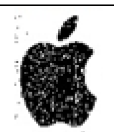
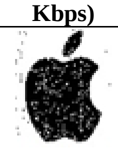
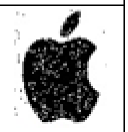
detection and correction method the watermarked audio file that is obtained is converted in many other audio formats and in each of the formats after extracting the watermark data the bit error rate is reduced to a significant percentage as the hamming decoding procedure successfully detects and corrects 1 bit error in the data word. Other than Arnold Transformation two private keys were used in this method. One is when the watermark data is converted to hamming data word appends extra bits to make it a multiple of 4 if necessary. To make the watermark data more secure some extra bits were appended to the hamming coded single dimensional matrix so that its length becomes prime and it is considered as the second private key. This key is used for the sole intention to make it almost impossible for any intruder to reconstruct the image matrix from the watermark data that was extracted from the audio files by any means cause the length of the watermark data would be of prime numbered.

Table 1. ODG Description

ODG >=	Description
0	Imperceptible
-1	Perceptible, not annoying
-2	Slightly annoying
-3	Annoying
-4	Very annoying

Table 2 and Table 3. Extracted watermark images from software 1 and software 2 by proposed algorithm

Original Image	Extracted Images from				
	 .wav	 .flac(wav to flac)	 .wav(wav to flac to wav)		
 .ogg(wav to ogg at 44.1KHz and 128Kbps)	 .wav(wav to ogg to wav at 44.1KHz and 128Kbps)	 .ogg(wav to ogg at 44.1KHz and 192Kbps)	 .wav(wav to ogg to wav at 44.1KHz and 192Kbps)	 .ogg(wav to ogg at 44.1KHz and 320Kbps)	 .wav(wav to ogg to wav at 44.1KHz and 320Kbps)
 .aac(wav to aac at 22.5KHz and 64Kbps)	 .wav(wav to aac to wav at 22.5KHz and 64Kbps)	 .aac(wav to aac at 44KHz and 192Kbps)	 .wav(wav to aac to wav at 44KHz and 192Kbps)	 .aac(wav to aac at 48KHz and 192Kbps)	 .wav(wav to aac to wav at 48KHz and 192Kbps)
 .mp3(wav to mp3 at 44.1KHz and 128Kbps)	 .wav(wav to mp3 to wav at 44.1KHz and 128Kbps)	 .mp3(wav to mp3 at 44.1KHz and 320Kbps)	 .wav(wav to mp3 to wav at 44.1KHz and 320Kbps)	 .mp3(wav to mp3 at 44.1KHz and variable Kbps)	 .wav(wav to mp3 to wav at 44.1KHz and 128Kbps)
 .wma(wav to wma at 44.1KHz and 128Kbps)	 .wav(wav to wma to wav at 44.1KHz and 128Kbps)	 .wma(wav to wma at 44.1KHz and 192Kbps)	 .wav(wav to wma to wav at 44.1KHz and 192Kbps)	 .wma(wav to wma at 44.1KHz and 256Kbps)	 .wav(wav to wma to wav at 44.1KHz and 256Kbps)

Original Image	**Extracted Images from**				
	 **.wav**	 **.mp4(wav-mp4 44.1KHz 108Kbps)**	 **.wav(wav-mp4-wav 44.1KHz 108Kbps)**	 **.mp4(wav-mp4 44.1KHz 180Kbps)**	 **.wav(wav-mp4-wav 44.1KHz 180Kbps)**
 **.ogg(wav-ogg 44.1KHz 128Kbps)**	 **.wav(wav-ogg-wav 44.1KHz 128Kbps)**	 **.ogg(wav-ogg 44.1KHz 192Kbps)**	 **.wav(wav-ogg-wav 44.1KHz 192Kbps)**	 **.ogg(wav-ogg 44.1KHz 320Kbps)**	 **.wav(wav-ogg-wav 44.1KHz 320Kbps)**
 **.aac(wav-aac 44.1KHz 108Kbps)**	 **.wav(wav-aac-wav 44.1KHz 108Kbps)**	 **.aac(wav-aac 44.1KHz 180Kbps)**	 **.wav(wav-aac-wav 44.1KHz 180Kbps)**	 **.aac(wav-aac 44.1KHz 750Kbps)**	 **.wav(wav-aac-wav 44.1KHz and 750Kbps)**
 **.mp3(wav-mp3 44.1KHz 128Kbps)**	 **.wav(wav-mp3-wav 44.1KHz 128Kbps)**	 **.mp3(wav-mp3 44.1KHz 320Kbps)**	 **.wav(wav-mp3-wav 44.1KHz 320Kbps)**	 **.mp3(wav-mp3 44.1KHz variable Kbps)**	 **.wav(wav-mp3-wav 44.1KHz 128Kbps)**
 **.wma(wav to wma at 44.1KHz 96Kbps)**	 **.wav(wav to wma to wav at 44.1KHz 96Kbps)**	 **.wma(wav to wma at 44.1KHz 128Kbps)**	 **.wav(wav to wma to wav at 44.1KHz 128Kbps)**	 **.wma(wav to wma at 44.1KHz 192Kbps)**	 **.wav(wav to wma to wav at 44.1KHz 192Kbps)**

