An Implementation of Advanced Audio Watermarking Algorithm Based on Spread Spectrum(SS) Technology

Zichao Zhang, 300145760, Hao Zhang, 300111466, Faculty of Engineering, University of Ottawa

Abstract—The proposed watermarking algorithm introduces enhanced method based on spread spectrum with higher embedding capacity. The conventional algorithms cannot efficiently embed more information without comprising robustness or imperceptibility. Hadamard matrix is used to generate orthogonal carrier sequences, which embed multiple watermark bits.

Index Terms—Audio watermarking, spread spectrum, discrete cosine transform, robustness, human auditory system, interleaving, Hadamard matrix, hamming code.

I. Introduction

Watermarks, first introduced by paper manufacturers in $18th$, served as labels for consumers to distinguish the work of paper makers. They had been laterly used on postage stamps, currency, and other government documents. Replicating an analog product costs much more than purchasing a copy. Nowadays, extensive application of digital data is making reproduction and spreading of multimedia data much more easier, which at the same time gives rise to multimedia piracy.

A fine approach is the embedded audio watermark, the main subject of our project. According to [1], different kinds of watermarks can be categorized as secret watermarks and public watermarks, both should be performing well in signal processing properties, security properties and general properties. Till now, many embedding methods have emerged, based on various schemes as Spread Spectrum(SS), echo-hiding, patchwork, and others.

In our project, we chose watermarking algorithm built on spread spectrum technique with high capacity, which is introduced in [2]. Depending on the domain that embedding the watermarks, watermarking algorithms can be splited into two kinds: embedded in time domain and embedded in transform domain.[3] Time-domain embedding schemes are easy and perform well in real-time but fragile to fundamental attacks. Typical algorithm such as Lowest Significant Bits(LSB) is easy to implement, which determins the vulnerability of this algorithm. The lowest bit in a byte is less significant that can be used to embed watermark information, which is hard for human eyes to perceive. Basic modifications and attacks like additive white gaussian noise can easily destroy the watermark.

Meanwhile, embedding on transform domain shows quite satisfying results. Utilizing human perceptual characteristics, researchers embed watermarks on frequency domain or DCT domain which makes watermarks imperceivable.

II. Watermarking Algorithm

A. Conventional Scheme

Here we choose [2] as the basis of our proect. The whole watermarking procedure can be modeled as a communication system: the watermark embedder as transmitter, the attacks as noise in channel, and the detector as receiver. The multimedia data is considerd as channel, watermark information is the message we want to send. And watermark is embedded in host audio signal like a message transmitting in channel. According to the theory of channel capacity of Shannon, in a specified channel, if transmitting power is rather low, in another word, the signal to noise ratio(SNR) in the receiving side is not enough to assure correct reception, we can compensate which by expanding the bandwith of the transmitted signal. That's the thinking in Spread Spectrum(SS) communication. Basically, we can append some other properties to gain even better performance. In [4] and this paper, the masking properties of the human auditory system(HAS) is refered to, which can be modeled as 26 bandpass filters. One of two signals becomes imperceptible if they lie in the same band and the other signal is louder.

The existing audio watermarking methods got advanced step by step, involving [2], each progress is based on the previous achievements. In general, they all follow a similiar procedure: firstly generate a pseudo-noise sequence as the spreading spectrum sequence, then multiplicate the sequence with the watermark bit, embed the product into a specific transform domain. The embedded sequence usually is scaled by a HAS coefficient to enhance imperceptibility. There are generally three ways to embed the sequence, which will be explained later.

To model the whole process, we start from the representing of vectors. As said in [5], we have $\mathbf{x}$ as the coefficient vector extracted from the specific transform of origianl signal and the vector $\mathbf{y}$ is the received vector, of course, in the transform domain. A "key" is used to generate the pseudo-noise sequence, $\mathbf{p}$, which can also be produced in the receiver side only if the key is known. The $\mathbf{y}$, $\mathbf{x}$, $\mathbf{p}$ have the same length N. We assume signal $\mathbf{x}$ to be uncorrelated

white Gaussian stochastic process, let x_i denote elements in $\mathbf{x}$, then we get

$$x_i \sim N(0, \sigma_x^2) \tag{1}$$

The watermark bit is denoted by b, here b takes values from $\{+1, -1\}$, and the HAS coefficient is α. As for embedding, the three basic ways are $\mathbf{y} = \mathbf{x} + \alpha b\mathbf{p}$, $\mathbf{y} = \mathbf{x} * (1 + \alpha b\mathbf{p})$ and $\mathbf{y} = \mathbf{x}(e^{\alpha b\mathbf{p}})$. As we can see, if the amplitude of x_i is rather high compared to embedding data, the perceptual quality won't degrade much, however, there will be a serious distortion if it's a weak signal and the amplitude of x_i is close to or smaller than the embedding data. So we choose the second way, as said before the embedded signal is.

$$\mathbf{y} = \mathbf{x} + \alpha b\mathbf{p} \tag{2}$$

Every bit of information we embed requires one PN sequence, and in the conventional way, the watermark is detected by correlating the watermarked signal and the same PN sequence as used in transmitting side. we define

$$z = \frac{\mathbf{y}\mathbf{p}^T}{\mathbf{p}\mathbf{p}^T} \tag{3}$$

The dectected bit is decided by the symbol of z, b is 1 when $z > 0$, -1 for $z < 0$, 0 for $z = 0$. Substitute (2) into (3), we get

$$\begin{aligned} z &= \frac{(\mathbf{x} + \alpha b\mathbf{p})\mathbf{p}^T}{\mathbf{p}\mathbf{p}^T} \\ &= \alpha b + x' \end{aligned} \tag{4}$$

and

$$x' = \frac{\mathbf{x}\mathbf{p}^T}{\mathbf{p}\mathbf{p}^T} \tag{5}$$

As this method requires nothing of original signal, it is a blind method. But x' interferes us from correctly extracting the embedded bit.

