High capacity audio watermarking using the high frequency band of the wavelet domain

M. Fallahpour and D. Megias

Estudis d'Informàtica, Multimèdia i Telecomunicació, Universitat Oberta de Catalunya, Rambla del Poblenou, 156, 08018 Barcelona, Spain

MFallahpour@uoc.edu, DMegias@uoc.edu

Abstract: This paper proposes a novel high capacity robust audio watermarking algorithm by using the high frequency band of the wavelet decomposition at which the human auditory system (HAS) is not very sensitive to alteration. The main idea is to divide the high frequency band into frames and, for embedding, to change the wavelet samples depending on the average of relevant frame's samples. The experimental results show that the method has a very high capacity (about 11,000 bps), without significant perceptual distortion (ODG in [–1 ,0] and SNR about 30dB), and provides robustness against common audio signal processing such as add noise, filtering, echo and MPEG compression (MP3).

Keywords: *Audio watermarking, digital wavelet transform.*

1. Introduction

Digital watermarking is one of the most popular approaches for providing copyright protection of digital contents. This technique is based on direct embedding of additional information data into the digital contents. The watermarking process should not introduce any perceptible artifacts into the original contents (*e.g.* an audio signal). Ideally, there must be no perceptible difference between the watermarked and the original digital contents. *I.e.* the watermark data should be embedded imperceptibly into the audio media. Using the properties of the human auditory system (HAS) is a usual approach to design imperceptible and robust algorithms. Apart from imperceptibility, capacity and robustness are two fundamental properties of audio watermarking schemes. The watermark should be extractable after various intentional and unintentional attacks. These attacks may include additive noise, re-sampling, MP3 compression, low-pass filtering, re-quantization, and any other attack which removes the watermark or confuse the watermark extraction system. Considering a trade-off between capacity, transparency and robustness is the main challenge for audio watermarking applications.

Many audio watermarking schemes take advantage of the properties of the human auditory system (HAS) and different transforms, resulting in various techniques such as embedding algorithms based on low-bit coding, echo, patchwork [3], rational dither modulation [4], Fourier transform [5, 6], quantization [7, 8, 10] and the wavelet transform [9, 11].

Considering the embedding domain, audio watermarking techniques can be classified into time domain and frequency domain methods. Time domain watermarking schemes are relatively easy to implement and require less computing resources compared to transform domain watermarking methods. On the other hand, time domain watermarking systems are usually weaker against signal-processing attacks compared to the transform domain counterparts. Phase modulation [1] and echo hiding [2] are well known methods in the time domain.

In frequency domain watermarking, after taking one of the usual transforms such as the Discrete/Fast Fourier Transform (DFT/FFT) [5, 6], the Modified Discrete Cosine Transform (MDCT) or the Wavelet Transform (WT) [9, 11, 17, 18] from the signal, the hidden bits are embedded into the resulting transform coefficients. For example, [18] takes advantage of the mean of absolute values to design a scheme which has capacity equal to 40 bits (which are embedded in a 20-second audio signal in the experiments given in the paper), and robustness against common attacks. In [5, 6] the FFT domain is selected to embed watermarks for making use of the translation-invariant property of the FFT coefficients to resist small distortions in the time domain. In particular, [5, 6, 9, 11, 17, 18] show that the frequency domain provides excellent robustness against attacks. In fact, using methods based on transforms provides a better perception quality and robustness against common attacks at the price of increasing the computational complexity.

Among the existing transforms, the wavelet transform has several advantages in audio signal processing. Its inherent frequency multi-resolution and logarithmic decomposition of the frequency bands resemble the human perception of frequencies, since it provides the decomposition to mimic the critical band structure of the HAS.

In the proposed scheme, the last high frequency band of second level wavelet decomposition (DD), for which the HAS is not very sensitive to alteration, is used for embedding. In the embedding process, the samples are changed based on the corresponding secret bit. The main idea is to select a part of the samples in each

frame and change them based on average of a relevant frame. *E.g.* if we want embed "1" into a sample with value equal to one, the value may be changed to 0.5, but if we want embed "1" in a sample with value ten, then it may be modified to 5. If we used 0.5 for embedding "1" at all samples, then the scheme would be very fragile to attacks. On the other hand, if we changed the values to 5 always, then we would be enforcing a large distortion to the marked audio signal. Thus, it is advisable to change the samples based on their values. To design a blind scheme and, also, to achieve good robustness and transparency results, the high frequency band (DD) is divided into small frames and the average of each frame is used as a reference value to change the value of the samples. These reference values are the same in the coder/decoder or sender/receiver. When the elements of a set are divided by their average, the new values of the elements will be near one. In this algorithm, we divide each element by the average of the corresponding frame and then we use all values in the interval [–*k*, *k*] for embedding, where *k* is the embedding interval value. If the secret bit is "0", the corresponding sample in the interval is changed to $-m_i$, whereas for embedding a "1" the sample is altered to $+m_i$ (where m_i is the mean of the *i*-th frame).

The experimental results show that high capacity, remarkable transparency and robustness against most of common attacks are achieved.

The rest of the paper is organized as follows. In Section 2, the proposed method is presented. In Section 3, a discussion on the transparency and robustness of the suggested scheme is provided, and the experimental results are shown. Finally, Section 4 summarizes the most relevant conclusions of this research.