From Table 4 and Table 5 it is very clear that by applying hamming encoding algorithm in all the given type of converted audio file the bit error rate is less than the direct watermark insertion algorithm without applying hamming code in the watermark data. By applying hamming code in the watermark data, the length of the watermark data increases but it does not distort the audio quality because the ODG values of all the audio files fall within the standard values [Table 1]. The peak signal to noise ratio (PSNR) and mean square error (MSE) values were calculated for each audio file and shown in Table 4 and Table 5 In this audio watermarking approach we aimed to minimize the loss of hidden watermarked information from audio data while it is transcoded from wav format to other audio formats by securing the watermark information. In addition, we corrected most of the error bits of the data by two steps, first by Arnold transform and then by Hamming Code (1 bit error detection and correction) mechanism. We demonstrated how the watermark information persists in the audio data as evidenced by two to be within the normal values. For various others robustness tests all the different converted audio files went through low pass filter attack, high pass filter attack and quantization attack. In all cases the output audio quality fells within the standard ODG range. The Bit Error Rate varies from one format to other but not more than 25% in any case. In our experiment we used image as the watermark file but text and audio files can also be used as sample watermark files for embedding into wav audio files.

6. CONCLUSION

Popular audio converter software. In many instances the Bit Error Rate was high as we changed the sampling frequency but compared with the direct watermark data insertion the later the Bit Error Rate was always minimal with the Hamming Coded data insertion. In future, we plan to focus on improving the overall robustness of the technique and further minimize the Bit Error rate for low sampling frequency and bit rates.

Table 4. Different watermarked audio formats from Software 1 (Free Audio Converter) and audio quality parameters evaluations