So there is a need to eliminate the influence of the original siginal. A method talked in [5] is to set a coefficient λ

$$\mathbf{y} = \mathbf{x} + (\alpha b - \lambda x')\mathbf{p}, \quad 0 < \lambda \leq 1 \tag{6}$$

We get a new equation

$$z = \alpha b + (1 - \lambda)x' \tag{7}$$

If let λ approach 1, the influence of x' can be very small, which result in better robustness, but at the same time it lowers perceptual quality since value of b can be negative. To ensure imperceptiblity, the length N should be long enough to make x' small enough. The value of $\mathbf{p}$ is either $+1$ or -1, and the mean value of which is 0, so

$$\begin{aligned} D(x') &= D(\frac{\mathbf{x}\mathbf{p}^T}{\mathbf{p}\mathbf{p}^T}) \\ &= \frac{1}{N^2} D(\sum_{i=0}^{N-1} x_i p_i) \\ &= \frac{1}{N^2} \sum_{i=0}^{N-1} D(x_i p_i) \\ &= \frac{1}{N^2} \sum_{i=0}^{N-1} E(x_i^2 p_i^2) - [E(x_i)E(p_i)]^2 \\ &= \frac{1}{N^2} \sum_{i=0}^{N-1} E(x_i^2) \\ &= \frac{\sigma_x^2}{N} \end{aligned} \tag{8}$$

This shows that when N gets bigger, the variance or standard deviation of x' can become smaller, which means the values of x' closer to its mean value, zero. However, longer original siginal segment also means lower embedding capacity, which leads to the research of better way for embedding.

III. Proposed Algorithm

A. Watermark Embedding Process

The proposed watermark embedding scheme mainly includes three sections.

a) Generation of Near-orthogonal PN Sequences

The proposed algorithm changed the way to embed watermark bits, conventional ways use one PN sequence for one watermark bit, it's more efficient to represent more than one bit with one sequence. For example, if we want to represent two bits with one sequence, like $\{+1, +1\}, \{+1, -1\}, \{-1, +1\}, \{-1, -1\}$, we will need four sequences.

Meanwhile, considering the complexity of using 2^{N_b} sequences to represent watermark information, the author suggests to use a rotationally shifted seed PN sequence randomly generated.

b) DCT Operation and Segmentation

The DCT coefficients, denoted by $X(k)$, are obtained after conducting DCT transform on original signal $x(n)$ with length K.

$$X(k) = l(k) \sum_{n=0}^{K-1} x(n) \cos\{\frac{\pi(2n+1)k}{2K}\} \tag{9}$$

$k = 0, 1, 2, \cdots, K-1$.

$$l(x) = \begin{cases} \frac{1}{\sqrt{K}} & , \text{if} \quad k = 0 \\ \sqrt{\frac{2}{K}} & , \text{if} \quad 1 \leq k \leq K \end{cases} \tag{10}$$

In DCT transform, we only take frequency components robust to attacks, so a certain range of frequency is indicated as $[f_l, f_h]$, f_l and f_h. The selected coefficients

are divided into N_s segments, the length of which is $2N$. The ith segment is represented as

$$X_i(k) = [X_i(0), X_i(1), \cdots, X_i(2N-1)] \quad i = 0, 1, \cdots, N_s \tag{11}$$

The $X_i(k)$ is then divided in to even and odd parts $\mathbf{x}_{i,1}$ and $\mathbf{x}_{i,2}$

$$\mathbf{x}_{i,1} = [X_i(0), X_i(2), \cdots, X_i(2N-2)] \tag{12}$$

$$\mathbf{x}_{i,2} = [X_i(1), X_i(3), \cdots, X_i(2N-1)] \tag{13}$$

c) Modification on DCT Coefficients

Now let PN sequence $\mathbf{p}_t$ represent the watermark bits to be embedded into ith $X_i(k)$ segment. The embedding method is

$$\tilde{\mathbf{x}}_{i,1} = (\mathbf{1} + \beta\mathbf{p}_t) \circ \mathbf{x}_{i,1} \tag{14}$$

$$\tilde{\mathbf{x}}_{i,2} = (\mathbf{1} + \beta\mathbf{p}_t) \circ \mathbf{x}_{i,2} \tag{15}$$

$\mathbf{1}$ is a length N row vector, whose elements are all 1. $\circ$ is Hadama operator, β is the HAS constant and $0 < \beta < 1$. Change the notion of elements of $\tilde{\mathbf{x}}_{i,1}$ and $tilde\mathbf{x}_{i,2}$, we get

$$\tilde{\mathbf{x}}_{i,1} = [\tilde{X}_{i,1}(0), \tilde{X}_{i,1}(1), \cdots, \tilde{X}_{i,1}(N-1)] \tag{16}$$

$$\tilde{\mathbf{x}}_{i,2} = [\tilde{X}_{i,2}(0), \tilde{X}_{i,2}(1), \cdots, \tilde{X}_{i,2}(N-1)] \tag{17}$$

The watermarked signal must be reconstructed as follows:

$$\begin{aligned}\tilde{X}_i(k) = [&\tilde{X}_{i,1}(0), \tilde{X}_{i,2}(0), \tilde{X}_{i,1}(1), \tilde{X}_{i,2}(1), \cdots, \\ &\tilde{X}_{i,1}(N-1), \tilde{X}_{i,2}(N-1)]\end{aligned} \tag{18}$$

At last, we get watermarked signal by inverse DCT transform. The preliminary code is appended.

B. Watermark Extraction Process

As said in the Introduction section, most existing algorithms based on spread spectrum suffer a common problem: the interference of host signal. The influence can be weakened by introducing the difference between two adjacent samples. Generally, the frequency domain of a natural signal shows a degree of continuity, so the difference between a pair of adjacent samples won't be significant, ordinarily. That's why each embedding segment in the host signal was separated into two sequences of coefficients and was manipulated differently. After the receiver gets the watermarked signal, correlation operation is conducted to extract watermark information, so the original pseudo-noise sequence is required. However, we don't need to transmit the whole set of sequences, as the whole set is rotationally shifted and only the first sequence is randomly generated by a 'Key'. Thus, only the key for regenerating the PN sequence is transmitted.