2. Proposed scheme

A wide work has been performed over the years in understanding the characteristics of the HAS and applying this knowledge to audio compression and audio watermarking. Figure 1 shows a typical absolute threshold curve, where the horizontal axis is the frequency measured in hertz (Hz) and the vertical axis is the absolute threshold in decibels (dB). As it can be seen, human beings tend to be more sensitive towards frequencies in the range from 1 to 4 kHz, while the threshold increases rapidly at very high and very low frequencies. Based on the HAS, the human ear sensitivity in higher frequencies is lower than in middle

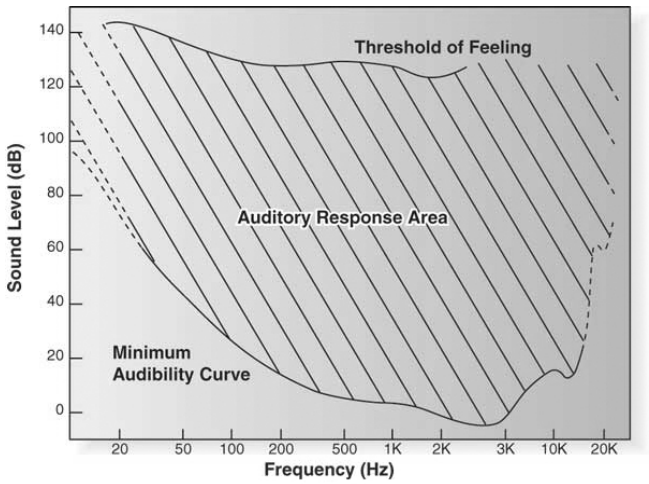


Fig 1. Typical absolute threshold curve of the human auditory response

frequencies. It is thus clear that, by embedding data in the high frequency band, which is used in the proposed scheme, the distortion will be mostly inaudible and thus more transparency can be achieved.

2.1 Embedding

The embedding steps are described below.

1. Compute the second level wavelet transform of the original signal.
2. Divide the cDD samples into frames of a given length and, based on average of the absolute values of each frame's samples, compute the average m_i for each frame by using Equation (1).

$$m_i = \frac{1}{s} \sum_{j=(i-1)s+1}^{is} |c_j| \qquad (1)$$

Where $\{c_j\}$ are the wavelet coefficients of the high-frequency sub-band (DD), s is the frame size and m_i is the average of the i-th frame.

3. The marked wavelet coefficients $\{c'_j\}$ are obtained by using Equation (2).

$$c'_j = \begin{cases} m_i & |c_j/m_i| < k,\ w_l = 1 \\ -m_i & |c_j/m_i| < k,\ w_l = 0 \\ c_j & |c_j/m_i| \geq k \end{cases} \qquad (2)$$

Where $i = \lfloor j/s \rfloor + 1$, m_i stands for the frame average, w_l is the l-th bit of the secret stream, k is the embedding interval ($k > 2$) and $\lfloor \cdot \rfloor$ denotes the floor function. *I.e.* if c_j in $[-km_i, km_i]$ then, depending on the secret bit, it is changed to $-m_i$ or $+m_i$. Each secret bit is embedded in a suitable sample and thus, after embedding the bit, the index l is incremented and the next secret bit is embedded in next suitable wavelet sample.

4. Finally, the inverse DWT is applied to the modified wavelet coefficients to get the marked audio signal.

The modified area of DWT coefficients for each frame is [$-km_i$, km_i] which is determined by the absolute mean value of each frame and the embedding interval, k. By increasing k, the interval is extended in such a way that the number of modified coefficients which satisfy the condition $\left|c_j/m_i\right| < k$ is increased and, thus, capacity and distortion also become greater. To manage robustness and transparency, we use a scale factor, α, which defines strength of watermark ($0.5< \alpha < k$). In fact, in Equation 2, instead of changing c_j' to m_i , we can change it to αm_i.

Fig. 2 illustrates the effect of the embedding steps in wavelet samples. Fig. 2 (a) shows the high frequency wavelet decomposition (cDD) of a RIFF WAVE file of a second, c_j. Fig. 2 (b) shows the modified samples c_j/m_i and Fig. 2 (c) illustrates the marked samples, c_j'. This figure shows that, by dividing samples by the average of each frame, all of them will be in the same range. It also illustrates that, after embedding, the marked samples are very similar to the original ones.