Audio Formats (watermarked to other formats)	Bit rate and sampling frequency	A : Bit Error Rate Normal [%]	B : Bit Error Rate with hamming [%]	Difference in methods (A - B)	MSE	PSNR	Time Taken	ODG Value
wav	Same as original	0.55946	0.087758	0.471702	0.31637	104.53	10.656	0.115
wav to mp3	128kbps 44 KHz	1.1957	0.257	0.9387	0.34677	105.53	11.656	0.078
	320kbps 44 KHz	2.385	1.285	1.1	0.32739	106.03	10.719	0.145
	320kbps 48 KHz	21.275	19.288	1.987	0.35263	105.38	12.266	-1.05
	Variable bit rate 44 KHz	1.0312	0.27581	0.75539	0.37268	104.9	9.6563	-1.95
wav to mp3 to wav	128kbps 44 KHz	3.8723	0.257	3.6153	0.14815	112.92	8.0156	0.245
	320kbps 44 KHz	2.385	1.285	1.1	0.13606	113.65	8.0156	-1.15
	320kbps 48 KHz	21.275	19.288	1.987	0.35263	105.38	8.3906	-2.15
	Variable bit rate 44 KHz	1.0202	0.26954	0.75066	0.14819	112.91	7.875	-2.15
wav to ogg	128kbps 44 KHz	2.7644	0.92773	1.83667	0.14926	112.85	14.766	0.095
	192kbps 44 KHz	2.1062	0.58923	1.51697	0.14921	112.85	17.047	-1.36
	320kbps 48 KHz	33.732	19.257	14.475	0.35444	105.34	20.797	-2.48
wav to ogg to wav	128kbps 44 KHz	2.7753	0.92773	1.84757	0.14859	112.89	8.2031	0.395
	192kbps 44 KHz	2.0952	0.38923	1.70597	0.12839	111.89	8.3065	-1.66
	320kbps 48 KHz	33.721	19.244	14.477	0.35402	105.35	8.4844	-2.8
wav to wma	128kbps 44 KHz	26.503	12.48	14.023	0.35126	105.42	17.813	-1.26
	192kbps 44 KHz	26.547	12.462	14.085	0.35126	105.42	16.781	-1.5
	256kbps 48 KHz	33.227	19.626	13.601	0.35243	105.39	16.969	-2.46
wav to wma to wav	128kbps 44 KHz	26.382	12.549	13.833	0.35104	105.42	7.875	-1.76
	192kbps 44 KHz	26.437	12.593	13.844	0.35104	105.42	8.1563	-1.6
	256kbps 48 KHz	33.392	19.445	13.947	0.35231	105.39	8.1406	-2.86
wav to aac	64kbps 22 KHz	34.116	20.169	34.116	0.34009	105.7	9.7969	-2.35
	192kbps 44 KHz	33.962	18.169	15.793	0.31009	104.7	9.1969	-1.5
	256kbps 48 KHz	34.028	25.063	34.028	0.35502	103.8	8.8569	-2.89
wav to aac to wav	64kbps 22 KHz	34.171	20.45	34.171	0.37009	106.8	8.8969	-3.01
	192kbps 44 KHz	33.787	19.169	14.618	0.35003	108.3	9.8763	-1.87
	256kbps 48 KHz	34.006	20.501	34.006	0.35009	105.7	7.79	-2.9
wav to flac	128kbps 48 KHz	33.71	18.001	15.709	0.34009	103.7	8.7902	-2.5

Table 5. Different watermarked audio formats from Software 2 (Total Audio Converter) and audio quality parameters evaluations

Audio Formats (watermarked to other formats)	Bit rate and sampling frequency	A : Bit Error Rate Normal [%]	B : Bit Error Rate with hamming [%]	Difference in methods (A - B)	MSE	PSNR	Time Taken	ODG Value
wav	Same as original	0.55946	0.087758	0.471702	0.23612	101.8	9.87	0.112
wav to mp3	96kbps 44 KHz	9.6204	3.993	5.6274	0.33612	105.8	10.859	0.115
	128kbps 44 KHz	12.165	5.2404	6.9246	0.33645	105.79	11.016	-1.25
	192kbps 44 KHz	2.5998	0.77101	1.82879	0.34074	105.68	11.594	-1.85
wav to mp3 to wav	96kbps 44 KHz	9.6314	3.9867	5.6447	0.14251	113.25	5.9688	0.35
	128kbps 44 KHz	12.154	5.2404	6.9136	0.14322	113.21	6.9531	0.85
	192kbps 44 KHz	2.6218	0.77728	1.84452	0.14409	113.16	5.625	-2.5