Assume the received signal is $y(n)$, either with the presence of attack or not, if without attacks, $y(n) = \tilde{x}(n)$. Similarly, conduct DCT transform on $y(n)$, then extract the coefficients of the same index in embedding process, which can be easily realized knowing f_L, f_H, N_s and N. In the extracting session, our group took the value from [2], $N = 750$, $N_p = 64$. The experiment audio segments are also 10 seconds long, which has $441k$ samples each, we indicate $f_L = 0$ and $f_H = 90000$ because we found that the main part of the spectrum is in that domain. But a drawback of this frequency is that common attacks aiming at frequency domain like low pass filter can easily mess up the watermark.

After obtaining the DCT coefficients, represented by $\mathbf{y}_{i,1}$ and $\mathbf{y}_{i,2}$, both of which the length is N. We can know

$$\mathbf{y}_{i,1} = \tilde{\mathbf{x}}_{i,1} = (1 + \beta\mathbf{p}_t) \circ \mathbf{x}_{i,1} \tag{19}$$

$$\mathbf{y}_{i,2} = \tilde{\mathbf{x}}_{i,2} = (1 + \beta\mathbf{p}_t) \circ \mathbf{x}_{i,2} \tag{20}$$

Because $\mathbf{p}_t$ take values from $\{-1, +1\}$, $0 < \beta < 1$, $1 \pm \beta$ will always bigger than 0. Thus, we have

$$\begin{aligned}\mathbf{y}_{i,d} &= |\mathbf{y}_{i,1}| - |\mathbf{y}_{i,2}| \\ &= |(1 + \beta\mathbf{p}_t) \circ \mathbf{x}_{i,1}| - |(1 - \beta\mathbf{p}_t) \circ \mathbf{x}_{i,2}| \\ &= (1 + \beta\mathbf{p}_t) \circ |\mathbf{x}_{i,1}| - (1 - \beta\mathbf{p}_t) \circ |\mathbf{x}_{i,2}| \\ &= (|\tilde{\mathbf{x}}_{i,1}| - |\mathbf{x}_{i,2}|) + \beta\mathbf{p}_t \circ (|\mathbf{x}_{i,1}| + |\mathbf{x}_{i,2}|)\end{aligned} \tag{21}$$

The absolute operation represents taking absolute value for every element in the vector. Next step is to find which PN sequence was embedded, which is implemented with correlation between a matrix composed of all possible PN sequences and the difference of the received signal $\mathbf{y}_{i,d}$.

Create a matrix $\mathbf{P}0$ in the receiver side

$$\begin{aligned}\mathbf{P}0(:, 1) &= \mathbf{p}_1 \\ \mathbf{P}0(:, 2) &= \mathbf{p}_2 \\ &\cdots \\ \mathbf{P}0(:, N_p) &= \mathbf{p}_{N_p}\end{aligned} \tag{22}$$

We can tell that $\mathbf{P}0$ is a matrix with N_p columns and N rows. Then multiply $\mathbf{P}0$ with $\mathbf{y}_{i,d}$, find the maximum in the result matrix, which is a row matrix with N_p columns. The extraction is based on the irrelevance between any two $\mathbf{p}$ vectors. Assume $\mathbf{p}_t$ is embedded in $\mathbf{y}_{i,d}$, after multiply $\mathbf{y}_{i,d}$ with $\mathbf{P}0$, we get

$$\begin{aligned}\mathbf{y}_{i,d}\mathbf{p}_t^T =&(|\mathbf{x}_{i,1}| - |\mathbf{x}_{i,2}|)\mathbf{p}_t^T \\ &+ \beta(\mathbf{p}_t \circ (|\mathbf{x}_{i,1}| + |\mathbf{x}_{i,2}|))\mathbf{p}_t^T \\ =&(|\mathbf{x}_{i,1}| - |\mathbf{x}_{i,2}|)\mathbf{p}_t^T \\ &+ \beta(\mathbf{p}_t \circ \mathbf{p}_t)(|\mathbf{x}_{i,1}^T| + |\mathbf{x}_{i,2}^T|) \\ =&(|\mathbf{x}_{i,1}| - |\mathbf{x}_{i,2}|)\mathbf{p}_t^T \\ &+ \beta\mathbf{1}(|\mathbf{x}_{i,1}^T| + |\mathbf{x}_{i,2}^T|)\end{aligned} \tag{23}$$

Meanwhile, the product of received signal and other PN sequences

$$\begin{aligned}\mathbf{y}_{i,d}\mathbf{p}_j^T =&(|\mathbf{x}_{i,1}| - |\mathbf{x}_{i,2}|)\mathbf{p}_j^T \\ &+ \beta(\mathbf{p}_t \circ (|\mathbf{x}_{i,1}| + |\mathbf{x}_{i,2}|))\mathbf{p}_j^T \\ =&(|\mathbf{x}_{i,1}| - |\mathbf{x}_{i,2}|)\mathbf{p}_j^T \\ &+ \beta(\mathbf{p}_t \circ \mathbf{p}_j)(|\mathbf{x}_{i,1}^T| + |\mathbf{x}_{i,2}^T|)\end{aligned} \tag{24}$$

If we compare this two, because $\mathbf{x}_{i,1}$, $\mathbf{x}_{i,2}$ are independent of $mathbfp_j$, $(|\mathbf{x}_{i,1}| - |\mathbf{x}_{i,2}|)\mathbf{p}_j^T$ and $(|\mathbf{x}_{i,1}| - |\mathbf{x}_{i,2}|)\mathbf{p}_t^T$ are comparable. Since pt and pj are random sequences obtained by rotational shift so there are little relevance between them, approximately half the elements of $pt \circ pj$.