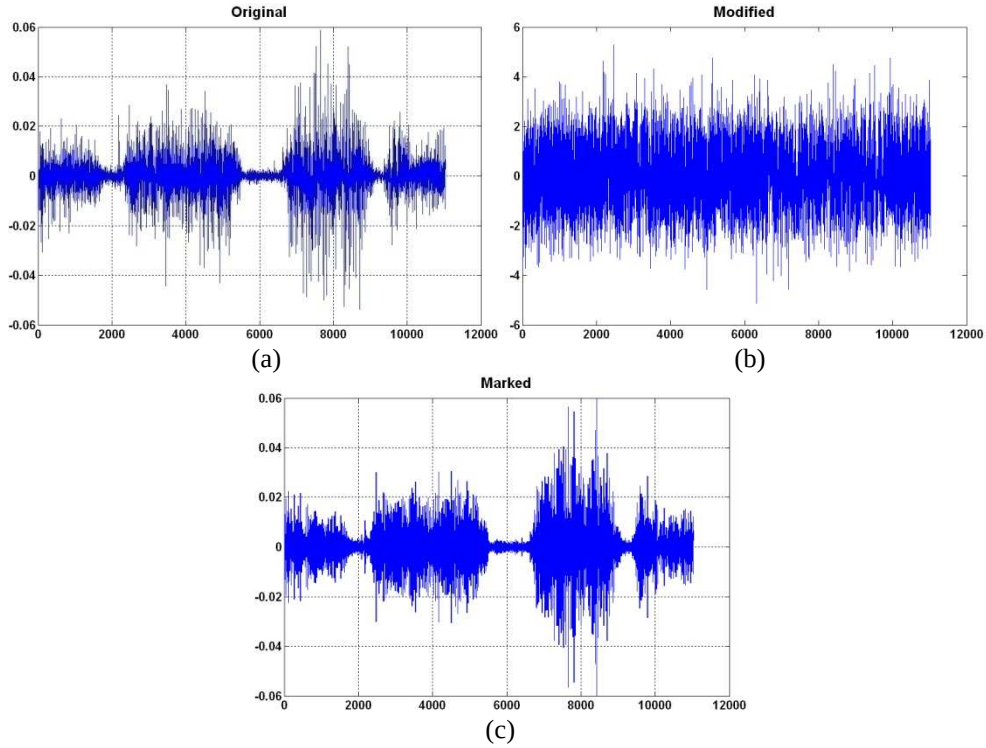


Fig. 2. Wavelet samples in embedding steps

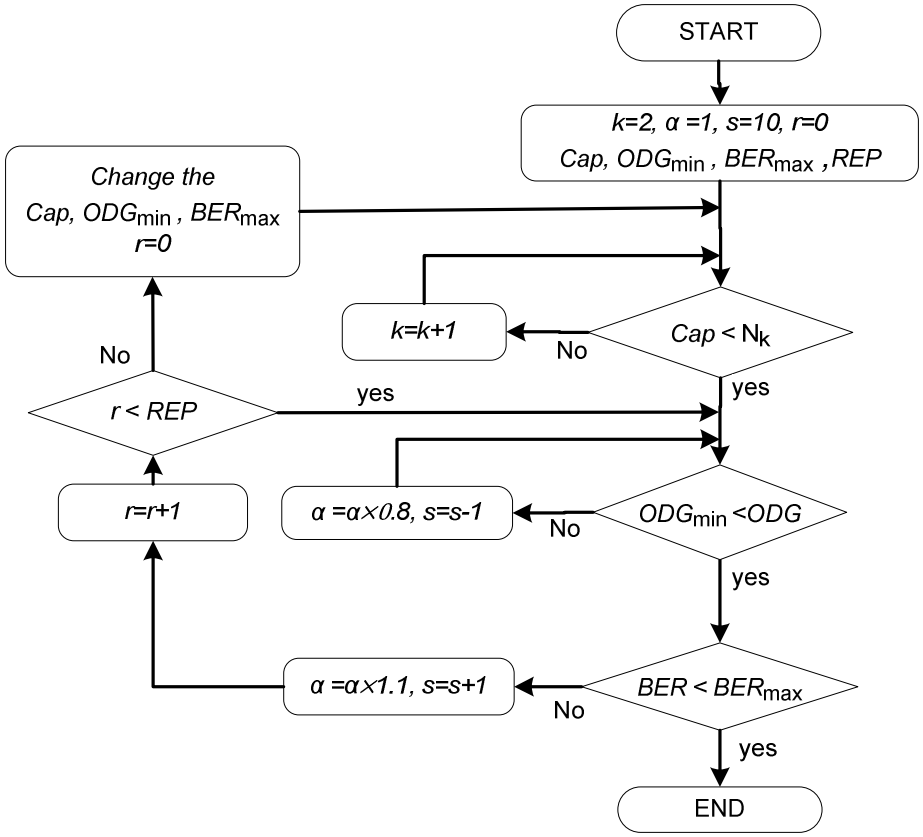


Fig. 3. Flowchart of tuning the embedding parameters

Fig. 3 shows the flowchart for the selection of the embedding parameters. In the flowchart, the required capacity is denoted by *Cap*, N_k is the number of samples in selected embedding interval, $ODG_{\min}$ is threshold of acceptable distortion, $BER_{\max}$ is maximum tolerable of BER and REP is the number of times the loop is repeated to reach to the demanded properties. In the tuning steps, first a suitable embedding interval, *k*, is fixed based on *Cap*. If there are not enough samples in the interval which is defined by *k*, the interval should be extended. *I.e.* by increasing *k*, the interval is extended and capacity is increased. Then *ODG* and *BER* are regulated by scale factor, α, and the frame size, *s.* As mentioned above, increasing α and *s* increases robustness and distortion. Thus, to obtain suitable transparency and robustness, these parameters can be changed. Considering the trade-off between the properties (capacity, robustness and transparency) of watermarking techniques, in some cases alteration in the requested properties is necessary. *E.g.* obtaining parameters to achieve *Cap* = 11,000 bps, $BER_{\max} = 0$ and $ODG_{\min} = 0.0$ is difficult or maybe impossible. However, finding tuning parameters to obtain $BER_{\max} = 1$ and $ODG_{\min} = -0.5$ should not be difficult.

2.2 Extracting

In the receiver, $m_i^{'}$, which stands for the marked frame average, is calculated by using Equation (3) and an interval is defined such that, if c_j' is in the interval, a secret bit can be extracted. The secret bit stream is achieved by using Equation (4).

$$m_i^{'} = \frac{1}{s} \sum_{j=(i-1)s+1}^{is} \left| c_j^{'} \right| \qquad (3)$$

$$w_l' = \begin{cases} 1 & 0 \le \left| c_j'/m_i' \right| \le ((k+\alpha)/2) \\ 0 & -((k+\alpha)/2) \le \left| c_j'/m_i' \right| < 0 \end{cases} \qquad (4)$$

Where c_j' is the sample of the high frequency band of the second level wavelet decomposition (cDD) of the marked signal, α is the strength of watermark and w_l' is the *l*-th bit of the extracted secret stream. *E.g.* if $k = 2$ and $\alpha = 1$ then, if c_j' in $[0, 1.5m_i']$ the secret bit is "1" and, if in $[-1.5m_i', 0)$, the secret bit is "0".

Since, in the coder, the DWT samples in the interval $[-km_i, km_i,]$ are changed to αm_i or $-\alpha m_i$. It is thus clear that the average of the absolute values is equal to αm_i in the receiver. If the signal is distorted by attacks, the absolute mean of the coefficients m_i' is slightly modified. However, the experimental results show that this change does not affect the extraction process since an interval, not a constant number, is used for extracting. *E.g.* under the MP3-128 compression attack, the variation is about 5% which is acceptable for extraction.