wav to ogg	96kbps 44 KHz	7.4814	2.3757	5.1057	0.25205	108.3	14.641	-1.85
	160kbps 44 KHz	7.2839	2.1814	5.1025	0.25173	108.31	17.094	0.74
	192kbps 44 KHz	7.229	2.2065	5.0225	0.25171	108.31	16.25	0.98
wav to ogg to wav	96kbps 44 KHz	7.4814	2.3757	5.1057	0.25179	108.31	7.3594	-2.79
	160kbps 44 KHz	7.2839	2.1814	5.1025	0.25158	108.32	7.1563	-1.10
	192kbps 44 KHz	7.202	2.2065	4.9955	0.2516	108.32	7.5	-1.88
wav to wma	96kbps 44 KHz	4.2453	1.4104	2.8349	0.14861	112.89	12.813	-1.03
	128kbps 44 KHz	4.0698	1.354	2.7158	0.14859	112.89	11.094	0.78
	192kbps 44 KHz	3.1593	0.87758	2.28172	0.14839	112.9	12.219	-1.25
wav to wma to wav	96kbps 44 KHz	4.2453	1.4229	2.8224	0.14788	112.93	7.5625	-2.11
	128kbps 44 KHz	4.0478	1.3352	2.7126	0.14785	112.93	7.0313	-1.56
	192kbps 44 KHz	3.1483	0.87131	2.27699	0.14786	112.93	9.2656	-1.45
wav to aac	108kbps 44 KHz	18.166	7.5597	10.6063	0.34018	105.7	24.813	0.84
	180kbps 44 KHz	16.839	7.1146	9.7244	0.34034	105.69	27.953	0.34
wav to aac to wav	108kbps 44 KHz	10.695	3.6545	7.0405	0.14775	112.94	7.5781	-1.06
	180kbps 44 KHz	6.6257	2.5387	4.087	0.14778	112.94	7.2969	-1.256
wav to flac	750kbps 44 KHz	0.55946	0.087758	0.471702	0.14823	112.91	10.484	-2.98
wav to flac to wav	750kbps 44 KHz	0.55946	0.087758	0.471702	0.14001	117.91	11.484	-1.251
wav file to mp4	108kbps 44 KHz	18.166	7.5597	10.6063	0.34018	105.7	23.922	0.98
	180kbps 44 KHz	16.839	7.1146	9.7244	0.34034	105.69	24.641	-1.563
wav to mp4 to wav	108kbps 44 KHz	10.695	3.6545	7.0405	0.14775	112.94	8.0469	-2.15

7. REFERENCES

[1] Mat Kiah, M. L., et al(2011). "A review of audio based steganography and digital watermarking." International Journal of Physical Sciences Vol. 6 No. 16 pp: 3837-3850.

[2] Kaur, Harleen. "STUDY ON AUDIO AND VIDEO WATERMARKING. "International Journal of Communication Network Security" ISSN: 2231-1882.

[3] Dimitrios Koukopoulos, and Yiannis Stamatiou."An Efficient Watermarking Method for MP3 Audio Files." International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:1, No:7, 2007.

[4] N. Moghadam, and H. Sadeghi."Genetic Content-Based MP3 Audio Watermarking in MDCT Domain." International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:1, No:12, 2007.

[5] Bailong Yang,Penghui Wu,Yaque Jing,and Jing Mao" Lossless and Secure Watermarking Scheme in MP3 Audio by Modifying Redundant Bit in the Frames." 2013 6th International Conference on Information Management, Innovation Management and Industrial Engineering.

[6] Changsheng Xu, Jiankang Wu & David Dagan Feng "Content-Based Digital Watermarking for Compressed Audio." Kent Ridge Digital Labs 21 Heng Mui Keng Terrace Singapore 119613.

[7] Kaur, Navjot, and Usvir Kaur. "Audio Watermarking using Arnold transformation with DWT-DCT." International Journal of Computer Scirnce Engineering. Vol.2 No. 6 pp: 286-294.

[8] Free Audio Converter Retrieved from http://www.dvdvideosoft.com/products/dvd/Free-Audio-Converter.htm (Datetime - 2016-03-18 19:18).

[9] Free Audio Converter Retrieved from https://www.coolutils.com/TotalAudioConverter(Datetime-2016-03-05,18:16).

[10] Sample wav audio file Retrieved from http://www.stiftung-swk.de/matlab/FFT_Audio_Watermark(Datetime - 2016-03-14 19:18).

[11] Sample watermark grayscale image file Retrieved from http://logok.org/wp-content/uploads/2014/04/Apple-Logo-black.png(Datetime-2016-03-18 20).

[12] W. Bender, D. Gruhl, N. Morimoto and A. Lu, "Techniques for data hiding," IBM Systems Journal, Vol. 35, No. 3&4, pp. 313-336, 1996.

[13] Ketcham, M., & Vongpradhip, S. (2007). "Intelligent audio watermarking using genetic algorithm in DWT domain". International Journal Of Intelligent Technology, Vol.2 No.2, pp:135-140.

[14] L. Boney, A. Tewfic and K. Hamdy, "Digital watermarks for audio signals," IEEE Int. Conf. on Multimedia Computing and Systems, pp.473-480, 1996.