Thus, $xi1T + Xi2T$ is usually much smaller than with a row vector of full 1s. So this product comes to a maximum when $t = j$. We can find the maximum number along with its index then we get the corresponding 6bits code word by mapping the index to the table of code words.

C. Selection of β

β represents the embedding coefficient, the vaue of which determines the robustness and imperceptibility of watermarks. When β gets bigger, a higher extent of robustness is achieved while compromises the imperceptibility at the same time. So β shouldn't be too small nor too big, so a range of value for β is set, $[\beta_{min}, \beta_{max}]$, they are determined by exeriments. Basically, if we want to correctly detect the watermark, we should ensure $\mathbf{y}_{i,d}\mathbf{p}_t^T$ bigger than $\mathbf{y}_{i,d}\mathbf{p}_j^T$, which means

$$(|\mathbf{x}_{i,1}| - |\mathbf{x}_{i,2}|)\mathbf{p}_t^T + \beta\mathbf{1}(|\mathbf{x}_{i,1}^T| + |\mathbf{x}_{i,2}^T|) > (|\mathbf{x}_{i,1}| - |\mathbf{x}_{i,2}|)\mathbf{p}_j^T + \beta(\mathbf{p}_t \circ \mathbf{p}_j)(|\mathbf{x}_{i,1}^T| + |\mathbf{x}_{i,2}^T|)$$

That is

$$\beta > \frac{(|\mathbf{x}_{i,1}| - |\mathbf{x}_{i,2}|)(\mathbf{p}_j^T - \mathbf{p}_t^T)}{(\mathbf{1} - (\mathbf{p}_t \circ \mathbf{p}_j))(|\mathbf{x}_{i,1}^T| + |\mathbf{x}_{i,2}^T|)} \tag{25}$$

So β should at least meet this requirement. In this paper, the β value is specified in an analysis-by-synthesis approach, which creates a buffer for mistaken detection while maintains β the least required value.[2] First, a matrix $\bar{\mathbf{p}}$ is made by

$$\bar{\mathbf{p}} = \left[\mathbf{p}_1^T, \cdots, \mathbf{p}_{t-1}^T, \mathbf{p}_{t+1}^T, \cdots, \mathbf{p}_{N_p}^T\right] \tag{26}$$

Then a few more steps are taken:

1) At the beginning, set β to β_{min}, construct the $\bar{\mathbf{p}}$ by (26).
2) compute
   $\mathbf{d} = (|\mathbf{x}_{i,1}| - |\mathbf{x}_{i,2}|) + \beta\mathbf{p}_t \circ (|\mathbf{x}_{i,1}| + |\mathbf{x}_{i,2}|)$
   $u_1 = \mathbf{d}\mathbf{p}_t^T$
   $u_2 = max(\mathbf{d}\bar{\mathbf{p}}^T)$
   max returns the maximum value of its parameters.
3) If $\gamma_1 u_1 > \gamma_2$, set $v = \gamma_1 u_1$. Otherwise, set $v = \gamma_2$
4) If $u_2 \geq u_1 - v$, go to the next step. Otherwise, end.
5) Add β by $\triangle\beta$. If $\beta \leq \beta_{max} - \triangle\beta$, go to the second step. Otherwise, end.

The value of γ_1 and γ_2 can be chosen experimentally. In our simulation, $\gamma_1 = 0.1$, $\gamma_2 = 2$, and $\beta_{min} = 0.001$, $\beta_{max} = 0.2$, $\triangle\beta = 0.005$. We can see v provides a buffer for errors, either $\gamma_1 u_1$ or γ_2 ensures that the product of embedded sequence and corresponding PN sequence is bigger than that of embedded sequence and other PN sequences, which means, $u_1 > u_2$.

IV. Follow-up work of the author

Right after we finished implementing the proposed algorithm and the debugging session, we realized that our result isn't so satisfying. The difference between the embedded audio segments and their original version were quite obvious. Even worse, we did as said in [2], set β to β_{min} and the detection and decoding results are terrible. So we found the first author's website online for further completion of this scheme. Fortunately, a newer research [6] was found, which is an improvment based on [2].

A. Generation of Orthogonal PN Sequences

This improved watermark embedding scheme is said to perform better than its former design. The multiple orthogonal PN sequences is introduced, which is said to have fixed the problem of host signal interference. Rather than rotationally shift a PN sequence, which creates a series of near-orthogonal PN sequence, [6] used a set of orthogonal PN sequences. Similarly, $\mathbf{p}_0$ is also a randomly generated pseudo-noise sequence by a seed, with length L_N. $\mathbf{p}_0$ takes values from $\{-1, +1\}$, According to Gram-Schmidt orthogonalisation method, we construct an L_N by L_N identity matrix $\mathbf{P}_0$, then substitute the first column by $\mathbf{p}_0^T$, i.e.,

$$\mathbf{P}_0(:, 1) = \mathbf{p}_0^T \tag{27}$$

Then we get every orthogonal sequence horizontally by orthogonalisation method. Represent row number of $\mathbf{P}_0$ by j, so the jth row will be $\mathbf{P}_0(:, j)$. The orthogonal PN sequences are $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{N_p}$.

$$\begin{aligned}
\mathbf{p}_1' &= \mathbf{P}_0(1, :),\ \mathbf{p}_1 = \frac{\mathbf{p}_1'}{\|\mathbf{p}_1'\|} \\
\mathbf{p}_2' &= \mathbf{P}_0(2, :) - \langle \mathbf{P}_0(2, :), \mathbf{p}_1 \rangle \mathbf{p}_1,\ \mathbf{p}_2 = \frac{\mathbf{p}_2'}{\|\mathbf{p}_2'\|} \\
&\vdots\ \vdots \\
\mathbf{p}_{N_p}' &= \mathbf{P}_0(N_p, :) - \langle \mathbf{P}_0(N_p, :), \mathbf{p}_{N_p-1} \rangle \mathbf{p}_{N_p-1} \\
&\quad - \langle \mathbf{P}_0(N_p, :), \mathbf{p}_{N_p-2} \rangle \mathbf{p}_{N_p-2} \\
&\quad - \cdots - \langle \mathbf{P}_0(N_p, :), \mathbf{p}_1 \rangle \mathbf{p}_1,\ \mathbf{p}_{N_p} = \frac{\mathbf{p}_{N_p}'}{\|\mathbf{p}_{N_p}'\|}
\end{aligned}$$