In a real application, the cover signal would be divided into several blocks of a few seconds and it is essential that the detector can determine the position (the beginning sample) of each of these blocks. One of the most practical solutions to solve this problem is to use synchronization marks such that the detector can determine the beginning of each block. [11] is used with the method described here in order to produce a practical self-synchronizing solution. Note that the synchronization method described in [11] is already shown to be robust against different types of manipulations and, more precisely, against attacks which lead to de-synchronization, such as re-sampling, re-quantization and random cropping. Because of this reason, de-synchronization attacks will not be examined in the experimental results of this paper, since they are already analyzed in [11].

To increase security, pseudo-random number generators (PRNG) can be used to change the secret bit stream to a stream which makes more difficult for an attacker

to extract the secret information. For example, the embedded bitstream can be constructed as the XOR sum of the real watermark and a pseudo-random bit stream. The seed of the PRNG would be required as a secret key both at the embedder and the detector [16].

3. Discussion and experimental results

To show the performance of the proposed scheme and to consider the applicability of the scheme in a real scenario, five songs (RIFF WAVE files) included in the album Rust by No, Really [12] have been selected. All audio clips are sampled at 44.1 kHz with 16 bits per sample and two channels. The two-level wavelet decomposition is implemented using the 8-coefficient Daubechies wavelet (db8) filter. The experiments have been performed for each channel of the audio signals separately. We provide imperceptibility results both as SNR and Objective Difference Grade (ODG), where ODG = 0 means no degradation and ODG = –4 means a very annoying distortion. SNR is provided only for comparison with other works, but ODG is a more appropriate measurement of audio distortions, since it is assumed to provide an accurate model of the subjective difference grade (SDG) results which may be obtained by a group of human listeners. The SNR results are computed using the whole (original and marked) files, whereas the ODG results are provided using the advanced ITU-R BS.1387 standard [13] as implemented in the Opera software [14] (which computes the average ODG of measurements taken in frames of 1024 samples).

In order to reduce computation time and memory usage, each song is divided into clips of 10 seconds, and the synchronization [11] and embedding algorithm is applied for each clip separately. We embed 16 synchronization bits, "1 0 1 1 0 0 1 1 1 1 0 0 0 0 1 0" with a quantization factor equal to 0.125, in the first 80 samples of each clip, then the information watermark is embedded and, finally, all these clips are joined together to generate the marked signal.

3.1 Discussion on transparency and robustness

This section provides a discussion of the robustness and transparency of the suggested scheme. These results are not purely theoretical, but the reasons why both transparency and robustness are achieved are outlined. These results have

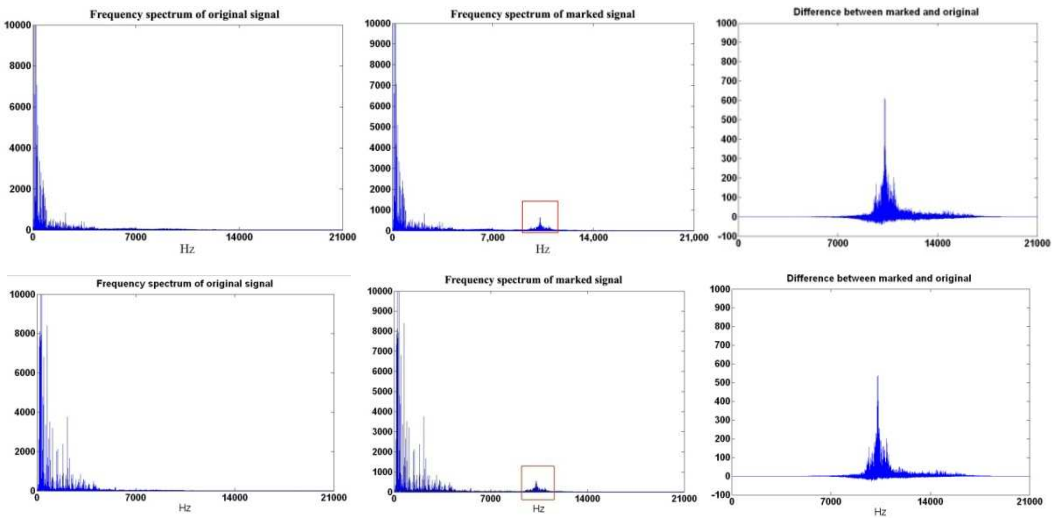


Fig. 4. Spectrum of the original, marked and difference between marked and original signals for two audio files of [12]

been obtained for $k = 6$, $\alpha = 2$ and the frame size equal to 10, but they can be easily extended to other values of the tuning parameters. As transparency is concerned, Fig. 4 shows the spectrum of the original, the marked signals and the difference between them. To make the comparison between the original and the marked signals easier, the scale difference of the difference has been magnified 10 times. Note that most changes occur around the 10 kHz region, where the human auditory response is not as sensitive (the auditory threshold is about 20 dB) as it is in lower frequencies ranges, such as [200, 5000] Hz. These plots do not prove that the distortion introduced around 10 kHz is below the audible threshold (20 dB), since the final power of the final audio signal depends on different factors such as the physical device used to generate it, including different parameters such as volume and equalizers. However, the experimental results given in Section 3.2, in terms of both ODG and SNR, show that the imperceptibility of the suggested scheme is very remarkable (imperceptible or not annoying).

With respect to robustness, considering general attacks and their effect on the marked signal in a theoretical manner is a complex process, since the effect depends on both the embedding scheme and the attack. For example, in an embedding method using the LSB of the signal, it is evident that attacks like LSBZero, requantization and Amplify will remove the secret information.