$\| \ \|$ is the norm of a vector, $\langle \ \rangle$ means the inner product of two vectors, $\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=1}^{N} a_i b_i$. Similarly, we also only need to pass the $\mathbf{p}_0$ or just pass the key to generate it. The general equation can be written as

$$\mathbf{p}_j = \frac{\mathbf{P}_0(j, :) - \sum_{l=1}^{j-1} \langle \mathbf{P}_0(j, :), \mathbf{p}_l \rangle \mathbf{p}_l}{\|\mathbf{P}_0(j, :) - \sum_{l=1}^{j-1} \langle \mathbf{P}_0(j, :), \mathbf{p}_l \rangle \mathbf{p}_l\|} \tag{28}$$

Where $j = 1, 2, 3, \ldots, N_p$.

B. Modification of DCT Coefficients

Unlike in [2], the DCT coefficients are modified in a addition way, denote the coefficients to be modified by $\mathbf{x}_i$, let $\mathbf{p}_t$ be the orthogonal PN sequence associated with the concerned watermark bits, $\mathbf{y}_i$ be the watermarked counterpart of $\mathbf{x}_i$.

$$\mathbf{y}_i = \mathbf{x}_i + \alpha_i \left(\frac{\mathbf{x}_i \mathbf{p}_t^T}{|\mathbf{x}_i \mathbf{p}_t^T|} \right) \mathbf{p}_t \tag{29}$$

Right now the problem is to find a suitable α_i. Here, α_i is also related to embedding coefficient, so just like what we

did before, we constraint α_i in a certain range of value. However, as we can see in the figure below,

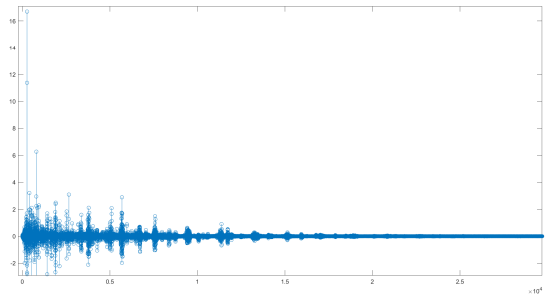


Fig. 1. The DCT coefficients of audio segment

the DCT coefficients decrease as the frequency goes up. So α_i should also become smaller at bigger i. So α_i should be determined this way

$$\alpha_i = (\alpha_{upper} - \alpha lower)e^{-c(i-1)} + \alpha_{lower} \tag{30}$$

Where α_{upper} and α_{lower} are the boundaries of α_i, c is a constant, in the simulation session of this paper its value is 0.002, α_{upper} is set to 0.6, α_{lower} is set to 0.3.

C. Watermark Extraction

The received audio signal is denoted by $y'(n)$, again, we apply DCT transform on $y'(n)$. As you can see, here we didn't separate the coefficients into two groups, because our new approach to eliminate the interference of host signal is the introduction of orthogonal sequences. Segment it into N parts, $\mathbf{y}'_1, \mathbf{y}'_2, \cdots, \mathbf{y}'_N$. Assume there's no attacks, we conduct multiplication on

$$D_{i,j} = |\mathbf{y}_i \mathbf{p}_j^T| \quad j = 1, 2, \cdots, N_p, i = 1, 2, \cdots, N \tag{31}$$

When $j = t$, replace $\mathbf{y}_i$ from (29).

$$\begin{aligned} D_{i,t} &= |\mathbf{y}_i \mathbf{p}_t^T| \\ &= \left| \left(\mathbf{x}_i + \alpha_i \left(\frac{\mathbf{x}_i \mathbf{p}_t^T}{|\mathbf{x}_i \mathbf{p}_t^T|} \right) \mathbf{p}_t \right) \mathbf{p}_t^T \right| \\ &= \left| \mathbf{x}_i \mathbf{p}_t^T + \alpha_i \left(\frac{\mathbf{x}_i \mathbf{p}_t^T}{|\mathbf{x}_i \mathbf{p}_t^T|} \right) \|\mathbf{p}_t\|^2 \right| \\ &= |\mathbf{x}_i \mathbf{p}_t^T| \cdot \left| 1 + \alpha_i \left(\frac{\|\mathbf{p}_t\|^2}{|\mathbf{x}_i \mathbf{p}_t^T|} \right) \right| \end{aligned} \tag{32}$$

Because α_i, $\|\mathbf{p}_t\|^2$, and $|\mathbf{x}_i \mathbf{p}_t^T|$ are positive values, rewrite (32)

$$D_{i,t} = |\mathbf{x}_i \mathbf{p}_t^T| \cdot \left(1 + \alpha_i \left(\frac{\|\mathbf{p}_t\|^2}{|\mathbf{x}_i \mathbf{p}_t^T|} \right) \right) \tag{33}$$

When $j \neq t$,

$$\begin{aligned} D_{i,j} &= \left| \left(\mathbf{x}_i + \alpha_i \left(\frac{\mathbf{x}_i \mathbf{p}_t^T}{|\mathbf{x}_i \mathbf{p}_t^T|} \right) \mathbf{p}_t \right) \mathbf{p}_j^T \right| \\ &= \left| \mathbf{x}_i \mathbf{p}_j^T + \alpha_i \left(\frac{\mathbf{x}_i \mathbf{p}_t^T}{|\mathbf{x}_i \mathbf{p}_t^T|} \right) \mathbf{p}_t \mathbf{p}_j^T \right| \end{aligned} \tag{34}$$

Since $\mathbf{p}_t \mathbf{p}_j^T = 0$

$$D_{i,j} = |\mathbf{x}_i \mathbf{p}_j^T| \tag{35}$$

Because $\|\mathbf{p}_t\|^2 \gg |\mathbf{x}_i \mathbf{p}_t^T|$, we get

$$\left(1 + \alpha_i \left(\frac{\|\mathbf{p}_t\|^2}{|\mathbf{x}_i \mathbf{p}_t^T|} \right) \right) \gg 1 \tag{36}$$

Besides, $|\mathbf{x}_i \mathbf{p}_j^T|$ and $|\mathbf{x}_i \mathbf{p}_t^T|$ are comparable,

$$D_{i,t} \gg D_{i,j}, \quad \forall j \neq t \tag{37}$$

So the index t is found by

$$t = \quad \text{argmax} \quad D_{i,j} \tag{38}$$

D. Problems in the Simulation of This Scheme

Unfortunately, we didn't have single one correct detection of watermark bits. If we replace the first column of matrix with random sequence taking values from ± 1, the result vectors will be a series of row vectors with only one not-zero value, which is ± 1. So this can be very vulnerable as a detection method.

V. Adjustment on the Proposed Algorithm

In [2], audio segments for testing session are set in the duration of 10 seconds. Audio file has sampling rate of $44.1k$Hz, every sample is quantized with 16 bits, which corresponds to the *wav* file format. However the simulation result didn't go as we expected. At first, we set the β value to the minimum, as said in the [2], then we experimentally increase the value of β, and found that the minimum required β value varies with different audio segment. So we believe this embedding scheme needs developers decide β value for every audio file they are going to modify. Basically, our design and program follow the block diagram in fig.2. Moreover, we changed the f_L and f_H, which can be a deficiency when it's under the attack by high pass filter.

In the follow-up work, a method that requires orhtogonal PN sequence is applied, so we took this idea and made some changes. Including the orthogonal sequence, some other methods are also taken to improve performance under attacks.

VI. Modifications and Improvements

The method presented by [2] has reasonable performance in our testing, but this method can be optimized.

The bitrate of this method is limited by the spectrum used, and the efficiency of spectrum can be improved. And, in this method a circular shifted pseudorandom PIN sequence is used to represent different state of the signal, but the PN sequence varies in different situation, and this could reduce the performance in some scenario. Also, in original method no data redundancy method is applied to the random sequences, it is possible that some specific attack will damage the data the watermarking signal contains.

We are capable of find a few approaches of the method that could be used to improve this watermarking method.

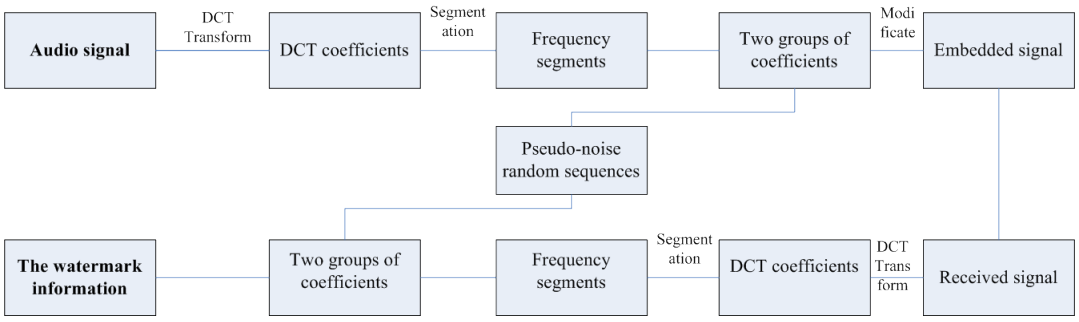


Fig. 2. The block diagram of introduced scheme

The core of [2] is implementing different sequence into a part of the frequency spectrum to represent different state of signal. In the original method. Due to the characteristic of a pseudo-random sequence, in different scenario, random generator will generate different PN sequence, thus the performance may also vary. Also, circular shift this sequence into different sequences result in a non-uniform hamming distances between these sequences, thus more possible to having errors in decoding.

A. Orthogonal sequences

In our approach, instead of relying on unstable circular-shifted randomly generated sequences, we are using orthogonal sequences in this method. Orthogonal sequences have united hamming distance between each code, thus make decoding less possibly to have an error. Also, we are using Hadamard matrix to generate uniform orthogonal sequences in different scenario. A Hadamard matrix is the matrix used in CDMA for generating orthogonal code words. Generating a Hadamard matrix by hand is rather easy, although we have the help of Matlab.

$$H_1 = [1]$$
$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$
$$H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

Fig. 3. Generating of Hadamard matrix

Continuously append the last matrix with the negative matrix of it, then compute negative, append which under the results in the last step, we get a Hadamard matrix. While this does have an effect of the watermarking method (a separate encryption of the watermarking signal is required), the performance does improve with this approach. In our testing, this approach alone averagely improves the performance by about 5 percent.

B. Better robustness gain by lower capacity

In the original method, the efficiency of the method can be improved. By using a shorter orthogonal sequence, and a lower amount of state, we are capable of fitting double amount of data into the same spectrum, while does not significantly affect the audio signal.

C. Hamming code

To improve our method's performance against an attack, we also introduced error-correction coding to the watermarking signal. By using an error-correction coding, this method is capable to correct a number of errors generated by implementing the watermark or attacking. We decided to use a 7/4 hamming code in this method, which result in a 75% increase of spectrum used.

D. Interleaving

Interleaving is used to avoid continuous errors, the interleaving depth is 2, which means every two hamming code word is used to construct a two row martix and we take information bit vertically to embed watermark into frames.