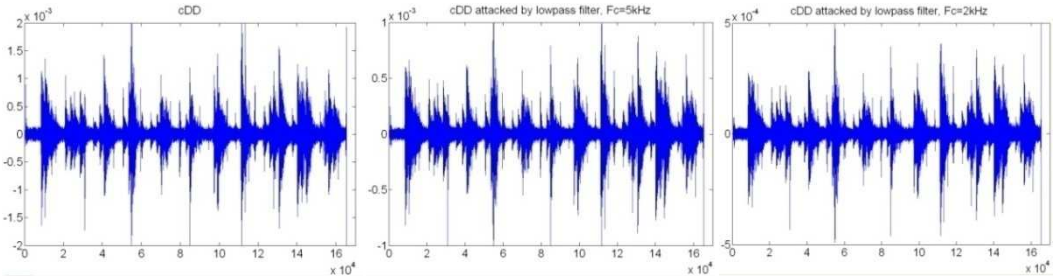
In our case, the proposed scheme takes advantage of the wavelet transform, which is a time-frequency function, and thus to consider the theoretical reasons why some attacks are survived needs complex equations and conditions which can be different for different types of audio signals.

However, note that this scheme is robust against all attacks which produce a scaling change in the DD wavelet coefficients. If the cDD wavelet samples are scaled, the mean of these samples is scaled accordingly, and the extracting process is still successful. Such a scaling change in the DD area occurs in several attacks. A simple attack which produces scaling is Amplify, which changes the amplitudes of the (time domain) samples.

Another attack which is relevant for this particular scheme is RC low-pass filtering, since the high frequency area is used for embedding in this scheme, what would seem to imply that the suggested scheme is fragile against this kind of attack. However, the suggested method is able to overcome these attacks, as shown in Fig. 5. Fig. 5 (a) shows the original cDD, Fig. 5 (b) illustrates the cDD of the signal attacked using an RC low-pass filter with cut-off frequency equal to 5 kHz and Fig. 5 (c) shows the cDD of the signal attacked using an RC low-pass filter with a cut-off frequency equal to 2 kHz. It can be noticed that these RC low-pass filters do not destroy the cDD samples, but their amplitudes are scaled down by some factor (lower than 1). However, as mentioned above, this change in the scale does not affect the extracting process, since we use the ratio between the wavelet sample and the average of its frame. Hence, if the cDD samples are scaled then the average of the samples is scaled as well, the ratio is not changed by scaling and the extraction procedure is successful. In case of using other kind of filters with attenuation higher than that of RC low-pass filters, the watermark might be erased, but the perceptual quality of the attacked file would also be seriously damaged (since all frequencies beyond the cut-off frequency would be practically suppressed).

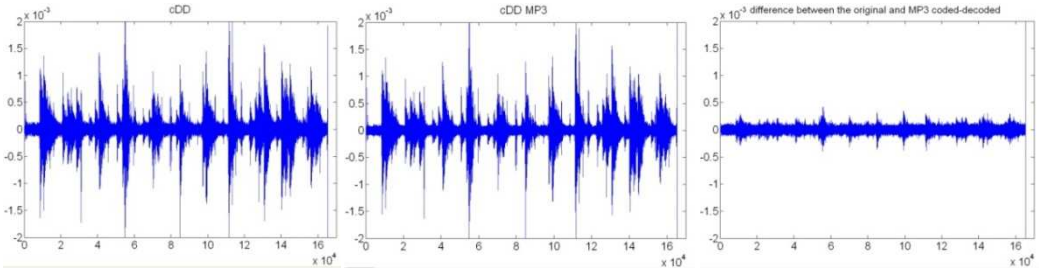
To consider the effect of MP3 compression and RC low-pass filter on the high frequency band of the wavelet decomposition, a part of “Beginning of the End” audio file is used as a sample and the attacks are performed on it. Fig. 6 (a) shows the last high frequency band of the two-level wavelet decomposition with the

8- coefficient Daubechies wavelet, cDD of 15 seconds of “Beginning of the End”. As Fig. 6 (b) illustrates, the cDD, samples after coding and decoding by MP3-128, are similar to the original cDD samples. Furthermore, Fig. 6 (c) shows that the difference between the original and coded-decoded cDD samples is too small to affect the extracting process, as the experimental results presented in the next section show.



(a) (b) (c)

Fig. 5. cDD (a) original (b) after RC_lowpass with cut-off frequency 5 kHz (c) after RC_lowpass with cut-off frequency 2kHz



(a) (b) (c)

Fig. 6. cDD (a) original (b) coded-decoded (c) difference between original and coded-decoded

3.2 Experimental results and comparative analysis

Table 1 shows the perceptual distortion and the payload obtained for the five songs with BER equal to zero (or near zero) under the attacks detailed in Table 2, for k = 6, α = 2 and the frame size equal to 10. In fact, by selecting k = 6, almost all wavelet samples are used for embedding. The following conditions can be assumed to obtain different capacity and transparency:

1. No robustness. In this case, very good capacity and transparency can be achieved.

2. Robustness against MP3 is demanded. In this case, more distortion should be accepted, compared with Condition 1.

3. Robustness against the attacks in Table 2 is demanded. This is more complicated than the previous conditions since we need robustness against most common attacks. Thus, according to trade-off between capacity, transparency and robustness, a sacrifice in capacity and transparency is required.

The results provided below try to provide robustness against common attacks. We have used several random bits for embedding, leading to different transparency results which are shown in the ODG column. Note that all the results have an ODG between 0 (not perceptible) and –1 (not annoying), the average SNR is 30 dB and capacity is around 11,000 bps for all the experiments. The proposed

method is thus able to provide large capacity whilst keeping imperceptibility in the admitted range (–1 to 0).

Table 2 illustrates the effect of various attacks provided in the Stirmark Benchmark for Audio v1.0 [15] on ODG and the BER for the five audio signals of Table 1. The synchronization [11] which is robust against common attacks and the embedding method described in section 2 have been used and, then the SMBA software has been used to attack the whole marked files. Finally, the attacked file is scanned in time domain to find the synchronization codes then the secret information of each clip is extracted. The ODG in table 2 is calculated between the marked and the attacked-marked files.

Table 1: Results of 5 mono signals (robust against table 2 attacks)

Audio File	***Time (m:sec)***	***SNR (dB)***	***ODG of marked***	***Payload (bps)***
Beginning of the End	3:16	30 to 33.1	–0.4 to –0.8	11003
Citizen, Go Back to Sleep	1:57	26.8 to 31.2	–0.6 to –0.9	11001
Go	1:51	29 to 32.2	–0.7 to –0.9	11005
Thousand Yard Stare	3:57	31.4 to 35.1	–0.2 to –0.8	11002
Rust	2:33	26.2 to 30.3	–0.6 to –0.8	10999
Average	**2:43**	**30**	**– 0.7**	**11002**

Table 2: Robustness test results for five selected files and comparison with schemes in this literature

Attack name	**ODG of attacked file**	***parameters***	***BER %***						
			proposed	**[3]**	**[4]**	**[5]**	**[9]**	**[10]**	**[17]**
AddBrumm	–3.1 to –3.7	1-5k, 1-6k	0 to 1	–	0	0 to 1	–	–	–
AddDynNoise	–2.1 to –2.5	1-2	2 to 7	–	2	0 to 8	–	–	–
ADDFFTNoise	–0.3 to –0.1	2048,400	0 to 2	–	1	1 to 2	–	–	–
Addnoise	–0.8 to –0.4	1-20	0 to 6	2	1	0 to 1	–	0	5 to 25
AddSinus	–3.1 to –2.5	1-5k,1-7k	0	–	0	0	–	–	–
Amplify	–0.2 to –0.0	20 - 200	0 to 1	–	0	0	–	–	–
BassBoost	–3.8 to –3.3	1-50,1-50	6 to 14	–	–	0	–	–	–
Echo	–3 to –1.3	1-5	1 to 28	1.2	63	0 to 1	–	6	–
FFT_HLPassQuick	–3.7 to –3.3	2048,1-10k,18k-22k	12 to 17	–	5	1 to 4	–	–	–
FFT_Invert	–3.8 to –3.1	1024	0	–	2	1 to 2	–	–	–
FFT_RealReverse	–3.5 to –3	2-2048	14 to 29	–	–	–	–	–	–
FFT_Stat1	–3.6 to –2.9	2-2048	21 to 37	–	1	–	–	–	–
Invert	–3.6 to –2.8	-	0	–	–	0	–	–	–
Resampling	–2.1 to –1.8	44/22/44	7 to 11	1	0	5	0	0	0
LSBZero	– 0.2 to 0.0	-	0	–	0	0	–	0	–
MP3	– 0.4 to 0.0	≥128	0 to 2	0.3	–	0 to 5	0	–	1
Noise_Max	– 0.4 to –0.1	1-2,1-14k,1-500	1 to 4	–	–	0 to 1	–	–	–
Pitchscale	–3.7 to –3.1	1.1	31 to 51	–	–	0 to 1	–	–	–
RC_HighPass	–3.7 to –3.1	1-14k	0 to 5	–	–	0 to 1	–	–	–
RC_LowPass	–3.8 to –0.4	2k – 22k	0 to 8	2	0	0	0	3	–
Smoth	–3.6 to –3.3	–	14 to 22	–	–	–	–	–	–
Stat1	–2.1 to –1.4	–	9 to 12	–	8	–	–	–	–
TimeStretch	–3.8 to –3.2	1.05	34 to 61	–	–	–	–	–	–
Quantization	–0.6 to –0.2	16-12	5 to 9	0.5	–	–	0	0	0

The parameters of the attacks are defined based on SMBA web site [15] for the proposed scheme. Other schemes may use different parameters. For example, in AddBrumm, 1-5 k shows the strength and 1-6 k shows the frequency. This row illustrates that any value in the range 1-5 k for the strength and 1-6 k for the frequency could be used with slightly change in BER. In fact, this table shows the range (the worst and best) of ODG and BER for the five test signals. When the BER is (slightly) greater than zero, it can be made zero by using Error Correction Codes at the price of reducing the capacity. The BER column for proposed scheme shows the total BER after embedding synchronization mark and watermark. *E.g.* the BER of the BassBoost attack changes from 0 to 2 without considering the synchronization however BER is increased to 6 to 14 after using synchronization.

Only a few attacks such as Pitchscale and TimeStretch in Table 2 remove the hidden data (BER > 15%). Note, however, that the ODG of these attacks are extremely low (about –3.5). This means that these attacks do not only remove the hidden data, but also destroy the perceptual quality of the host signal.

As already remarked, this scheme uses the high frequency band of the wavelet coefficients for embedding. Hence, it may seem that it would be fragile against attacks which manipulate or suppress the high frequency data. In Table 3, The MP3 and RC low-pass filter attacks are analyzed in depth with different types of audio clips. This table shows that the BER is increased by decreasing the MP3 rate also by decreasing cut-off frequency of the low-pass filter. In spite of that, the suggested method is still robust (BER < 15%) against these attacks for a wide range of the attack parameters.