E. Non-blind detection

Also, we modified the decoder to use the original signal to improve performance if possible. By comparing the signal between watermarked and original, we are capable to reduce the amount of watermarking noise by 15db, while retain the same bit error rate. In addition, by utilizing the characteristic of orthogonal sequence generated by Hadamard sequence, we watermarked the data into two different spectrum, and the decoder is capable of retrieving data from these two sequences by using double-length sequences, which are also orthogonal, thus improve the performance when decoding blindly.

Approach We Are Using When Testing

In our test, we are using a 64x64 Hadamard matrix to generate orthogonal sequences, and we uses 4 of them to represent 2 bit of data. The details are shown in the picture.

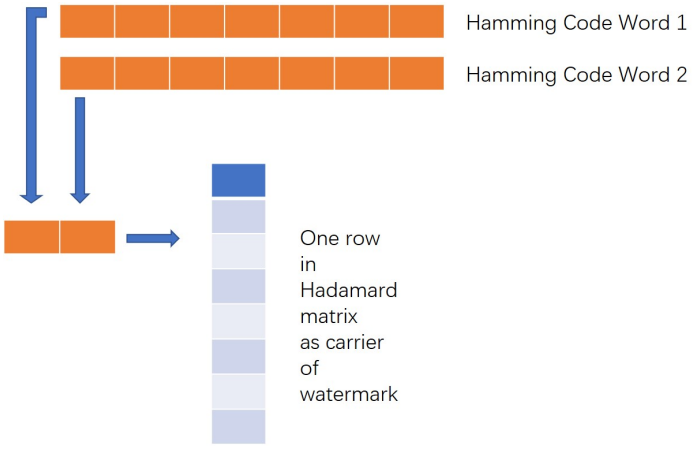


Fig. 4. Using of Hadamard marix

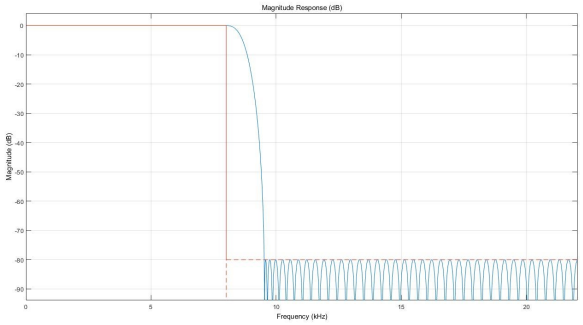
In total 8 seconds of audio, We are capable of watermarking 1408 bit of actual watermark data in 1232 signals into two channel of original signal. We use 0-10 KHz and 10-20 KHz to store two identical data for redundancy.

VII. Evaluations and Attacks

A. Attacks

Currently we are working on the evaluation method for our approach, since there does not exist a unified evaluation method now. For testing purposes, we are using our own testing and evaluation method unless the specific method is covered in future orientation. For testing of noise imperceptibility, we are using PEAQ method to test the perceptibility of the noise added by our watermarking algorithm.[7][8]

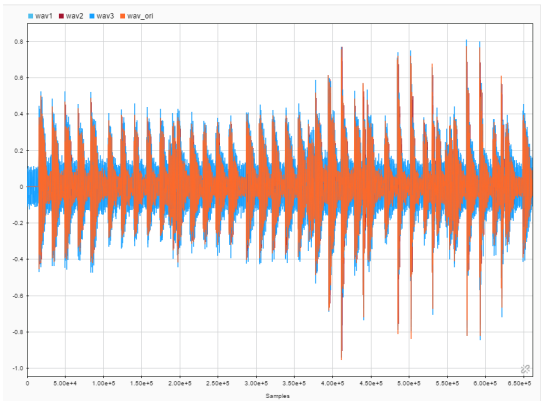
To test the robustness of the algorithm, the project performance will be evaluated using MATLAB. For low pass filtering, we are using a 8khz low pass filter on the test audio signal to attack the watermarking information.



For lossy compression, we are utilizing lame mp3 encoder to create a 16bit sample rate, ABR 128kbps version of compressed watermarked audio for testing.

For Gaussian white noise, we are currently gradually adding a 10db white noise to the audio signal.

The effect of these approach is described in graph below.



B. Performance

1) SNR under different payloads

During out test, the SNR under different capacity are as below:

TABLE I
SNR under different payloads

Audio file	"Piano.wav"		"Jazz.wav"		"Pop.wav"	
SNR/BER	SNR	BER	SNR	BER	SNR	BER
20 bits/s	26.0207	0	26.0206	0	26.0521	0
75 bits/s	26.0207	0	26.0206	0	26.0521	0
170 bits/s	26.0207	0	26.0206	0	26.0521	0

The data is obtained under no attacks, non-blind detection with $\beta = 0.05$. We can see that under different pay loads the SNR are the same, it's because we didn't cut down actual payload but repeated the same information, so in the receiving side the same message is received many times, which is a fine way to enhance robustness.

2) Performance under AWGN attack

As we can see in II, the power of noise has gone beyond that of watermark, so the performance under AWGN attack isn't satisfying. Nearly 0.5 error probility is like randomly generaing the result.

3) Re-sampling

As we can see in III, when we raise the down-sampling rate, we also get worse performance.

4) Re-quantization

As shown in IV, the watermark in different files are going through different damages.

5) Low-pass filtering

We concentrated watermark in the first $8kHz$ frequency range, and then put the same watermark in the frequency range behind it, but the length of carrier sequence is doubled. So the watermark goes through worst damage when cut off frequency is $4kHz$. And of course there's no influence on the watermark when the cut off frequency is $20kHz$.

6) Mp3 compression

The higher the bit rate, the better the performance is.