In Table 4, we compare the performance of recent audio watermarking strategies, which are robust against common attacks, with the proposed method. [4] measures distortion using the mean opinion score (MOS), which is a subjective measurement, and achieves transparency between imperceptible and perceptible

Table 3. Robustness results for a variety of audio types under MP3 and RC Low-pass filter attacks

MP3 attack	MP3 rate	320	256	192	160	128
	BER	0	0 to 2	0 to 4	0 to 5	2 to 13
	ODG of attacked file	0.0	–0.1 to 0.0	–0.2 to 0.0	–0.2 to 0.0	–0.4 to– 0.1
RC low-pass attack	Cut-off frequency of low-pass filter (kHz)	20	15	10	5	2
	BER	0 to 1	0 to 1	0 to 3	1 to 5	4 to 15
	ODG of attacked file	–0.2 to –0.0	–0.5 to 0.0	–0.7 to –0.3	–1.8 to –0.8	–3.7 to –2.7

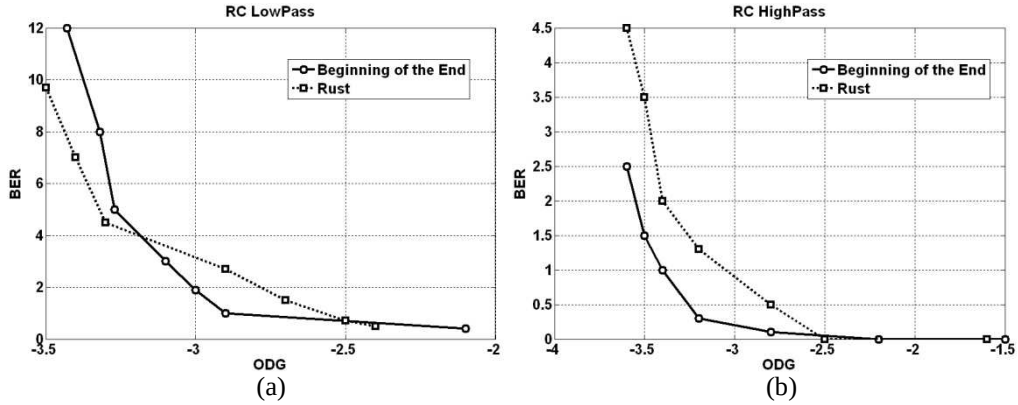
Table 4: Comparison of different watermarking algorithms

Algorithm	*Audio File*	*SNR (dB)*	*ODG of marked*	*Payload (bps)*
[3]	Song	25	–	86
[4]	Song	–	–	689
[5]	Song	30.5	– 0.6	2996
[9]	Song	30	–	172
[10]	Classical music	25	–	176
[17]	Song	25–40	–	172
proposed	Song	30	– 0.7	11002

but not annoying (MOS = 4.7). [10, 17] propose low capacity schemes, but they are robust against most common attacks. In particular, [17] is robust against most common signal processing and attacks, such as Gaussian noise, re-sampling, re-quantization, and MP3 compression. Although the chosen schemes from the literature use different audio signals and attack parameters, the properties of each algorithm in capacity of embedding secret information and transparency are summarized in Table 4, and robustness against attacks is shown in Table 2. The comparison shows that the compared schemes are robust against common attacks and transparency is in an acceptable range, about 30 dB. However, the capacity of these schemes is just a few hundred bps (except for the method [5]). This comparison shows that the capacity of the proposed scheme is very remarkable, whilst keeping the transparency and BER in their acceptable ranges.

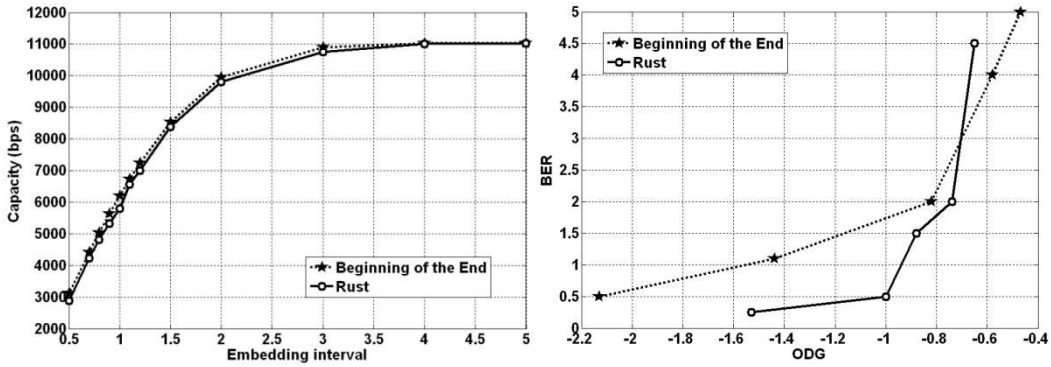
Using frames of wavelet samples results in an increased robustness against attacks, since the average of the samples is more robust than the value of each sample. Thus, by increasing the frame size, better robustness can be achieved. However, by increasing the frame size, we enforce the same value for a greater number of samples, which decreases the audio quality and transparency. In our experiments, the frame size equal to 10 has provided excellent transparency and acceptable robustness, but, depending on the specific application, this value might be adjusted.