VIII. Future Works

As we've finished the algorithm proposed in this paper, we are only left further improvements and modifications

TABLE II
Under AWGN attack($\beta = 0.05$, non-blind detection)

Audio file	"Piano.wav"		"Jazz.wav"		"Blues.wav"		"Pop.wav"		"Classic.wav"	
BER/NC	BER	NC	BER	NC	BER	NC	BER	NC	BER	NC
15dB	0.5	0.5690	0.4951	0.5594	0.4932	0.5625	0.4912	0.5706	0.4912	0.5630
18dB	0.4971	0.5705	0.4932	0.5618	0.493	0.561	0.4922	0.5713	0.4932	0.5612
20dB	0.4990	0.5695	0.4922	0.5637	0.493	0.559	0.4971	0.5661	0.4912	0.5636

TABLE III
Re-sampling($\beta = 0.05$, non-blind detection)

Audio file	"Piano.wav"		"Pop.wav"		"Blues.wav"		"Jazz.wav"		"Classic.wav"	
BER/NC	BER	NC	BER	NC	BER	NC	BER	NC	BER	NC
22.05kHz	0.1387	0.8889	0.2891	0.7546	0.1299	0.8954	0.2764	0.7633	0.1426	0.8849
11.025kHz	0.3125	0.7384	0.4678	0.5884	0.4355	0.6234	0.3936	0.6589	0.4189	0.6346

TABLE IV
Re-quantization($\beta = 0.05$, non-blind detection)

Audio file	"Piano.wav"		"Jazz.wav"		"Classic.wav"		"Pop.wav"		"Blues.wav"	
BER/NC	BER	NC	BER	NC	BER	NC	BER	NC	BER	NC
re-quantization	0.3086	0.7431	0.4248	0.6315	0.3262	0.7238	0.335	0.7119	0.291	0.755

TABLE V
Low-pass filtering($\beta = 0.05$, non-blind detection)

Audio file	"Piano.wav"		"Jazz.wav"		"Classic.wav"		"Pop.wav"		"Blues.wav"	
BER/NC	BER	NC	BER	NC	BER	NC	BER	NC	BER	NC
$f_c = 4kHz$	0.0146	0.9883	0.0439	0.9647	0.03	0.975	0.0137	0.9891	0.0059	0.9953
$f_c = 8kHz$	0.002	0.9984	0.0195	0.9843	0.0078	0.993	0	1	0.0059	0.9953
$f_c = 20kHz$	0	1	0	1	0	1	0	1	0.0020	0.9984

TABLE VI
Mp3 compression($\beta = 0.05$, non-blind detection)

Audio file	"Pop.wav"		"Jazz.wav"		"Classic.wav"		"Piano.wav"		"Blues.wav"	
BER/NC	BER	NC	BER	NC	BER	NC	BER	NC	BER	NC
$64kbps$	0.441	0.614	0.483	0.570	0.489	0.569	0.484	0.577	0.494	0.571
$128kbps$	0.388	0.664	0.386	0.655	0.477	0.580	0.454	0.604	0.445	0.616
$256kbps$	0.172	0.857	0.118	0.903	0.368	0.686	0.186	0.847	0.342	0.706

on this algorithm. We fail to accomplish the performance showed in the paper, so one choice is to ask the authors for source code to test the feasibility of this algorithm. As we know, frame length is strongly related to ability to correct detection, longer frame length is good for better detection performance. Further work can focus on extending frame length and compromising payload. Also, the interferance from host signal is still a big problem, applying orthogonal sequence only lower the BER a little in blind detection tests. If we want to achieve high correction rate, we must damage the perceptibility or lower the payload.

IX. Conclusions

In this report, we implemented a novel SS-based watermarking method for audio signals.The proposed method can embed multiple watermark bits into one audio segment, which increases embedding capacity. The watermark embedding is realized by inserting a corresponding orthogonal sequence from a Hadamard matrix into a pair of frames in the segment, which have similar property.

This embedding scheme can reduce the host signal interference occurred in the process of watermark extraction and thus enhances robustness. However, in the experiments the results of blind detection tests are not so satisfying. By controling the embedding coefficient β, we can control the perceptibility. So, the proposed method exhibits good performance in terms of high imperceptibility, robustness and embedding capacity.Compared with the newest SS-based watermarking algorithms, this scheme is easy to implement but the results of simulation are not so good as said in the paper. The efforts we've done improved the performance a little bit, but it's far from the results showed in their paper. Host signal interference is still a nonnegligible factor and is effecting our result pretty much. I hope an innovative algorithm that can fix this problem will be proposed in the future.

References

[1] M. Arnold, "Audio watermarking: features, applications and algorithms," in 2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No.00TH8532), vol. 2, pp. 1013–1016 vol.2, July 2000.

[2] Y. Xiang, I. Natgunanathan, Y. Rong, and S. Guo, "Spread spectrum-based high embedding capacity watermarking method for audio signals," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 23, pp. 2228–2237, Dec 2015.
[3] Y. Wang, J. Xiao, Y. Wang, et al., Theories and Technologies of Digital Watermarking, vol. 5. , 2007.
[4] X. Zhao, Y. Guo, J. Liu, and Y. Yan, "A spread spectrum audio watermarking system with high perceptual quality," in 2011 Third International Conference on Communications and Mobile Computing, pp. 266–269, April 2011.
[5] H. S. Malvar and D. A. F. Florencio, "Improved spread spectrum: a new modulation technique for robust watermarking," IEEE Transactions on Signal Processing, vol. 51, pp. 898–905, April 2003.
[6] Y. Xiang, I. Natgunanathan, D. Peng, G. Hua, and B. Liu, "Spread spectrum audio watermarking using multiple orthogonal pn sequences and variable embedding strengths and polarities," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 26, pp. 529–539, March 2018.
[7] N. Andersson, "PEAQ." https://github.com/NikolajAndersson/PEAQ, 2017. [Online; accessed 30-May-2017].
[8] H. Mu, W. Gan, and E. Tan, "An objective analysis method for perceptual quality of a virtual bass system," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 23, pp. 840–850, May 2015.