It may seem that using high frequencies for embedding the secret bits would lead to a fragile scheme against low-pass filtering. Indeed, the experimental results show that the secret stream is damaged by low-pass filters with a cut-off frequency lower than 2 kHz, but these filters damage the cover signal as well. Fig. 7 shows that, under the RC low-pass/high-pass filter attacks, the secret bit stream is extractable (BER < 5%) even when the ODG between the marked and the attacked file is about –3. *I.e.* this kind of filtering removes the secret information



(a) (b)

Fig. 7. Transparency versus BER under (a) low pass filtering attack (b) high pass filtering attack



(a) (b)

Fig. 8. (a) Capacity versus embedding interval (b) BER under Gaussian Noise versus ODG for various scale factors, α

only if the quality of the attacked file is far from acceptable (in the "very annoying" ODG scale). As mentioned above, depending on the specific application, the embedding interval and the scale factor could be changed. *E.g.* if k = 6 and α = 1 for the clip "Beginning of the End", ODG = –0.4 and BER under the attack MP3-128 is 0.07, but for k = 6 and α =2, ODG = –0.6 and BER = 0.01. This example shows how the tuning parameter α can be used to tune the trade-off between transparency and robustness. The embedding interval, k, and the scale factor, α, play a relevant role in adjusting the properties of the scheme. In fact, these parameters adjust the trade-off between capacity, transparency, and robustness. Fig. 8 (a) shows that increasing the embedding interval increases the number of modified samples in the interval, which defines the capacity of the scheme. Similarly, Fig. 8 (b) illustrates the effect of the scale factor, (watermark strength) on transparency (ODG between original and marked signal) and robustness against Gaussian Noise (BER). It is obvious that with a small scale factor better transparency is achieved and increasing α leads to better robustness (decreasing BER) and more distortion. It is worth pointing out that, in the experimental results shown in this figure, α is chosen in the interval [0.5, 3].

4. Conclusion

Using the high frequency band of the wavelet decomposition, for which the human auditory system (HAS) is not very sensitive to alteration, leads to a robust high-capacity watermarking algorithm for digital audio. The proposed scheme divides the high frequency band into frames and uses the frames' average (which is the same in the sender and receiver for each frame) as a key value, resulting in a blind scheme which provides robustness against common audio signal processing attacks. The experimental results show that this scheme has an excellent capacity (about 11 kbps) without significant perceptual distortion (ODG in the range [–1, 0] and SNR about 30 dB) and provides robustness against common signal processing attacks such as added noise, echo, filtering or MPEG compression (MP3). A comparison with other schemes in the audio watermarking literature is also provided, showing that the suggested scheme outperforms the capacity of other approaches whilst keeping robustness and transparency in their acceptable ranges.

References

[1] N. Lie, L. C. Chang, "Multiple Watermarks for Stereo Audio Signals Using Phase-Modulation Techniques", IEEE Trans. Signal Processing, Vol. 53, No. 2, pp. 806–815, Feb. 2005.

[2] H. J. Kim, Y. H. Choi, " A novel echo hiding scheme with backward and forward kernels", IEEE Trans. Circuit and Systems, pp. 885-889, Aug. 2003.

[3] H. Kang, K. Yamaguchi, B. Kurkoski, K. Yamaguchi, and K. Kobayashi, "Full-Index-Embedding Patchwork Algorithm for Audio Watermarking", *IEICE TRANS. on Information and Systems*, E91-D(11):2731-2734, 2008

[4] J. J. Garcia-Hernandez, M. Nakano-Miyatake and H. Perez-Meana, "Data hiding in audio signal using Rational Dither Modulation", *IEICE Electron. Express*, Vol. 5, No. 7, pp.217-222, 2008.

[5] M. Fallahpour, D. Megías, "High capacity audio watermarking using FFT amplitude interpolation" *IEICE Electron. Express*, Vol. 6, No. 14, pp. 1057-1063, 2009.

[6] M. Fallahpour, D. Megías, "High Capacity Method for Real-Time Audio Data Hiding Using the FFT Transform", Advances in Information Security and Its Application, Springer-Verlag. pp. 91-97, 2009.

[7] B. Chen , G. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Trans. Inf. Theory*, vol. 47, no. 4, pp. 1423–1443, May 2001.

[8] Z. Xu, K. Wang, X.h. Qiao, "Digital Audio Watermarking Algorithm Based On Quantizing Coefficients," *IEEE Proceedings of the International Conference on Intelligent Information Hiding and Multimedia Signal Processing* 0-7695-2745-0/06 , 2006.

[9] M. Pooyan, A. Delforouzi, “Adaptive and robust audio watermarking in wavelet domain” *Third International Conference on International Information Hiding and Multimedia Signal Processing* , V2 Pages 287-290, 2007

[10] M. A. Akhaee, M. J. Saberian, S. Feizi, F. Marvasti, “Robust Audio Data Hiding Using Correlated Quantization With Histogram-Based Detector” IEEE TRANS. ON Multimedia,V11, P 1-9, 2009.

[11] X.-Y. Wang and H. Zhao. A novel synchronization invariant audio watermarking scheme based on DWT and DCT. IEEE Trans. on Signal Processing, 54(12):4835–4840, 2006.

[12] No, Really, “Rust”. http://www.jamendo.com/en/album/7365

[13] T. Thiede, W. C. Treurniet, R. Bitto, C. Schmidmer, T. Sporer, J. G. Beerens, C. Colomes, M. Keyhl, G. Stoll, K. Brandenburg, and B. Feiten, “PEAQ - The ITU Standard for Objective Measurement of Perceived Audio Quality,” *Journal of the AES*, vol. 48(1/2), pp. 3–29, 2000.

[14] OPTICOM OPERA software site. http://www.opticom.de/products/opera.html.

[15] Stirmark Benchmark for Audio. http://wwwiti.cs.uni-magdeburg.de/~alang/smba.php.

[16] D. Megías, J. Herrera-Joancomartí, and J. Minguillón. “Total disclosure of the embedding and detection algorithms for a secure digital watermarking scheme for audio”. *Proceedings of the Seventh International Conference on Information and Communication Security*, pp. 427-440, Beijing, China, December 2005.

[17] S. Wu, J. Huang, D. Huang, Y. Shi, Efficiently self-synchronized audio watermarking for assured audio data transmission, IEEE Trans. Broadcasting 51 (1). 69–76, 2005.

[18] S. Xiang, H.J. Kim, J. Huang, “Audio watermarking robust against time-scale modification and MP3 compression” *Signal Processing*,v.88, 2372-2387, 2008.